

Informatyka



Zmiana MEN 2024



Oznaczono usunięte
i zmienione treści

3

Wojciech Hermanowski
Sławomir Sidor

Informatyka

Podręcznik dla **szkół ponadpodstawowych**

3

ZAKRES ROZSZERZONY

 **OPERON**
Edukacja jest podróżą

2024

Projekt okładki: Paweł Kwacz
Redaktor prowadzący: Sebastian Przybyszewski
Redakcja językowa: zespół
Redakcja graficzna i skład: Mateusz Zaborowski
Korekta: zespół
Zdjęcia: Shutterstock

OPERON Sp. z o.o. oświadcza, że z należytą starannością podjął wszelkie działania zmierzające do powiadomienia podmiotów majątkowych praw autorskich do utworów słownych i fotograficznych wykorzystanych w podręczniku o zamiarze ich publikacji celem uzyskania od twórców lub ich następców prawnych wymaganej zgody za przysługującym wynagrodzeniem w stosownej wysokości. Osoby, których prawa w sposób niezawiniony przez wydawnictwo zostały naruszone, prosimy o bezpośredni kontakt z wydawcą.

Podczas opracowywania treści podręcznika skorzystano z następujących stron internetowych: <http://link.operon.pl/ww1>, <http://link.operon.pl/ww2>, <http://link.operon.pl/ww3>

Zastrzeżonych nazw firm użyto w tym podręczniku wyłącznie w celu identyfikacji.

Podręcznik dopuszczony do użytku szkolnego przez ministra właściwego do spraw oświaty i wychowania i wpisany do wykazu podręczników przeznaczonych do kształcenia ogólnego do nauczania informatyki w zakresie podstawowym na poziomie szkoły ponadpodstawowej, na podstawie opinii rzeczoznawców: dr. Tomasza Huka, dr. Przemysława Ogonowskiego, mgr Teresy Kosyry-Cieślak.

Etap edukacyjny: III
Typ szkoły: szkoła ponadpodstawowa
Rok dopuszczenia: 2021
Numer dopuszczenia: 1048/3/2021

© Copyright by OPERON Sp. z o.o. & Wojciech Hermanowski, Sławomir Sidor
Gdynia 2021

Wszelkie prawa zastrzeżone. Kopiowanie w całości lub we fragmentach bez zgody wydawcy zabronione.

N7348

Wydawca:
OPERON Sp. z o.o.
81-541 Gdynia, al. Zwycięstwa 96/98
tel. centrali 58 679 00 00
e-mail: info@operon.pl
www.operon.pl

ISBN: 978-83-8197-124-9

Zgodnie z decyzją MEN od września 2024 r. obowiązuje zmieniona, uszczuplona o ok. 20% podstawa programowa. W związku z tym w podręczniku znalazły się stosowne oznaczenia.

 Takim symbolem oznaczono treści usunięte z podstawy programowej.

 Takim symbolem oznaczono treści, które zostały zmienione. Nauczyciel przedstawi uczniowi zmiany i wyjaśni, czego powinien się nauczyć.

Spis treści

I. Rozwiązywanie problemów z wykorzystaniem komputera w nauce i firmie

1. Nowoczesna firma, czyli jak programy komputerowe ułatwiają pracę	10
2. Kalkulujemy, czyli jak wykorzystać arkusz kalkulacyjny w zarządzaniu finansami	15
3. Z sieci do tabeli, czyli jak interpretować dane za pomocą arkusza kalkulacyjnego	21
4. Spośród wielu, czyli filtrowanie w arkuszu kalkulacyjnym	29
5. Z eksperymentu do arkusza, czyli analiza danych z doświadczenia	37
6. Edytor grafiki w pracy zawodowej, czyli tworzymy reklamę	43
7. Reklama jest ważna, czyli jak wykonać atrakcyjną prezentację	53
8. Prezentacja wideo, czyli jak przygotować prezentację filmową	65
9. Multimedia w prezentacji, czyli dźwięk i film na slajdach	71
10. Skuteczne wsparcie, czyli jak przygotować pokaz prezentacji	78
11. Atrakcyjnie i wygodnie, czyli jak upowszechniać prezentację	87
12. Utrzymujemy kontakt z klientami, czyli korespondencja seryjna	95
Podsumowanie przed sprawdzianem	105

II. Linux i inne systemy operacyjne

13. Różne sposoby instalacji, czyli przygotowujemy miejsce dla systemu Linux	110
14. Wiele wersji, czyli wybieramy dystrybucję Linuxa	118
15. Bez kosztów, czyli programy w Linux	123
16. Tryb tekstowy jest ważny, czyli poznajemy konsolę Linux	128
17. Okienka nie są potrzebne, czyli używamy konsoli Linux	135
18. Komputer w kieszeni, czyli jak wykorzystać system Android w nauce i pracy	141
19. Był pierwszym z okienkami, czyli macOS i jego właściwości	147
Podsumowanie przed sprawdzianem	153

III. Programowanie i rozwiązywanie problemów za pomocą komputera

20. Przypomnij sobie, czyli wracamy do środowiska programistycznego	158
21. Przypomnij sobie, czyli podstawy języka programowania	165
22. Czy litery to liczby, czyli kod ASCII i porównanie tekstów	172
23. Metoda naiwna, czyli szukamy wzorca w tekście	180
24. Realizacja algorytmu, czyli program szuka wzorca	184
25. Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego	187
26. Zbiór Cantora, czyli najprostsze tworzenie fraktali	192
27. Drzewo binarne, czyli hierarchiczna struktura danych	196
28. Trójkątny fraktal, czyli trójkąt Sierpińskiego	201
29. Krzywa fraktalna, czyli płatek Kocha	205
Podsumowanie przed sprawdzianem	209



IV. Wykorzystanie algorytmów w rozwiązywaniu problemów i programowaniu

30. Jak to zapisać, czyli zamiana systemów liczbowych z ósemkowego na szesnastkowy ...	214	
31. Mniej lub więcej ósemek, czyli dodawanie i odejmowanie w systemie ósemkowym.....	219	
32. Iloczyn i iloraz oktalny, czyli mnożenie i dzielenie w systemie ósemkowym.....	224	
33. Nie tylko dwójkowy i dziesiętny, czyli suma i różnica w systemie innym niż dziesiętny ...	230	
34. Iloczyn i iloraz w systemie innym niż dziesiętny, czyli wielokrotności w systemach pozycyjnych	233	
35. Wyciąganie elementów z listy, czyli znajdowanie najdłuższego spójnego podciągu niemalejącego	237	
36. Elementy występujące kolejno po sobie, czyli znajdowanie najdłuższego spójnego podciągu o największej sumie.....	241	
37. Ta sama kolejność, czyli szukanie najdłuższego wspólnego podciągu	245	
38. Z której strony, czyli położenia punktów względem prostej.....	250	
39. Algorytmy badające własności geometryczne, czyli przynależności punktu do odcinka....	253	
40. Skrzyżowanie dróg, czyli badanie przecinania się odcinków	260	
41. Wewnątrz czy na zewnątrz, czyli badanie przynależności punktu do trójkąta	264	
42. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą prostokątów	268	
43. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą trapezów	273	
44. Znajdowanie elementów w zbiorze, czyli wyszukiwanie liniowe	276	
45. Szukanie połówek, czyli wyszukiwanie elementów poprzez połowienie	279	
46. Ustawianie kart, czyli sortowanie przez wstawianie.....	282	
47. Reguła falsi, czyli przybliżone rozwiązywanie równań.....	285	
48. Geometria obliczeniowa, czyli sprawdzanie przynależności punktu do wielokąta wypukłego.....	290	
49. Algorytm wyszukiujący, czyli liniowe przeszukiwanie ciągu w poszukiwaniu żadanego elementu z wykorzystaniem wartownika	294	
50. Podejście zachłanne, czyli problem plecakowy	297	
51. Z miasta A do miasta B, czyli szukanie najkrótszej drogi metodą zachłanną	300	
52. Pakowanie plecaka, czyli programowanie dynamiczne	303	
53. Metoda haszowania, czyli wyszukiwanie wzorca w tekście	305	
54. Metoda Monte Carlo, czyli symulacja ruchów Browna	308	
Podsumowanie przed sprawdzianem	312	

V. Komputer pomaga w nauce

55. Korzystamy z office.com, czyli jak wykorzystać aplikacje chmury w nauce	318
56. Nie tylko w firmie, czyli wykorzystanie aplikacji komunikacyjnych w nauczaniu	324
57. Każdy ma notatki, czyli jak wykorzystać chmurę do wspólnej nauki.....	334
58. Walidacja, czyli sprawdzamy wyniki swojej pracy	339
59. Matura, czyli jak komputery wspomagają przygotowanie do egzaminu	342
60. Rozwiązywanie testów pomaga w nauce, czyli jak aplikacje testują wiedzę.....	349
Podsumowanie przed sprawdzianem	355

Wstęp

Jak jest zbudowany nasz podręcznik

Zanim zaczniecie korzystać z podręcznika, zapoznajcie się z jego budową i przyjrzyjcie się wszystkim jego elementom. Zostały one skonstruowane tak, by ułatwić wam naukę i pomóc w błyskawicznym znalezieniu potrzebnych informacji.



Każdy rozdział podręcznika otwiera spis tematów zamieszczonych w tym rozdziale.

Bezpośrednio pod tytułem rozdziału znajdziecie moduł „Na tej lekcji”, w którym są wypunktowane cele danego tematu.

W tekście pojawiają się wytłuszczenia. Dzięki nim łatwiej odnajdziecie istotne zagadnienia, pojęcia i nazwy.



18.3. Pełna współpraca, czyli integrujemy telefon z Androidem z komputerem osobistym

Integracja systemów w skali miasta, kraju, a często i całego świata nie jest już niczym nadzwyczajnym. W ten sposób współpracują ze sobą systemy informatyczne banków, szpitali, urzędów państwowych itp. Skoro sprawdza się to w dużej skali, to i w małej, np. w zasięgu jednego użytkownika, powinno sprawdzać się doskonale. W systemie Windows 10 do połączenia wielu funkcji z telefonem służy aplikacja Twój telefon (rys. 18.3.). Dzięki niej jest możliwa wymiana danych i usług pomiędzy telefonem z Androidem a komputerem PC. Warunkiem działania takiego połączenia jest wspólny dostęp do sieci wi-fi.

Wczytywanie aplikacji (twój telefon)

Program Twój telefon sparowany z telefonem z systemem Android wyróżnia się takimi cechami jak:

- odbieranie i wysyłanie SMS,
- prowadzenie rozmów telefonicznych,
- dostęp do zdjęć,
- zarządzanie powiadomieniami,
- wymiana plików.

Twój telefon może połączyć się z telefonami z systemami Android lub iPhone.

Po połączeniu z telefonem możesz korzystać z jego funkcji w komputerze PC.

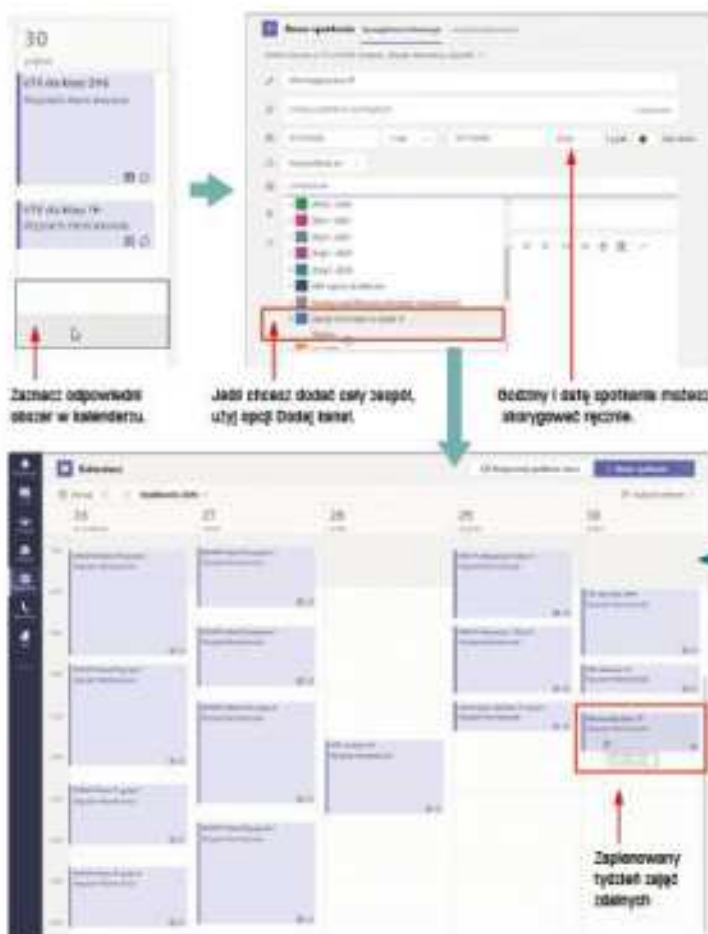


Rys. 18.3. Przykład wykorzystania aplikacji Twój telefon w systemie Windows 10

Przystępny język publikacji ułatwi wam zrozumienie treści.

Hasła na marginesach to treści obowiązkowe do zapamiętania z lekcji. Zostaną one powtórzone w podsumowaniu lekcji.

Schematy i rysunki pomogą wam usystematyzować wiedzę.



Rys. 26.10. Przykład tworzenia spotkania w Teams

Zrzuty ekranu pełnią funkcję tutoriali, dzięki którym będziecie mogli szybko je zastosować w praktyce.

Podsumowanie przed sprawdzianem zawiera najważniejsze treści do zapamiętania z danego rozdziału. Jeśli ich nie pamiętacie, wróćcie na stronę wskazaną obok i poszukajcie hasła na marginesie.

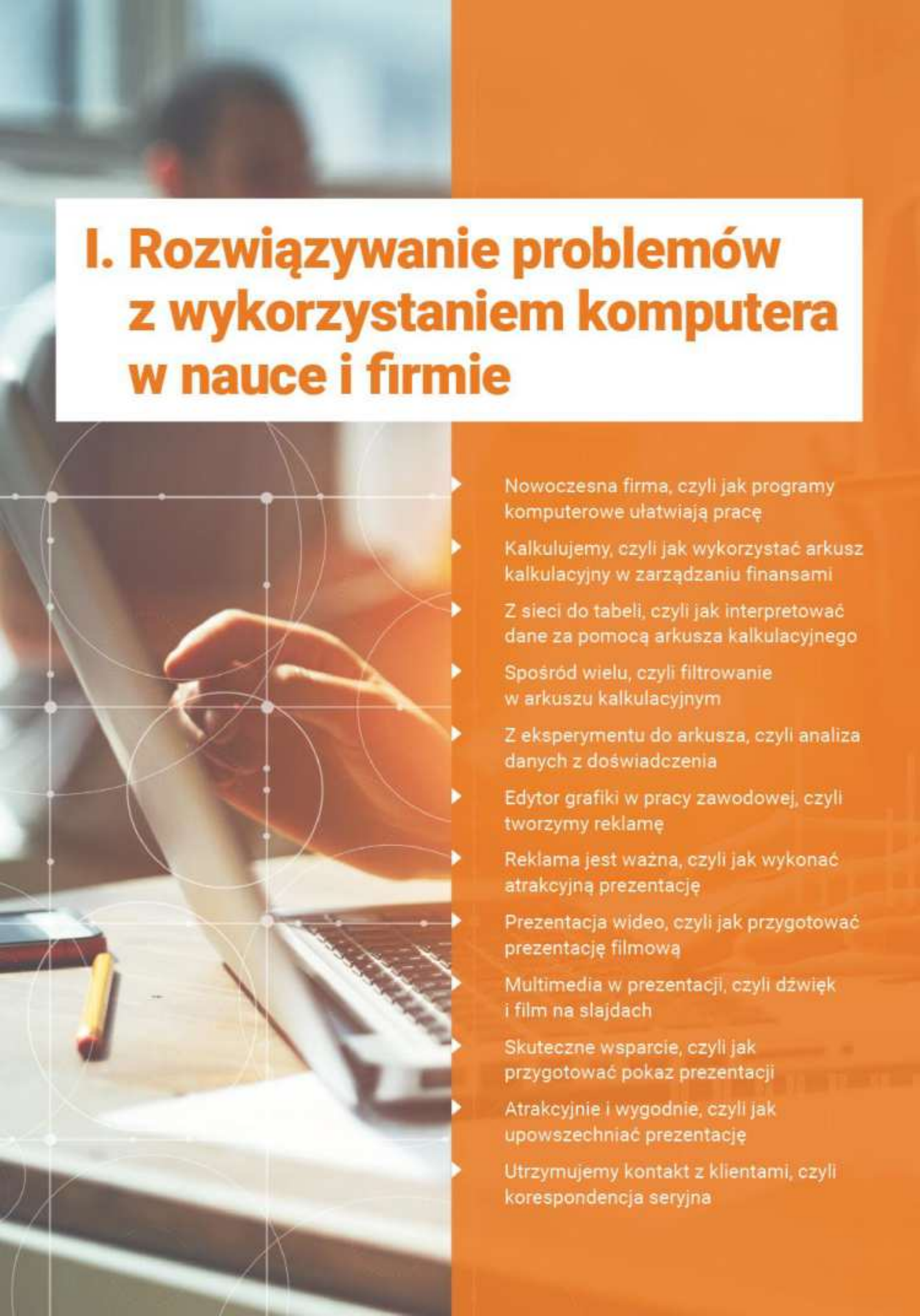
PODSUMOWANIE PRZED SPRAWDZIANEM	
20. Przypomnij sobie, czyli wracamy do środowiska programistycznego	
• Środowisko programistyczne Eclipse może być wykorzystane do programowania w języku C++.	s. 154
• Należy jednak pamiętać i zainstalować odpowiednią wersję, uwzględniającą także system operacyjny, w którym ma być zainstalowane.	s. 154
• Do kompilacji C++ w Eclipse są potrzebne narzędzia pozwalające na kompilację programów w środowisku Windows. Jednym z powszechnie stosowanych jest MinGW. Należy go zainstalować przed instalacją Eclipse dla C++.	s. 158
• Eclipse odjęty jest od strony domowej. Instalacja powinna automatycznie. Należy jednak pamiętać o wybraniu odpowiedniej wersji (dla C++) i zainstalować do folderu, w którym będą przechowywane projekty.	s. 159
• Eclipse ułatwia stosowanie pierwszych kroków w programowaniu w C++ między innymi poprzez przykładowe programy pokazujące poprawną strukturę. Dzięki temu można ćwiczyć kompilację i uruchamianie programów.	s. 160
• Dyskusyjny komentarz 4 i wykorzystanie przed kompilacją i powołanie odpowiednich bibliotek (biblioteki funkcji).	s. 160
• Ustawienie <code>newspawpaw std</code>	s. 160
To informacja dla kompilatora, że w programie zostanie użyty modułowa przestrzeń od. Przy składaniu danych programów, mogą zostać kodfity same, czyli różne obiekty mogą być nazwane tak samo. Można je wyznaczyć w różnych przestrzeniach i dzięki temu nie będą ze sobą kolidowały.	
• Program zaczyna działać od kodu zawartego w funkcji <code>main()</code> .	s. 160
• <code>cout << "Należy pamiętać, jak wyprowadzić komunikat";</code>	s. 160
21. Przypomnij sobie, czyli podstawy języka programowania	
• Poprawność kodu programu sprawdza debugger.	s. 162
• Kompilacja polega na dołączeniu modułów umożliwiających uruchomienie programu w systemie operacyjnym i utworzeniu pliku z kodem maszynowym zrozumiałym dla procesora.	s. 162
• Operatory relacyjne są używane w instrukcji warunkowej lub pętli.	s. 164
• W pętli <code>while</code> instrukcja jest wykonywana przynajmniej jeden raz bez względu na prawdziwość warunku.	s. 164
• Funkcja jest fragmentem programu, który nazwano i który jest wykorzystany poprzez podanie jego nazwy oraz ewentualnych argumentów (w nawiasie za nazwą).	s. 164
22. Czy litera to liczba, czyli kod ASCII i porównanie tekstów	
• Kod ASCII to liczby odpowiadające znakom drukowanym lub wydrukowanym. Każdemu znakowi przyporządkowane jest jedno miejsce.	s. 171
• W naszym świecie powstało wiele kodów ASCII zawierających znaki charakterystyczne dla danego języka, np. polskie <code>ą, ę, ś, ł</code> itp.	s. 171
• W systemie Windows istnieje się strona kodowa Windows-1250 CE, będąca odpowiednikiem ASCII dla języka Europejskiego Środkowo-Wschodniego.	s. 174

Zadania do wykonania przećwiczcie w toku lekcji lub wykorzystacie do nauki w domu albo przed kartkówką czy sprawdzianem.

UWAGA! W podręczniku nie należy umieszczać pisemnych odpowiedzi do zadań.

Podsumowanie lekcji to esencja najważniejszych treści do zapamiętania. Jeśli ich nie pamiętacie, wróćcie na stronę wskazaną obok. Znajdziecie tam na marginesie odpowiednie hasło, a w treści lekcji – jego omówienie.

3. MULTIMEDIA W PREZENTACJI, CZYLI DŹWIĘK I FILM NA SLAJACH	
	
Wybierz opcję Nagranie zawartości ekranu i kliknij Włączanie .	
Niewielkie okienko dialogowe zawiera wszystkie opcje potrzebne do nagrania.	
Rys. 9.9. Nagranie zawartości ekranu za pomocą narzędzia programu PowerPoint	
Prezentacje przygotowane z użyciem elementów multimedialnych może być bardzo atrakcyjna i znacznie wspomóc przekaz treści. Należy jednak pamiętać, że przygotowanie dla słuchacza prelekcji powinna wspierać prelegenta, a nie dominować.	
ZADANIA DO WYKONANIA	
1. Opracuj projekt i wykonaj prezentację na temat wykorzystania Audacity do nagrania głosu narzucając mu podziału pod tego rozdział na odpowiednich podtematach.	
2. Wykonaj prezentację zawierającą cztery slajdy i komentarze audio czytane na temat tematów i parametrów nagrania w programie Audacity.	
3. Wykonaj prezentację zawierającą filtry muzyczne w doci, odwołując się do, dotyczące nagrywania dźwięku za pomocą mikrofonu komputerowego, np. opcję Wzrost , Wzrost itp.	
4. Wykonaj prezentację dotyczącą się z jednego slajdu zawierającego tytuł trzech danych na ekranie nagrywanych podczas formatowania wideo w PowerPoint. Jeśli dysponujesz mikrofonem, zarejestruj również komentarz.	
5. Użyj opcji Nagranie dźwięku z programu PowerPoint i w ten sposób zarejestruj jedną z wykonanych przez siebie wcześniej prezentacji na dowolny temat.	
PODSUMOWANIE LEKCJI	
• Do rejestracji dźwięku można użyć narzędzi domowych wraz z systemem operacyjnym (np. Rejestrator głosu) lub narzędzi zewnętrznych (np. Audacity).	s. 78
• Program Rejestrator głosu rejestruje dźwięk z mikrofonu i zapisuje go w formacie <code>wav</code> .	s. 79
• Program Audacity może służyć do rejestracji dźwięku z dowolnych źródeł i miksowania, jak na diagramie schematu, edycji dźwięku.	s. 79
• Na wstępie Wstawianie w PowerPoint umożliwia opcje umieszczenia na slajdach dźwięków i filmów.	s. 79
• Można też umieszczać filmy z pliku (pobranie pliku prezentacji), z internetu, np. YouTube, lub skorzystać z rejestracji bezpośrednio z ekranu.	s. 79



I. Rozwiązywanie problemów z wykorzystaniem komputera w nauce i firmie

- ▶ Nowoczesna firma, czyli jak programy komputerowe ułatwiają pracę
- ▶ Kalkulujemy, czyli jak wykorzystać arkusz kalkulacyjny w zarządzaniu finansami
- ▶ Z sieci do tabeli, czyli jak interpretować dane za pomocą arkusza kalkulacyjnego
- ▶ Spośród wielu, czyli filtrowanie w arkuszu kalkulacyjnym
- ▶ Z eksperymentu do arkusza, czyli analiza danych z doświadczenia
- ▶ Edytor grafiki w pracy zawodowej, czyli tworzymy reklamę
- ▶ Reklama jest ważna, czyli jak wykonać atrakcyjną prezentację
- ▶ Prezentacja wideo, czyli jak przygotować prezentację filmową
- ▶ Multimedia w prezentacji, czyli dźwięk i film na slajdach
- ▶ Skuteczne wsparcie, czyli jak przygotować pokaz prezentacji
- ▶ Atrakcyjnie i wygodnie, czyli jak upowszechniać prezentację
- ▶ Utrzymujemy kontakt z klientami, czyli korespondencja seryjna



1. Nowoczesna firma, czyli jak programy komputerowe ułatwiają pracę

NA TEJ LEKCJI:

- poznasz przykłady wykorzystania programów komputerowych i chmury w różnych zawodach;
- dowiesz się, jakie korzyści przynosi stosowanie nowoczesnych rozwiązań informatycznych w przedsiębiorstwach.

Od wielu lat korzystasz z programów komputerowych i aplikacji. Z łatwością określisz ich przydatność w nauce i życiu codziennym. Zdajesz sobie jednak sprawę, że to tylko niewielki obszar ich zastosowań. Kontynuując naukę, a także podejmując działalność gospodarczą lub pracę zawodową, przekonasz się, że komputery są wszechobecne i w wielu przypadkach bez nich nie byłoby możliwe sprawne wykonanie wielu zadań. Wszystko, czego dowiesz się i nauczysz podczas korzystania z oprogramowania w trakcie lekcji, pozwoli ci na szybkie opanowanie nowych, zaawansowanych programów dla profesjonalistów.

1.1. Arkusz, czyli narzędzie uniwersalne

Arkusz kalkulacyjny, np. Excel, LibreOffice Calc lub podobny, może kojarzyć się z księgowością, obliczeniami i wykresami. Nic dziwnego, bo najczęściej właśnie takie ćwiczenia wykonuje się na lekcjach informatyki. Gdy jednak prześledzi się jego wykorzystanie w przedsiębiorstwach, okaże się, że zastosowań jest zdecydowanie więcej. Poniżej zapoznasz się z niektórymi z nich.

Cennik w arkuszu

Cennik usług lub towarów może się zmieniać w zależności od wielu czynników. Jeśli przygotujesz go do publikacji w edytorze tekstu, wprowadzanie zmian będzie pracochłonne, a przecież czas pracy pracownika jest cenny. Ponadto zmieniając poszczególne dane w edytorze, łatwiej popełnić błąd. Arkusz eliminuje te problemy, a ponadto pozwala na zbudowanie dokumentu, który w łatwy sposób można indywidualizować dla poszczególnych klientów. W poniższym ćwiczeniu zostaną użyte fikcyjne nazwy towarów, a ich liczba zostanie ograniczona.

Treść zadania:

Założmy, że dysponujesz siedmioma towarami. Niektóre z nich są objęte inną niż podstawowa stawką VAT (ich wysokość znajdziesz w sieci). Ponadto właściciel hurtowni docenia stałych klientów i przyznaje im rabaty (3%, 5%, 10%, 15% lub 20%). Utwórz arkusz, który pozwoli opracować indywidualny cennik dla danego klienta z uwzględnieniem różnych stawek VAT i przydzielonego rabatu. Użyj list



rozwijanych i stylu, w którym jest możliwe sortowanie według zawartości dowolnej kolumny. Twój arkusz może być zbudowany tak jak na rysunku 1.1.

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							

Rys. 1.1. Przykładowy arkusz z danymi

W wykonaniu zadania pomoże ci poniższy opis (rys. 1.2. i rys. 1.3.) lub system pomocy arkusza.

Przygotuj wybrane komórki arkusza.

Formatowanie jako tabeli ? X

Gdzie znajdują się dane do tabeli?

☒ Moja tabela ma nagłówki

OK Anuluj

Zaznacz wszystkie komórki przyszłej tabeli cennika i wybierz opcję **Formatuj jako tabelę** z wstęgi **Narzędzia główne**. Potwierdź zakres twojej tabeli.

Wybierz **Styl**.

Operacja sortowania będzie dostępna w rozwijanym menu.

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

	A	B	C	D	E	F	G
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							

Rys. 1.2. Tworzenie przykładowej tabeli cennika z opcją sortowania według kolumn



Gdy uporasz się z budowaniem tabeli, możesz przystąpić do tworzenia list rozwijanych. Skorzystaj z opisu w opcji **Pomoc**. Poprowadzi cię prosto do celu (rys. 1.3.).

The screenshot shows the Excel Help window with the search bar containing 'lista rozwijana'. The results list includes 'Tworzenie listy rozwijanej', 'Dodawanie lub usuwanie elementów z listy rozwijanej', 'Stosowanie sprawdzania poprawności danych do komórek', 'Wstawianie bieżącej daty i godziny do komórki', and 'Klip wideo: Zarządzanie listą adresów w programie Excel'. A green arrow points from the first result to a table example. A red arrow points from the search bar to the text 'W opcji pomocy wpisz „lista rozwijana” i skorzystaj z opisu.'.

W opcji pomocy wpisz „lista rozwijana” i skorzystaj z opisu.

Asortyment	Cena katalogowa	Rabat
Towar 1		
Towar 2		23
Towar 3		8
Towar 4		4
Towar 5		0
Towar 6		
Towar 7		

Tak powinna wyglądać lista rozwijana z przykładowego cennika.

Rys. 1.3. Wykorzystanie opcji **Pomoc** do tworzenia listy rozwijanej

Tak przygotowany cennik może służyć do różnych operacji, np. do szybkiego przygotowania oferty dla konkretnego klienta, szybkiej zmiany asortymentu z uwzględnieniem odpowiednich stawek VAT itp. Na jego podstawie można także wystawiać faktury – oczywiście za pomocą arkusza.

1.2. Nie tylko PowerPoint, czyli wykorzystanie prezentacji w firmie

Programy do tworzenia prezentacji złożonych z tekstów i elementów multimedialnych od dawna są podstawowym narzędziem w pracy wielu firm. Obecnie edytory prezentacji działają zarówno w chmurach, jak i lokalnie. Różnią się możliwościami narzędzi, sposobem publikacji, formatami zapisu itp. Jaki edytor wybrać? Na takie pytanie zawsze jest jedna odpowiedź – to zależy od wymagań użytkownika.

Popularne, rozbudowane pakiety biurowe – Microsoft Office lub Libre Office – zawierają programy do tworzenia prezentacji, które wzajemnie akceptują swoje formaty. Jest to wygodne, ale należy pamiętać, że importowane prezentacje mogą być wyświetlane nieco inaczej niż w oryginale. Należy więc zawsze sprawdzić to przed ich uruchomieniem. Chmury informatyczne także pozwalają na wykonywanie ciekawych prezentacji, np. w chmurze Office 365 umieszczono edytor podobny do odpowiednika off-line (aplikacja Sway). Wiele użytkowników korzysta także z chmur, które oferują aplikacje on-line służące do tworzenia bardzo atrakcyjnych prezentacji. Znajdziecie je bez problemu. Zwróćcie jednak uwagę na warunki korzystania z tych chmur.



W firmach prezentacje stanowią bardzo skuteczne wsparcie w szkoleniach czy przedstawianiu ofert. Są jednak także wykorzystywane do przekazywania instrukcji, regulaminów, opisów czynności, tutoriali itp.

1.3. Porządek w sekretariacie, czyli zarządzanie korespondencją i kontaktami

Każdy, kto prowadzi korespondencję z wieloma adresatami, wie, jak łatwo zagubić lub pominąć ważny list lub mail. Problemem jest także odnajdowanie danych adresowych. Oczywiście zdajecie sobie sprawę, że wielką w tym pomocą są programy komputerowe. Nazywamy je **klientami poczty**.

Niektóre działają on-line i są aplikacją uruchamianą w przeglądarce stron internetowych. Inne należy zainstalować w postaci aplikacji. Część takich programów potrafi synchronizować swoją pracę, a wtedy zawartość skrzynki i dane są aktualizowane na bieżąco. Ważniejsze cechy klientów poczty przedstawiono w tabeli (tab. 1.1.).

klient poczty

Lp.	Klient poczty off-line	Klient poczty on-line
1.	Można przeglądać aktualną pocztę bez konieczności łączenia się z siecią i logowania do konta pocztowego.	Przeglądanie poczty wyłącznie po zalogowaniu się do konta pocztowego.
2.	Przeglądanie odebranej wcześniej poczty nie wymaga logowania się do konta pocztowego.	Dostęp do archiwów poczty wymaga logowania się do konta pocztowego.
3.	Dostęp do aplikacji może być zabezpieczony dodatkowym hasłem.	Dostęp do aplikacji wymaga logowania do konta pocztowego lub chmury programu.
4.	Wymaga aktualizacji.	Aktualizacja przebiega automatycznie (użytkownik zostaje powiadomiony o zmianach), bez udziału użytkownika.
5.	Niektóre aplikacje wspomagają szyfrowanie korespondencji (mają odpowiednią opcję).	
6.	Istnieje możliwość zarządzania korespondencją przez tworzenie folderów, zakładek itp.	
7.	Można zarządzać kontaktami. Liczba informacji zapisywanych w bazie zależy od danej aplikacji.	
8.	Sortowanie e-maili jest możliwe według różnych kryteriów.	
9.	Możliwa jest synchronizacja pomiędzy wersjami.	

Tab. 1.1. Niektóre cechy klientów poczty



1.4. Kalendarze i harmonogramy, czyli zarządzanie czasem

Edytory i arkusze pozwalają na tworzenie dokumentów. Nie mniej ważne są także programy, których głównym zadaniem jest wspomaganie organizacji pracy, zarządzanie czasem i zadaniami pracowników.

kalendarz

Zadania na poszczególne dni i godziny pracy mogą być wpisywane przez przełożonego bezpośrednio do kalendarza pracownika. Ten ma do niego dostęp on-line na komputerze lub telefonie. Kalendarz pełni także inne funkcje. Może być bezpośrednim odnośnikiem do spotkań i narad on-line, tak jak program Teams z pakietu Office 365. Ponieważ w opisach wydarzeń można umieszczać wszelkie odnośniki i dokumenty, pracownik nie musi szukać potrzebnych materiałów do wykonania zadania. Chmura Google także oferuje podobne rozwiązania. Z kalendarzem możesz mieć również do czynienia, biorąc udział w zdalnym nauczaniu lub kursach on-line. Wyznaczają one terminy zaliczeń, wykonania zadań, spotkań z wykładowcami itp. Poznana wcześniej platforma Moodle jest przykładem dobrego zastosowania narzędzia Kalendarz.

W następnych rozdziałach poznasz więcej szczegółów dotyczących wykorzystania pakietów biurowych i chmur w nauce i w pracy firmy.

ZADANIA DO WYKONANIA

1. Po wykonaniu ćwiczenia z początku rozdziału uzupełnij projekt cennika o obliczanie kwot w zależności od liczby sztuk zamówionych towarów. Pracownik dzięki takiej modyfikacji powinien móc sprawnie obliczyć, ile klient będzie musiał zapłacić za swoje ewentualne zamówienie.
2. Czy w twoim komputerze jest zainstalowany klient poczty? Jeśli tak, to przedstaw krótko jego właściwości. Wykorzystaj swoje doświadczenie, system pomocy lub publikacje. Wymień jego zalety.
3. Skorzystaj z wyszukiwarki w celu znalezienia darmowych klientów poczty instalowanych w systemie komputera. Wybierz trzy, które twoim zdaniem są najlepsze. Porównaj je w tabeli. Uwzględnij funkcjonalność, liczbę przydatnych narzędzi i inne cechy. Skonfrontuj swoje obserwacje z tabelą 1.1.
4. Skorzystaj z wyszukiwarki w celu znalezienia chmur lub usług internetowych z aplikacjami do tworzenia prezentacji on-line. Porównaj ich właściwości i narzędzia. Wybierz twoim zdaniem najlepszą z nich i przedstaw ją w prezentacji programu PowerPoint lub LibreOffice Impress.
5. Porównaj program do tworzenia prezentacji w chmurze Google z jego odpowiednikiem LibreOffice Impress. Wymień różnice i podobieństwa. Sprawdź, czy pliki utworzone w obu programach są tego samego formatu.

PODSUMOWANIE LEKCJI

s. 13

- **Klient poczty** to program do zarządzania korespondencją elektroniczną. Pozwala także na zapamiętywanie niektórych danych adresowych.

s. 14

- **Kalendarze komputerowe**, pracujące jako oddzielne aplikacje lub będące częścią pakietów, pozwalają na skuteczne zarządzanie czasem pracowników lub własnym, np. podczas przygotowań do matury.



2. Kalkulujemy, czyli jak wykorzystać arkusz kalkulacyjny w zarządzaniu finansami

NA TEJ LEKCJI:

- wykonasz kalkulator obliczający wysokość VAT oraz kwoty brutto i netto;
- wykorzystasz arkusz do kalkulacji kosztów wytworzenia wyrobu lub świadczenia usługi.

2.1. Brutto, netto, VAT, czyli arkusz kalkulacyjny oblicza podatki

W sieci znajduje się sporo stron, na których można wykonywać różne obliczenia potrzebne w działalności zawodowej. Można jednak zastąpić je własnym arkuszem kalkulacyjnym. Wystarczy dobry projekt i znajomość układania i wykorzystania formuł.

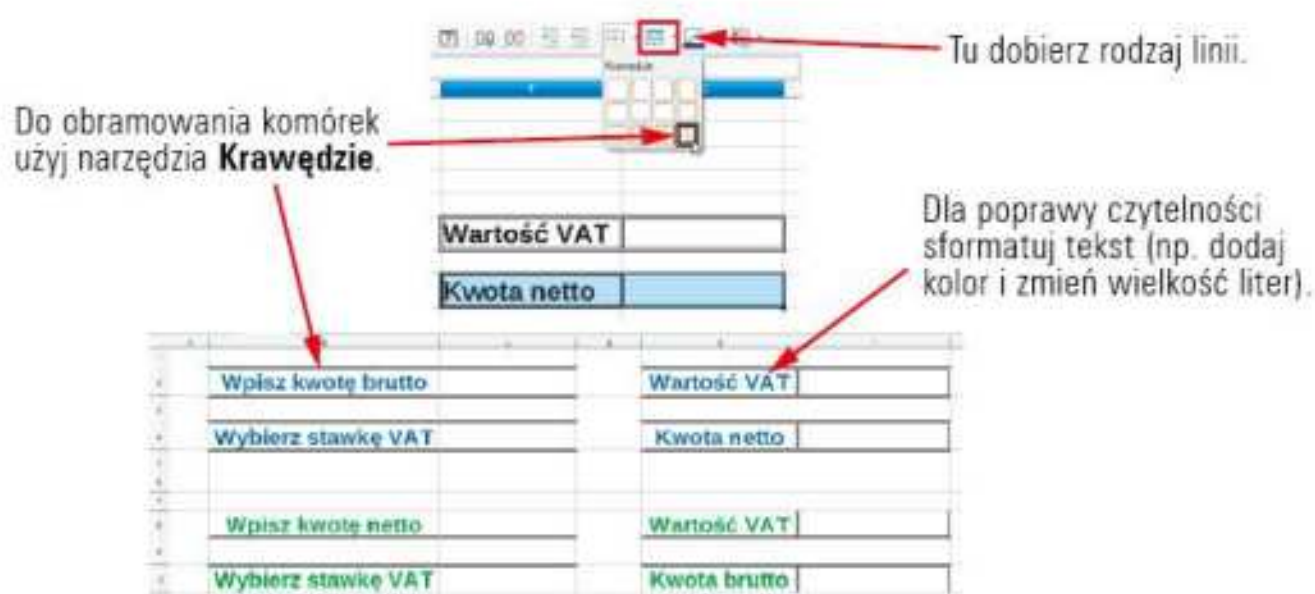
Założmy, że chcemy mieć narzędzie obliczające ceny netto, brutto i podatek VAT. Jest on naliczany według różnych stawek – 23%, 8%, 7%, 5% i 4% – w zależności od odpowiednich przepisów. Stawki mogą się zmieniać, dlatego arkusz powinien być tak zaprojektowany, by zmiany można było wprowadzić w prosty sposób. Jeśli zastosujesz odpowiednie mechanizmy i wykorzystasz możliwości arkusza, stworzony kalkulator stanie się uniwersalnym narzędziem pomocnym w pracy.

projektowanie
arkusza

W pierwszym przykładzie wykorzystano arkusz kalkulacyjny **Calc** z darmowego pakietu **LibreOffice**. Jest on dostępny dla **Windows** (wymagana instalacja) i w **Linux**, w którym zazwyczaj jest jednym z programów zainstalowanych wraz z systemem. Drugi przykład pokazuje rozwiązanie w Excelu.

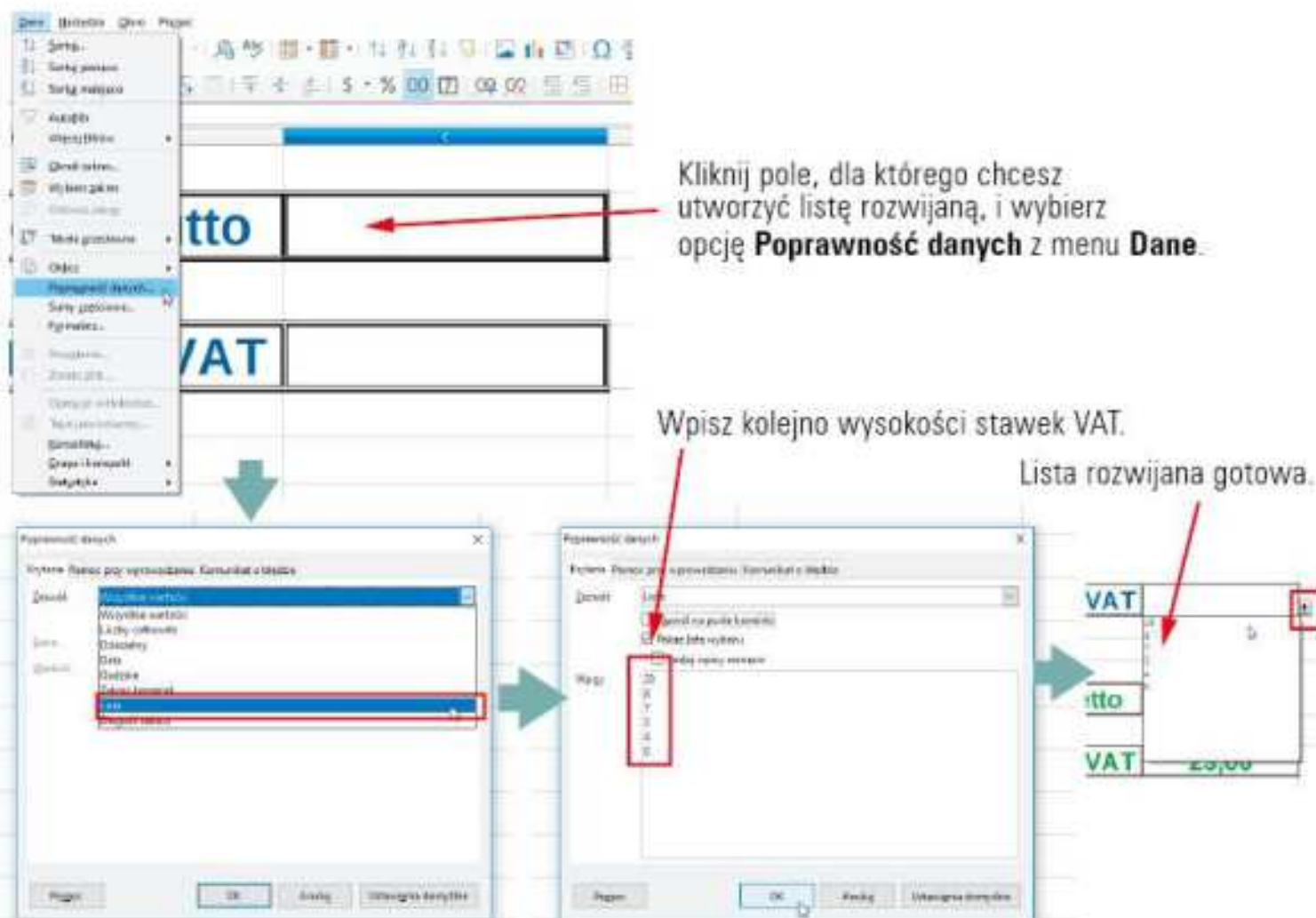
Opisany przykład zawiera wybór metod i narzędzi, które pozwolą na tworzenie podobnych arkuszy o innym przeznaczeniu.

Zacznij od przygotowania komórek dla danych i wyników. Zaplanuj ich ułożenie i wygląd (rys. 2.1.).



Rys. 2.1. Przykładowy rozkład komórek dla kalkulatora VAT w programie LibreOffice Calc

Zanim przystąpisz do układania formuł, zorganizuj wprowadzanie danych. Czy wszystkie dane musisz wprowadzać za każdym razem, gdy kalkulator zostanie użyty? Jak rozwiązać przypadek powtarzalnych danych, np. wartości stawek VAT? Są przecież niezmiennie w danym roku rozliczeniowym. W celu umożliwienia użytkownikowi wyboru wysokości stawki bez konieczności jej wpisywania, można użyć list rozwijanych (rys. 2.2.). Umiesz już to zrobić w MS Excel. Tym razem wykonasz to w LibreOffice Calc. Umiejętność ich tworzenia przyda się w kolejnych projektach. Arkusz stanie się łatwiejszy w obsłudze.



Rys. 2.2. Tworzenie listy rozwijanej w LibreOffice Calc



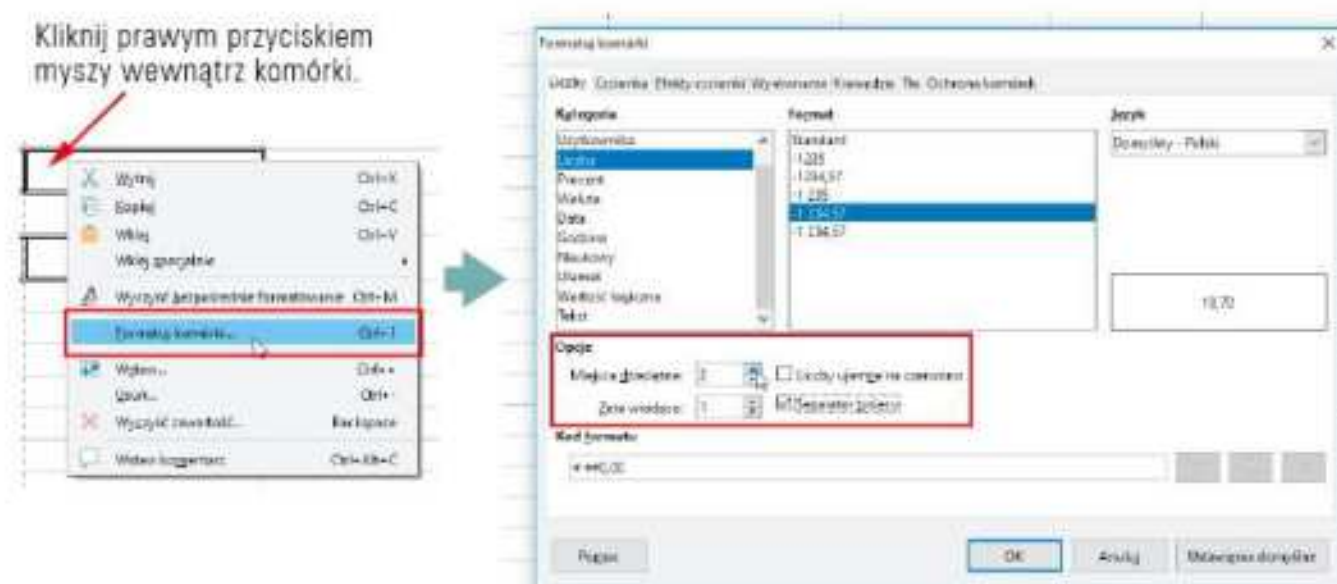
Lista rozwinie się po kliknięciu strzałki, która pokazała się po prawej stronie komórki. Wybrana wartość będzie stanowiła zawartość komórki. W przykładzie tę samą czynność wykonano dla komórek C4 i C10.

Czas na przygotowanie formuł. Wzory na wyliczenie zawartości VAT w kwocie brutto i obliczenie wartości VAT od kwoty netto znajdziesz w internecie. W tabeli ujęto ich zapis w postaci formuły arkusza.

Komórka	Formuła	Działanie
F2	$=C2 * C4 / (100 + C4)$	Zawartość VAT w kwocie brutto
F4	$=C2 - F2$	Kwota netto po odliczeniu VAT
F8	$=C8 * C10 / 100$	VAT dla kwoty netto
F10	$=C8 + F8$	Kwota brutto po dodaniu VAT

Tab. 2.1. Formuły zastosowane do obliczeń w przykładowym kalkulatorze VAT

Dla kwot w arkuszu należy wprowadzić ograniczenie liczby miejsc po przecinku do dwóch. W LibreOffice Calc robi się to podobnie jak w Excelu (rys. 2.3.).



Rys. 2.3. Ustawienie liczby miejsc dziesiętnych w liczbie z danej komórki

Pozostaje już tylko sprawdzić działanie arkusza, wprowadzając do komórki C8 obliczoną dla wprowadzonej kwoty brutto (C2) wartość z komórki F4 (rys. 2.4.). Wyniki działania formuł powinny się pokrywać.

1					
2	Wpisz kwotę brutto	100,00	Wartość VAT	18,70	
3					
4	Wybierz stawkę VAT	23,00	Kwota netto	81,30	
5					
6					
7					
8	Wpisz kwotę netto	81,30	Wartość VAT	18,70	
9					
10	Wybierz stawkę VAT	23,00	Kwota brutto	100,00	

Rys. 2.4. Sprawdzenie działania kalkulatora



2.2. Symulacje kosztów, czyli arkusz pomaga w planowaniu wydatków

Biegłość w obsłudze arkuszy jest pomocna przy planowaniu wykonania jakiejś usługi lub w sytuacjach domowych, np. wyliczeniu kosztów remontu. Przed przystąpieniem do robót warto wiedzieć, ile pieniędzy należy przeznaczyć na zakup materiałów i usług. Ponieważ cenniki nie są identyczne u każdego kontrahenta lub sprzedawcy, przydatne jest narzędzie do racjonalnej kalkulacji tych wydatków. Arkusz kalkulacyjny to idealne narzędzie do sporządzenia takich obliczeń. Po wykonaniu poniższego ćwiczenia bez trudu opracujesz arkusz dostosowany do swoich potrzeb.

Twoim zadaniem jest wykonanie remontu w pokoju. Przed przystąpieniem do prac należy wykonać kosztorys. Dla uproszczenia przykładu przyjmijmy, że kupujesz jedynie farby w pojemnikach po 10 i 5 litrów. Arkusz powinien obliczyć ilość potrzebnej farby i optymalną liczbę pojemników o danej pojemności. W swoich arkuszach możesz uwzględnić także inne materiały lub usługi.

Do tych danych możesz użyć listy rozwijanej.

Powierzchnia	Wydajność z 1 litra	Liczba malowań
100	13	2
8		2
9		2
10		3
11		3
12		3
13		3
14		3
15		3

Wydajność z 1 litra	Liczba malowań	Ile farby potrzebujesz?
13	2	15,4
10	2	20,0
10	3	30,0

Przeznaczenie farby	Powierzchnia	Wydajność z 1 litra	Liczba malowań	Ile litrów farby potrzebujesz?	Liczba pojemników 10 l	Liczba pojemników 5 l	Cena pojemnika 10 l	Cena pojemnika 5 l	Koszt farby
Sufit	100	13	2	15,4	1	2	42	25	92
Ściany kolor 1	100	10	2	20,0	2	0	45	30	90
Ściany kolor 2	100	10	3	30,0	3	0	89	47	267
SUMA:									449

$=E6 * C6 / D6$
 $=JEŻELI(G6 * 10 - F6 < > 0; ZAOKR.W.GÓRĘ(((F6 - G6 * 10) / 5); 1); 0)$
 $=JEŻELI(F6 < 10; 0; LICZBA.CAŁK(F6 / 10))$

Rys. 2.5. Przykładowy arkusz do kalkulacji zakupu materiałów do usługi z wpisanymi danymi

Komórka	Formuła
F6	=E6*C6/D6
G6	=JEŻELI(F6<10;0;LICZBA.CAŁK(F6/10))
H6	=JEŻELI(G6*10-F6<>0;ZAOKR.W.GÓRĘ(((F6-G6*10)/5);1);0)

Tab. 2.2. Formuły użyte w przykładzie arkusza kalkulacji farb



Formuła dla komórek F6, F7 i F8 jest łatwa do ułożenia. Trzeba pomnożyć liczbę warstw przez powierzchnię do malowania i podzielić przez wydajność odczytaną z pojemnika, a otrzyma się ilość potrzebnej farby.

Nieco trudniej ułożyć formułę, która wyliczy liczbę pojemników o danej objętości.

Liczba pojemników 10-litrowych jest obliczana w pierwszej kolejności, ponieważ większe gabaryty opakowań są zazwyczaj tańsze. Ponadto jeśli zostanie pewna ilość farby, lepiej, by była ona częścią mniejszego pojemnika.

Formuła dla komórki G6 zawiera badanie warunku. W uruchomionym Excel lub Calc można użyć opcji pomocy. Wystarczy wcisnąć klawisz F1 i wpisać słowo „JEŻELI”, a otrzymamy definicję takiej funkcji i sposób jej użycia (rys. 2.6. i rys. 2.7.). Przeczytaj je uważnie.

Wyniki dla „jeżeli”

JEŻELI, funkcja

Funkcja JEŻELI w programie Excel zwraca jedną wartość, jeśli warunek jest prawdziwy, lub inną wartość, jeśli jest fałszywy. Wewnątrz funkcji JEŻELI można użyć maksymalnie 64 dodatkowych funkcji JEŻELI. Program Excel zawiera też inne funkcje, których można używać do analizowania danych na podstawie warunku, takie jak funkcje arkusza LICZ.JEŻELI lub LICZ.WARUNKI.

Rys. 2.6. Definicja funkcji **JEŻELI** podana przez opcję **Pomoc** arkusza Excel

JEŻELI

Wykonuje test logiczny.

Składnia

JEŻELI (Test; Wtedy_wartość; W_przeciwnym_razie_wartość)

Test jest wyrażeniem mogącym przybierać wartość PRAWDA lub FAŁSZ.

Wtedy_wartość (parametr opcjonalny) jest wartością zwracaną przez funkcję, jeśli Test zwraca PRAWDA.

W_przeciwnym_razie_wartość (parametr opcjonalny) jest wartością zwracaną przez funkcję, jeśli Test zwraca FAŁSZ.

W funkcjach programu LibreOffice można nie podawać parametrów oznaczonych jako "opcjonalnych", tylko jeśli za nimi nie występują inne parametry. Na przykład w funkcji o czterech parametrach, w której dwa ostatnie parametry są oznaczone jako "opcjonalne", można nie podawać parametru 4 lub parametrów 3 i 4, ale nie można nie podawać tylko parametru 3.

Przykłady

=JEŻELI (A1>5;100;"za mało") Jeśli wartość w komórce A1 jest większa niż 5, do bieżącej komórki zostanie wprowadzona wartość 100. W przeciwnym przypadku do komórki zostanie wprowadzony opis "za mało" (bez cudzysłowów).

Rys. 2.7. Definicja funkcji **JEŻELI** podana przez opcję **Pomoc** arkusza LibreOffice Calc

Zaznajomieni z tymi definicjami przeanalizujmy formułę dla G6 (rys. 2.8.).

=JEŻELI (F6<10;0;LICZBA.CAŁK (F6/10))

Jeżeli F6, czyli liczba potrzebnych litrów danej farby jest mniejsza od 10, to liczba potrzebnych pojemników z farbą o pojemności 10 l wynosi 0. Tyle wpisz do G6.

W przeciwnym przypadku do G6 wpisz część całkowitą z dzielenia ilości potrzebnej farby przez objętość 10 l.

Rys. 2.8. Formuła obliczająca liczbę 10-litrowych pojemników z farbą



Zazwyczaj potrzebna ilość farby nie jest wielokrotnością 10. Kolejna formuła powinna więc obliczyć, ile powinniśmy dokupić pojemników 5-litrowych. Powinna ona także dać wynik 1 dla przypadku, w którym pozostało do zakupu mniej niż 5 litrów (rys. 2.9.).

=JEŻELI (G6*10-F6<>0;ZAOKR.W.GÓRĘ((F6-G6*10)/5);1);0)

Jeżeli ilość farby w pojemnikach 10-litrowych nie jest równa potrzebnej, czyli $G6*10-F6 \neq 0$, do H6 wpisz liczbę potrzebnych pojemników 5-litrowych.

Liczba pojemników 5-litrowych. Dzięki zaokrągleniu w górę otrzymasz liczbę potrzebnych pojemników, nawet gdy farby z jednego z nich nie wykorzystasz w całości. „1” po średniku oznacza liczbę cyfr zaokrąglenia.

W przeciwnym przypadku, czyli jeśli potrzebna farba w całości mieści się w pojemnikach 10-litrowych, do H6 wpisz 0.

Rys. 2.9. Formuła obliczająca liczbę potrzebnych pojemników 5-litrowych

Kwotę, jaką musisz przygotować na zakup farby, obliczysz, mnożąc cenę poszczególnych pojemników przez ich liczbę wyliczoną w arkuszu. Tabele można powielić przez schowek i wprowadzać dane dla różnych producentów farb lub sklepów. Uwaga! LibreOffice Calc importuje arkusze z programu Excel. Także Excel importuje arkusze z Calc. Jak widzisz, tworzenie prostych arkuszy służących symulacji nie jest trudne. Listy rozwijane pomagają w szybkim wypełnianiu tabel i ograniczają ryzyko pomyłki. Oczywiście symulacje w arkuszu nie muszą dotyczyć finansów. Wszystko zależy od potrzeb i umiejętności autora. Zapewne nieraz spotkacie się z problemami, w których rozwiązywaniu arkusz będzie pomocny. Warto zapamiętać, że to narzędzie nie tylko dla księgowych.

ZADANIA DO WYKONANIA

1. Na podstawie arkusza obliczającego VAT, kwotę netto i brutto zaprojektuj i wykonaj arkusz do obliczania tych wartości dla zakupów do 20 towarów lub usług.
2. Użyj funkcji JEŻELI do utworzenia arkusza, który w komórce B5 wyświetli komunikat czerwony: „Przekroczono limit wydatków” w przypadku, gdy kwota wpisana do komórki B2 przekroczy kwotę z komórki D2.
3. Zmodyfikuj arkusz z zadania 2 w taki sposób, by limit wydatków był wybierany z listy rozwijanej. Uwzględnij kwoty: 1000, 2000, 5000, 10 000, 20 000 i 50 000 złotych.
- 4*. Zaprojektuj i wykonaj arkusz obliczający twój zarobek na podstawie kwoty brutto zapisanej w umowie o pracę. Uwzględnij aktualne koszty pracy. Obowiązujące stawki znajdź w internecie.
5. Wykorzystaj formułę warunkową JEŻELI do utworzenia tłumacza ocen szkolnych. Arkusz powinien zamieniać oceny cyfrowe na wyrażone słownie, np. 4 – dobry.

PODSUMOWANIE LEKCJI

- Arkusz należy tak **projektować**, aby zmiany można było w nim wprowadzić w prosty i szybki sposób.



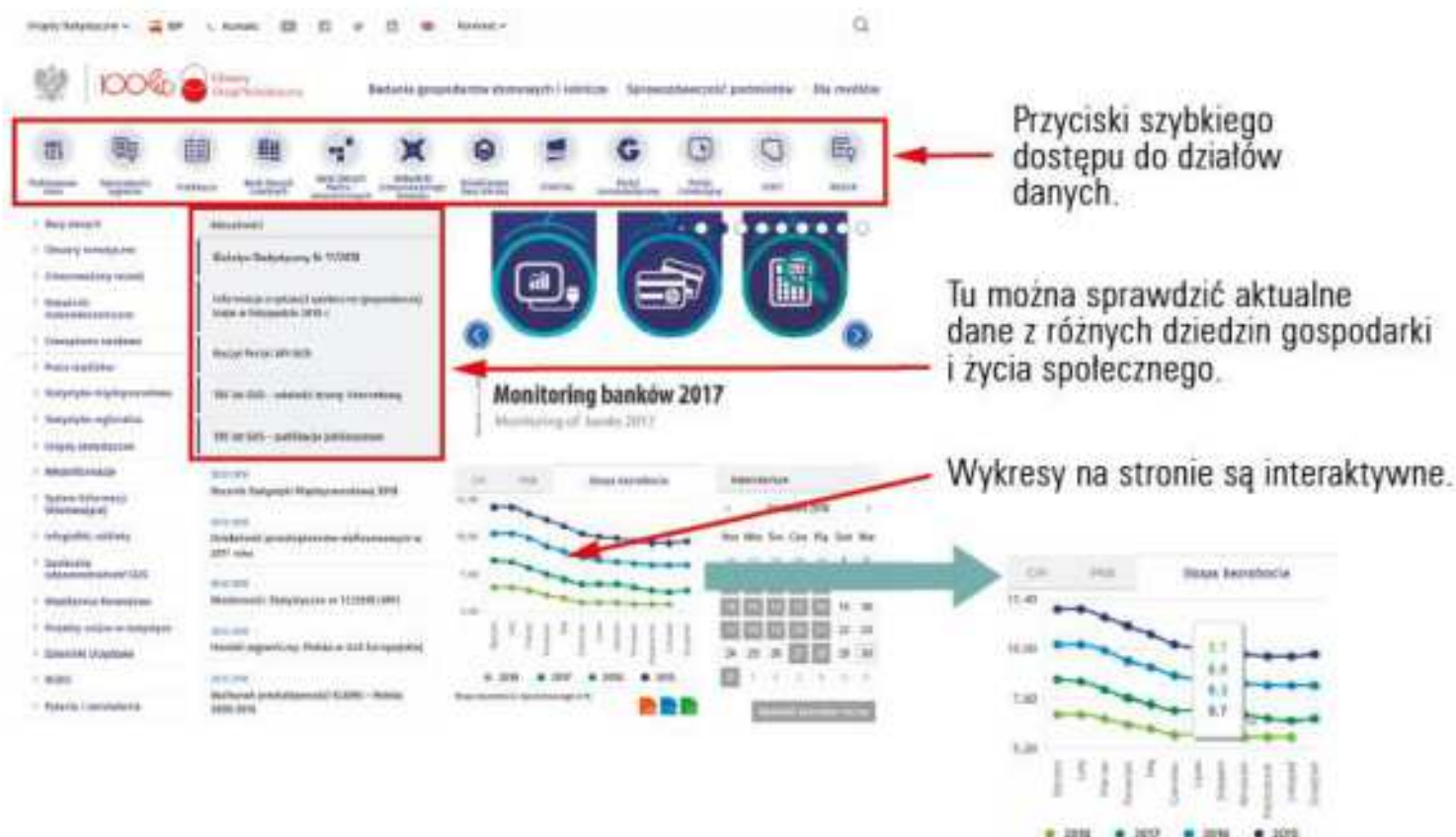
3. Z sieci do tabeli, czyli jak interpretować dane za pomocą arkusza kalkulacyjnego

NA TEJ LEKCJI:

- wyselekcjonujesz i przeniesiesz do arkusza dane z tabel pochodzących ze strony internetowej i dobierzesz sposób ich wizualizacji;
- przeniesiesz do arkusza i przygotujesz do analizy dane rankingu.

3.1. Wyszukane, importowane, czyli jak odnaleźć odpowiednie dane i wpisać do arkusza

W różnych dziedzinach działalności zawodowej są potrzebne precyzyjne dane. Można ich szukać w rocznikach statystycznych, kupić je w firmach badających rynek, a także znaleźć w internecie. Informacja jest towarem. Za bardzo precyzyjne i aktualne dane trzeba zapłacić. Bardzo pomocne mogą również okazać się niektóre informacje i statystyki dostępne w sieci. Przykładem mogą być dane umieszczone na stronie Głównego Urzędu Statystycznego (rys. 3.1.). Dane z rysunków mogą być już nieaktualne.



Rys. 3.1. Główny Urząd Statystyczny, stat.gov.pl



Urząd podaje zgromadzone dane i nie prowadzi badań opinii.

Sprawdźmy aktualne informacje o sytuacji gospodarczej w kraju i pobierzmy je bezpośrednio ze strony GUS-u (rys. 3.2.).

Jeśli chcesz sięgnąć do danych archiwalnych, kliknij **Archiwum**.

Kliknij.

Informacja o sytuacji społeczno-gospodarczej kraju w listopadzie 2018 r.

Wybrane wskaźniki sytuacji społeczno-gospodarczej kraju. Tablice w formacie XLSX

Podstawowe wskaźniki makroekonomiczne UE i OECD. Tablice w formacie XLSX

Otwieranie wybranych wskaźników sytuacji społeczno-gospodarczej kraju...

Rozpoznano pobierany plik:

Wybrany wskaźniki sytuacji społeczno-gospodarczej kraju Listopad_2018.xlsx

Typ pliku: Arkusz programu Microsoft Office Excel (24,3 KB)

Adres: <http://stat.gov.pl>

Po zakończeniu pobierania:

☒ Otwórz za pomocą Microsoft Office Excel (domyślnie)

☐ Zapisz plik

☒ Pamiętaj tę decyzję dla wszystkich plików tego typu

Ustawienia można zmienić w karcie Pliki w oknie Opcje programu Firefox.

Wybrane wskaźniki sytuacji społeczno-gospodarczej kraju	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Wykaz ogólny																				
4 - 2017																				
5 - 2018																				
Pozycje zatrudnienia w sektorze przedsiębiorstw w tys. osób	4	5060	5076	5082	5081	5081	5082	5082	5082	5082	5082	5082	5082	5082	5082	5082	5082	5082	5082	5082
Garnitur zapleceniowy (stan w końcu okresu) w tys. osób	5	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167	1167
Stopa bezrobocia (stan w końcu okresu) w %	6	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8	8,8
Pozycje miesięczne wynagrodzenia brutto w sektorze przedsiębiorstw w złotych	7	4277,32	4334,95	4371,86	4488,07	4560,94	4596,06	4591,52	4492,40	4473,06	4574,35	4616,79	4673,73							
średni miesięczny	8	4588,58	4592,72	4588,58	4584,44	4588,52	4584,16	4582,52	4708,27	4771,80	4821,50	4866,91								
indeks	9	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0	100,0							
analogiczny miesiąc poprzedniego roku	10	104,1	104,0	103,2	104,1	103,4	103,0	104,9	105,0	104,9	105,0	107,4	106,3							
Pozycje miesięczne wynagrodzenia netto brutto w sektorze przedsiębiorstw - analogiczny miesiąc poprzedniego roku	11	107,3	106,8	106,7	107,8	107,8	107,8	107,8	107,8	107,8	106,8	106,7	107,8							
analogiczny miesiąc poprzedniego roku	12	102,4	101,8	102,1	102,1	102,4	104,9	103,2	104,8	103,9	105,4	104,1	106,3							
	13	106,9	106,5	106,8	106,2	105,3	105,8	105,1	104,8	104,7	106,7	106,3	106,3							

Rys. 3.2. Pobieranie danych ze strony GUS

Tabela zawiera wiele informacji. Można z niej wybrać te, które interesują cię najbardziej (rys. 3.3.).

Uwaga!

Jeśli nie masz licencji na używanie Excela, arkusz możesz otworzyć w którymś z darmowych programów, np. LibreOffice.



Przebieg miesięczne wynagrodzenie nominalne brutto w sektorze przedsiębiorstw:

12																			
13	w złotych	a	4277,32	4304,85	4577,86	4489,07	4390,99	4508,08	4501,52	4492,63	4473,06	4574,35	4610,79	4973,73					
14		b	4568,58	4599,72	4888,56	4840,44	4698,59	4848,18	4825,02	4798,27	4771,86	4821,39	4966,61						

Utwórz nowy arkusz w tym samym skoroszycie lub w nowym.

Wklej ze schowka zapamiętane komórki tabeli (np. skrótem **Ctrl + V**).

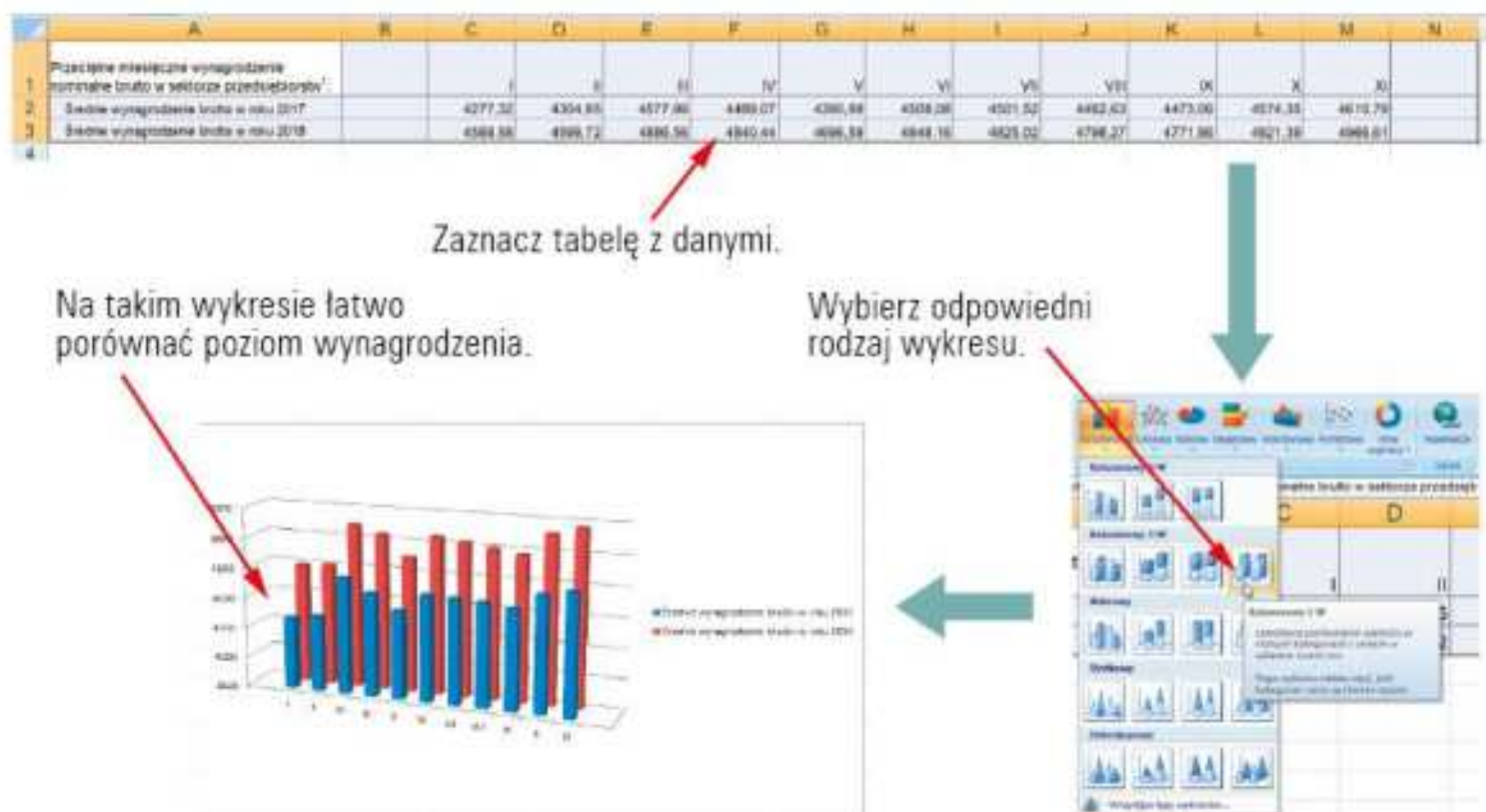
Zaznacz komórki z danymi i skopiuj do schowka (np. skrótem **Ctrl + C**).

Przebieg miesięczne wynagrodzenie nominalne brutto w sektorze przedsiębiorstw:

1																			
2	w złotych	a	4277,32	4304,85	4577,86	4489,07	4390,99	4508,08	4501,52	4492,63	4473,06	4574,35	4610,79	4973,73					
3		b	4568,58	4599,72	4888,56	4840,44	4698,59	4848,18	4825,02	4798,27	4771,86	4821,39	4966,61						

Rys. 3.3. Przenoszenie danych do nowego arkusza

Skoro dane zostały już wyodrębnione, pora na ich analizę. Ułatwi to utworzenie odpowiedniego wykresu. Dla danych z naszego przykładu najodpowiedniejsze będą wykresy, na których można porównać kwoty z obu lat (rys. 3.4.).



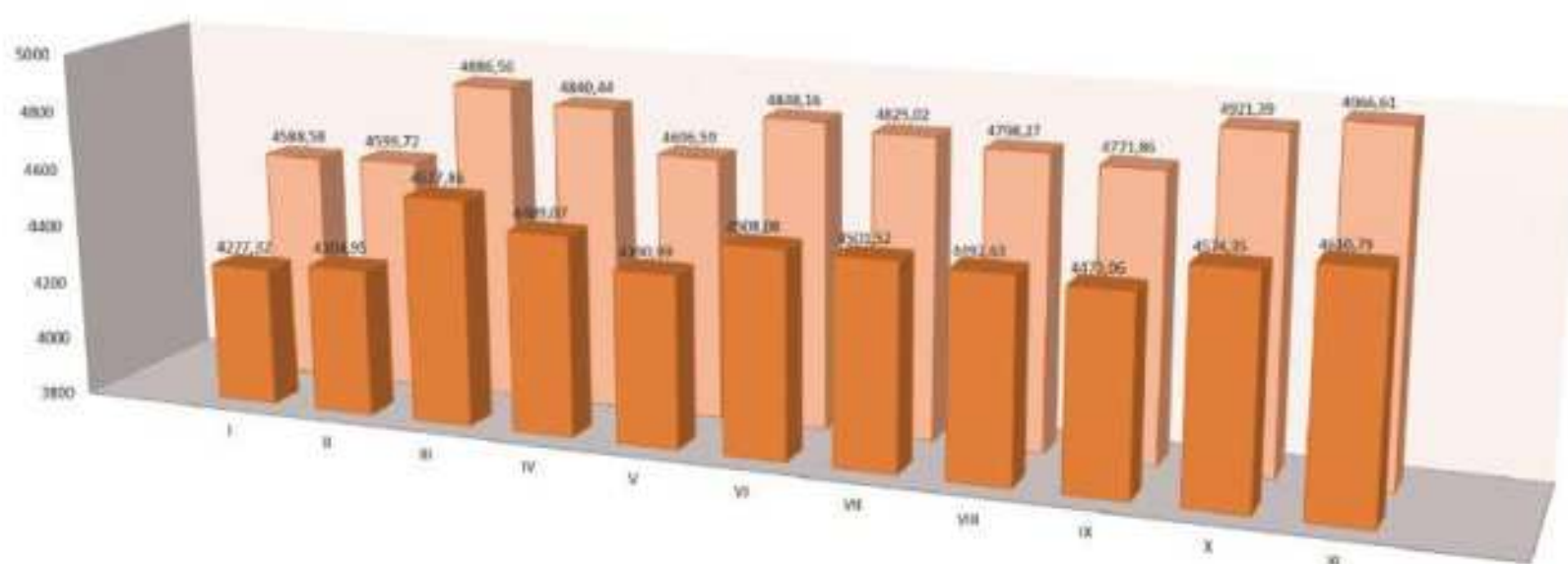
Rys. 3.4. Wizualizacja danych z arkusza kalkulacyjnego



Wykres danych z tabeli pobranej z GUS pozwala na łatwe porównanie tych samych parametrów dla różnych okresów. Trudno jednak odczytać dokładne wartości, dlatego program wyświetla je po naprowadzeniu kursora na dany słupkę i kliknięciu lewym przyciskiem (rys. 3.5.). Jeśli chcesz wyświetlać je dla wszystkich słupków, możesz zmienić **Układ** wykresu. Jeśli użyjesz opcji **Styl wykresu**, możesz także zmienić jego kolory i sposób wyświetlania słupków (rys. 3.6.). Podobnie postępuj dla każdego rodzaju wykresu.

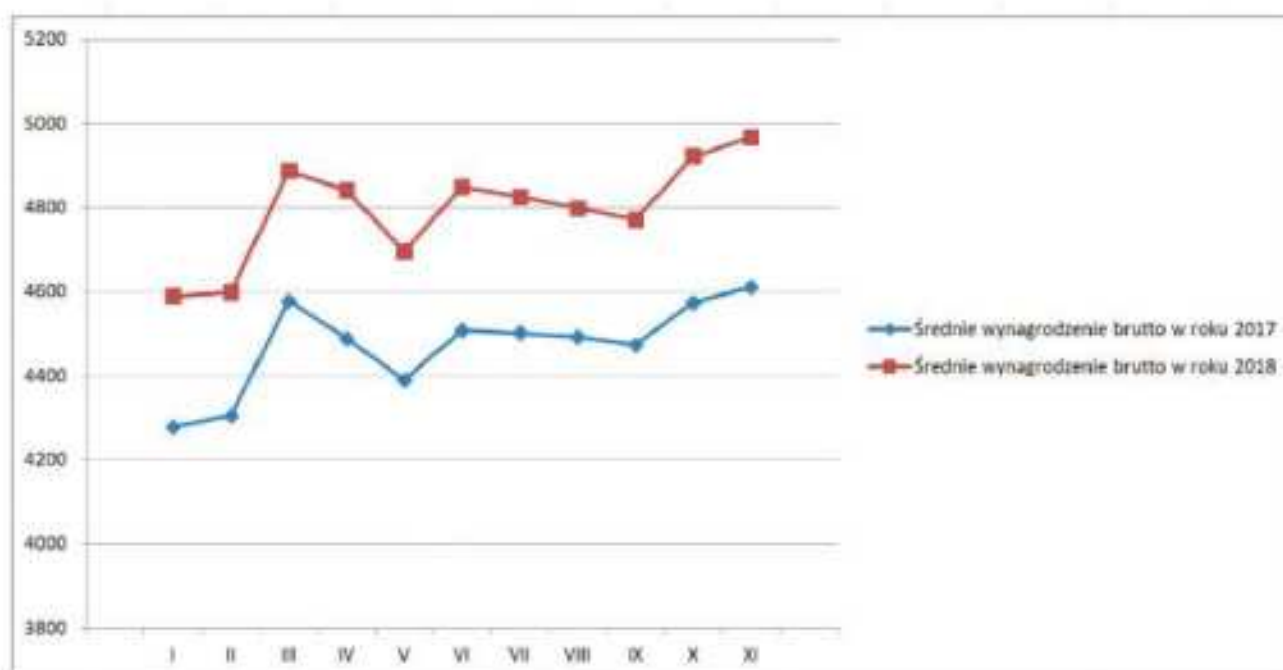


Rys. 3.5. Wyświetlanie dokładnej wartości danych dla elementu wykresu



Rys. 3.6. Wykres z danymi liczbowymi

By uwydatnić tendencje lub trendy, wystarczy zmienić typ wykresu (rys. 3.7.). Można odczytać, że w marcu i kwietniu w obu latach pensje spadały, w czerwcu wzrosły, a następnie lekko spadały aż do września.



Rys. 3.7. Wykres pokazujący trendy udziału wartości danych

Przy prowadzeniu własnej działalności można wpisać do arkusza przychody lub wydatki i śledzić je w danym okresie. Pomoże to w planowaniu.

Jeśli dane, które chcesz umieścić w arkuszu, znajdują się początkowo w dokumencie tekstowym, możesz je przenieść za pośrednictwem schowka systemu.

3.2. W przyszłym roku może być lepiej, czyli sortowanie i analiza danych rankingu

Każdy z was zapewne przynajmniej raz analizował jakiś ranking opublikowany w sieci. Niektórych interesowała popularność wykonawcy muzycznego, innych – ceny ubezpieczeń, a jeszcze innych – polecane mikroprocesory. Jeśli takie rankingi są układane według obiektywnych kryteriów, mogą być bardzo pomocne, np. w podejmowaniu decyzji o zakupie danego towaru, wyborze usługi, uczelni itp. Zdarza się, że tabele z rankingami zawierają wiele danych, w tym jednak takie, które nas nie interesują. Trzeba więc posegregować informacje według innych kryteriów. Najczęściej nie można zrobić tego na stronie internetowej. Jeśli budowa strony na to pozwoli, można przenieść tabelę do arkusza. Domyślasz się, że po takiej czynności możesz wykonać wiele operacji na danych z tabeli. Czas na przykład.

Załóżmy, że zamierzasz kupić laptop. Wybrałeś/wybrałaś już kilka modeli. Przyszedł czas na ostateczną decyzję. Zanim ją podejmiesz, chcesz przyjrzeć się bliżej ich procesorom. W tym celu zapoznajesz się z rankingami z kilku stron. Z łatwością je znajdziesz, sprawnie posługując się wyszukiwarką.



Tabele często są jednak rozbudowane i zawierają wiele danych, które z twojego punktu widzenia mogą nie być ważne. Chciałbyś/chciałabyś je uprościć i móc segregować procesory według różnych parametrów. Na szczęście wiele takich tabel da się przenieść do arkusza (rys. 3.8.). Tym razem użyjemy Excela.

import tabel

Mamy do dyspozycji narzędzia, które pozwalają importować tabele z różnych miejsc lub formatów, w tym ze strony internetowej. Wykorzystajmy je.

Ze wstążki **Dane** wybierz narzędzie **Z sieci Web**. Wpisz adres strony z danymi (np. tabelą).

Poczekaj na wczytanie się obiektów. Wybierz **Tabele**, a następnie wciśnij **Łańcuch**.

Na rysunku przedstawiono tabelę po formatowaniu – ustaleniu odpowiedniej szerokości kolumn.

Id	Nazwa procesora	L3 Cache + L2 Cache	TDP	Min. Turbo	Max. Turbo	Współczynnik / Wzrost
1	AMD Ryzen Threadripper 3970X	128MB + 64MB	350	3000	4200	24/48
2	AMD Ryzen 9 3900X	64MB + 32MB	165	3000	4700	16/32
3	AMD Ryzen 9 3900	32MB + 16MB	105	3000	4600	16/36
4	AMD Ryzen 7 3800X	32MB + 16MB	105	3000	4400	16/36
5	AMD Ryzen Threadripper 2990X	64MB + 32MB	180	3500	4400	16/32
6	AMD Ryzen 9 2900X	32MB + 16MB	105	3000	4400	16/36
7	AMD Ryzen 7 2800X	32MB + 16MB	105	3000	4400	16/36
8	AMD Ryzen 5 2600	32MB + 16MB	65	3000	4000	16/36
9	AMD Ryzen 5 2400	32MB + 16MB	65	3000	4000	16/36
10	AMD Ryzen Threadripper 2950X	32MB + 16MB	180	3500	4200	16/32
11	AMD Ryzen 7 2700	32MB + 16MB	65	3000	4000	16/36

Rys. 3.8. Przykład kopiowania danych ze strony internetowej do arkusza Excel

sortowanie w tabeli arkusza

Tak skopiowane dane można sortować według danych z poszczególnych kolumn (rys. 3.9.). Znacznie ułatwi to interpretację informacji zawartych w tabelach.



Kliknij przycisk otwierający okno dialogowe sortowania i wybierz kryterium. Ponieważ szukasz najszybszego procesora w rankingu (zdobywcy największej liczby punktów), wybierz **Sortuj od Z do A**.

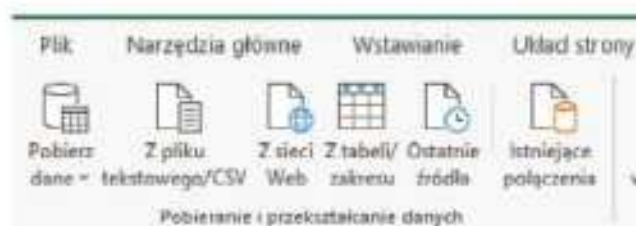
	A	B	C	D	E	F	G
	Nazwa procesora	L2 Cache + L3 Cache	TDP	MHz - Turbo	Rdzenie / Wątki	Perf. Rating	Cinebench Single 640
2	Procesor 1	12MB + 64MB	250	3000 - 4200	24/48	11	173
3	Procesor 2	8MB + 64MB	105	3500 - 4700	16/32	11	209
4	Procesor 3	18MB + 24.8MB	165	3000 - 4600	18/36	11	208n3
5	Procesor 4	18MB + 24.8MB	165	3000 - 4400	18/36	11	197.5n2
6	Procesor 5	8MB + 32MB	180	3500 - 4400	16/32	11	176
7	Procesor 6	6MB + 64MB	105	3800 - 4600	12/24	11	208n2
8	Procesor 7	6MB + 64MB	65	3100 - 4300	12/24	11	197
9	Procesor 8	2.5MB + 20MB	125	3700 - 5300	10/20	11	224
10	Procesor 9	8MB + 32MB	180	3400 - 4200	16/32	11	159
11	Procesor 10	6MB + 32MB	180	3500 - 4300	12/24	11	178
12	Procesor 11	4MB + 32MB	65	3600 - 4400	8/16	11	204
13	Procesor 12	2MB + 16MB	127	4000 - 5000	8/16	11	216n2
14	Procesor 13	2MB + 16MB	95	3600 - 5000	8/16	11	212n9
15	Procesor 14	2MB + 16MB	45	2400 - 5300	8/16	11	205.5n2
16	Procesor 15	2MB + 16MB	45	2400 - 5300	8/16	11	
17	Procesor 16	2MB + 16MB	45	2400 - 5300	8/16	11	

	A	B	C	D	E	F
	Nazwa procesora	L2 Cache + L3 Cache	TDP	MHz - Turbo	Rdzenie / Wątki	Perf. Rating
2	Procesor 2	8MB + 64MB	105	3500 - 4700	16/32	94.4
3	Procesor 3	18MB + 24.8MB	165	3000 - 4600	18/36	88.5
4	Procesor 4	18MB + 24.8MB	165	3000 - 4400	18/36	87.6
5	Procesor 1	12MB + 64MB	250	3000 - 4200	24/48	85.9
6	Procesor 6	6MB + 64MB	105	3800 - 4600	12/24	85.4
7	Procesor 8	2.5MB + 20MB	125	3700 - 5300	10/20	85.3
8	Procesor 5	8MB + 32MB	180	3500 - 4400	16/32	79.1
9	Procesor 12	2MB + 16MB	127	4000 - 5000	8/16	76.7
10	Procesor 9	8MB + 32MB	180	3400 - 4200	16/32	75
11	Procesor 10	6MB + 32MB	180	3500 - 4300	12/24	74.6

Rys. 3.9. Sortowanie w tabeli z pobranymi danymi

Oczywiście gdy dane ze wskazanego źródła zostaną już pobrane, możesz dowolnie nimi operować w ramach możliwości arkusza: wykonać wykres, użyć dowolnej formuły itp. Dzięki temu interpretacja danych z innych źródeł, np. ze strony internetowej, czy włączenie ich do własnych dokumentów (pamiętaj o prawach autorskich) będzie ułatwiona.

W niektórych przypadkach narzędzie do pobierania danych zawiedzie. Możesz wtedy spróbować użyć innego (rys. 3.10.). Sprawdź w opcji **Pomoc** (F1), w jakich sytuacjach będą one przydatne.



Rys. 3.10. Narzędzia do pobierania i przekształcania danych z różnych źródeł do tabeli arkusza Excel

operacje na importowanych danych

wstawianie tabeli z innego dokumentu



Czasem sprawdza się najprostsze rozwiązanie z użyciem **Schowka** (**Ctrl+C**, **Ctrl+V**).

Uwaga!

Jeśli interesują cię testy procesorów i ich wyniki, sprawdź w sieci znaczenie nazw niektórych kolumn przykładowej tabeli. Dowiedz się, czym jest Cinebench R15 CPU Single 64Bit lub x264 Pass 1.

ZADANIA DO WYKONANIA

1. Na stronie GUS-u umieszczono infografiki będące wykresami danych z różnych dziedzin, którymi zajmuje się urząd (<http://stat.gov.pl/infografiki-widzety>). Przeczytaj instrukcję ich obsługi, a następnie wypróbuj działanie. Zaplanuj wykonanie takich infografik działających w arkuszu kalkulacyjnym. Po kliknięciu fragmentu wykresu powinna się pojawić wartość danej. Ten mechanizm nie działa poza arkuszem.
2. Jeśli po maturze planujesz pójście na studia, znajdź w internecie strony poświęcone rankingom uczelni. Wczytaj do arkusza tabele zawierające interesujące cię dane na ich temat. Użyj optymalnego sposobu pobierania danych i uzasadnij jego wybór. Zastosuj narzędzia sortowania do analizy ofert. Wymień posortowane dane i uzasadnij swój wybór.
3. Jeśli po ukończeniu szkoły planujesz rozpocząć pracę zawodową, wyszukaj w internecie strony, na których znajdziesz dane związane z wybranym zawodem, np. dotyczące narzędzi lub urządzeń używanych przez pracowników, cenniki usług itp. Sprawdź, które z nich udostępniono jako tabele. Użyj optymalnego sposobu pobierania danych i uzasadnij jego wybór. Jeśli to ułatwi analizę, wykonaj odpowiednie wykresy.
4. Przygotuj prezentację omawiającą przeznaczenie wybranych wykresów arkusza. W tym celu wykonaj ich zrzuty ekranowe. Dane wybierz tak, by jednoznacznie ilustrowały opis zastosowania. Prezentację ulóż za pomocą dowolnego edytora.
5. Opisz w prezentacji funkcje, jakie pełnią narzędzia z rysunku 3.10. Wykorzystaj w tym celu opcję Pomoc (F1). Dla wybranego narzędzia opis zilustruj wykonanymi przez siebie zrzutami ekranowymi poszczególnych faz działania narzędzia.

PODSUMOWANIE LEKCJI

s. 26

- Pobieranie (import) danych do tabeli w arkuszu kalkulacyjnym można wykonać na kilka sposobów.

s. 27

- Przeniesienie danych do arkusza, dzięki możliwości użycia jego narzędzi, formuł itp., umożliwi **wykonanie na nich dowolnych operacji** dostępnych w arkuszu, co znacznie ułatwi ich wizualizację i analizę.

s. 26

- **Sortowanie w tabeli arkusza** pozwala na szybsze i dokładniejsze analizowanie danych.



4. Spośród wielu, czyli filtrowanie w arkuszu kalkulacyjnym

NA TEJ LEKCJI:

- poznasz sposoby filtrowania informacji zapisanych w arkuszu;
- nauczysz się tworzyć nowe arkusze na podstawie filtrowanych tabel.

W poprzednim rozdziale dowiedziałeś/dowiedziałaś się, jak wczytywać do arkusza tabele z różnych źródeł. Nauczyłeś/nauczyłaś się również, jak sortować dane w listach danych (kolumnach tabel). Wiesz już, że to pozwala ustawiać kolejność wierszy według różnych wielkości zapisanych w kolumnach. Przykładem może być sortowanie informacji w rankingu uczelni, np. ustawienie kolejności według miejsc, liczby miejsc na danych wydziałach, liczby punktów potrzebnych do przyjęcia itp. W tym rozdziale poznasz filtrowanie, czyli kolejny sposób na wyszukiwanie informacji w tabelach arkusza. Po jego przeczytaniu i wykonaniu opisanych ćwiczeń docenisz te mechanizmy.

4.1. Ustalamy kryteria, czyli kiedy warto skorzystać z filtrowania

Kiedy warto stosować filtrowanie? To dobre pytanie, ale odpowiedź nie jest prosta. Rozważmy kilka przypadków dla tabel z danymi zapisanymi w wielu wierszach i listach (kolumnach). W dużym zbiorze danych trudno znaleźć wszystkie interesujące nas informacje. Załóżmy, że arkusz zawiera dane 1000 osób, w tym imiona, nazwiska, adresy i wiek.

1. **Szukamy osób zamieszkujących te same miejscowości.** Bez filtrowania trzeba by przeszukać całą kolumnę, w której zapisano miejscowości, a następnie skopiować znalezione dane, np. do innego arkusza. W tym przypadku z pomocą może przyjść znany już mechanizm sortowania, jednak samo przenoszenie danych do innego arkusza lub tabeli będzie czasochłonne i żmudne.
2. **Znalezienie i skopiowanie danych mieszkańców tej samej miejscowości** przy odrobinie wprawy nie będzie bardzo trudne. A co zrobić, jeśli wśród tych mieszkańców chcemy wyodrębnić dodatkowo wszystkich o imieniu Zofia lub Adam, którzy mają 25 lat? Tu filtry okażą się niezbędne.
3. A jeśli wśród danych znajdują się **dłuższe teksty**, np. nazwy ukończonych szkół lub uczelni, a nam będzie zależało na znalezieniu absolwentów techników? Kolumna **Ukończona szkoła** może przecież zawierać pełne nazwy szkół. Filtrowanie w arkuszu pozwoli na wyszukanie konkretnych fraz i wyświetlenie w tabeli jedynie wierszy z szukaną w danej kolumnie wartością, np. *Technikum Elektroniczne*.

przykłady
zastosowań
filtrowania danych



- Podobnie rzecz się ma z danymi, które chcemy **wykluczyć**. Można użyć filtru, który pozostawi jedynie dane osób niemających w wybranej kolumnie wpisanej pewnej frazy lub liczby. Użycie w takim filtrze wspomnianej wyżej sekwencji znaków *Technikum Elektroniczne* spowoduje, że w arkuszu zostaną wyświetlone dane osób, które technikum elektronicznego nie kończyły.
- A czy można **filtrować według koloru**? Domyślasz się zapewne, że tak. Zdarza się, że kolorów używa się do wyróżniania różnych wartości, np. najważniejszych, ekstremalnych albo mieszczących się w pewnym zakresie.
- Dane, które pozostaną po filtrowaniu, dzięki użyciu **opcji zaawansowanych** można **automatycznie przenieść** do wskazanych komórek arkusza.

Uwaga!

Po operacji filtrowania dane, których nie widać w tabeli, nie są usuwane. Nadal pozostają w arkuszu, przestają być jedynie wyświetlane.

filtrowanie nie usuwa danych z arkusza

W Excelu filtrowanie dla danej listy można włączyć lub wyłączyć za pomocą opcji w sekcji **Sortowanie i filtrowanie** z wstążki **Dane**.

4.2. Mniejsze, większe, czyli filtrujemy zakresy danych

W przykładzie użyjemy nieznacznie zmienionej tabeli z poprzedniego rozdziału (rys. 4.1.). Nie jest bardzo rozbudowana, ale zawiera wystarczającą liczbę danych, by można było prześledzić proces filtrowania i jego efekty.

W filtrowaniu pomoże menu kontekstowe dopasowujące się do rodzaju filtrowanych danych.

menu kontekstowe

	A	B	C	D	E	F	G	H	I	J	K	L	M
		Pełna nazwa	L2 Cache + L3 Cache	TDP	Hz - Turbo	Ważność / Wzrost	Perf. Rating	Cinebench R15 CPU Single 64bit	Cinebench R15 CPU Multi 64bit	Cinebench R20	wPrime 32M J	AIDA64 Pass 1	AIDA64 Pass 2
2		Processor AMD 1	12MB + 39MB	250	3000 - 4200	24/18	85.3	173	4276	13988	4.18	302	138.5
3		Processor INTEL 1	9MB + 8448	105	3500 - 4700	14/12	70.4	209	4002	8075	4.11	365	198
4		Processor INTEL 2	10MB + 24 0MB	165	3000 - 4800	18/16	87.7	288	5748	8533.5	3.81	215	157.5
5		Processor AMD 2	10MB + 24 0MB	105	3000 - 4400	18/16	86.5	190.3	3904	8837	3.41	134.25	187
6		Processor AMD 3	9MB + 32MB	180	3000 - 4400	14/12	78.3	170	3110	6936	4.19	302.31	144.88
7		Processor INTEL 3	9MB + 8448	105	3000 - 4800	14/12	86.6	288	3176.5	8887.5	3.68	207.3	187
8		Processor AMD 4	9MB + 8448	85	3100 - 4300	14/12	75.5	187	2884	6331		255.5	133.9
9		Processor INTEL 4	2.5MB + 30MB	125	3700 - 5000	18/18	94.4	224	3620	6281	3.38	280.3	149
10		Processor INTEL 5	9MB + 32MB	180	3600 - 4200	14/12	76.2	159	2994	6790	3.24	299.58	134.28
11		Processor AMD 5	9MB + 32MB	180	3000 - 4200	14/12	73.4	158	3038	5868	3.03	286.5	131

Rys. 4.1. Przykładowa tabela arkusza

Gdy analizujemy rankingi, najczęściej interesują nas produkty o określonych parametrach. W pierwszym przykładzie odfiltrujemy wszystkie procesory, które wydzielają najwięcej ciepła, a więc potrzebują najwięcej mocy pobieranej z zasilacza do wykonania określonych zadań. Zależy nam bowiem na jak najmniejszym zużyciu energii. Ta właściwość jest określana przez parametr TDP mówiący o ilości ciepła, które trzeba odebrać z procesora w ciągu jednej sekundy.

Uwaga!

Przykładowa tabela zawiera jedynie 10 przykładów procesorów. Użyto jej dla uproszczenia ćwiczeń. W rzeczywistych zestawieniach takich wierszy może być znacznie więcej i zastosowanie filtrów stanie się niezbędne.



W ćwiczeniu pozostawimy na ekranie jedynie procesory, których współczynnik TDP znajduje się pomiędzy 100 a 165 (rys. 4.2.).

Pełna nazwa	L3 Cache + L3 Cache	TDP	MHz - Turbo	Rdzenie / Wątki
Processor AMD 1	128	Łącznie najwyżej sześć do najniższych	3000 - 4200	24/48
Processor INTEL 1	8M	Łącznie najwyżej sześć do najniższych	3500 - 4700	16/32
Processor INTEL 3	18M	Łącznie sześć do najniższych	3000 - 4600	18/36
Processor AMD 2	18M	Łącznie sześć do najniższych	3000 - 4400	18/36
Processor AMD 3	8M	Łącznie sześć do najniższych	3500 - 4400	16/32
Processor INTEL 3	6M	Łącznie sześć do najniższych		
Processor AMD 4	6M	Łącznie sześć do najniższych		
Processor INTEL 4	2.5	Łącznie sześć do najniższych		
Processor INTEL 5	8M	Łącznie sześć do najniższych		
Processor AMD 5	6M	Łącznie sześć do najniższych		

Dla listy TDP rozwiń menu sortowania i filtrowania.
Wybierz opcję **Filtry liczb / Między**.

Auto fit spreadsheet

Initial values, in steps

per step increase in full volume 100

per step increase in rate volume 100

Symbol 1 reduplicate density peak.

Symbol 2 reduplicate density gap profile.

OK Cancel

W odpowiednie pola wpisz zakres filtrowanych liczb.

Pełna nazwa	L2 Cache + L3 Cache	TDP	MHz - Turbo
Procesor INTEL 1	8MB + 64MB	105	3500 - 4700
Procesor INTEL 2	18MB + 24.8MB	165	3000 - 4600
Procesor AMD 2	18MB + 24.8MB	165	3000 - 4400
Procesor INTEL 3	6MB + 64MB	105	3800 - 4600
Procesor INTEL 4	2.5MB + 20MB	125	3700 - 5300

Miedzy z parametrem „i”

Pełna nazwa	L2 Cache + L3 Cache	TDP	MHz - Turbo
Procesor INTEL 2	18MB + 24.8MB	165	3000 - 4600
Procesor AMD 2	18MB + 24.8MB	165	3000 - 4400
Procesor INTEL 4	2.5MB + 20MB	125	3700 - 5300
Procesor INTEL 1	8MB + 64MB	105	3500 - 4700
Procesor INTEL 3	6MB + 64MB	105	3800 - 4600

Po użyciu opcji sortowania łatwiej będzie wybrać procesor.

Rys. 4.2. Przykład użycia filtra liczb z warunkiem **Miedzy**

Użyty filtr pozwalał ustalić, czy warunek **Między** ma mieć postać większy od i mniejszy od czy większy od lub mniejszy od.

Zbadajmy, na czym polega różnica (rys. 4.3.). Podczas analizy przyda się przypomnienie pojęć **koniunkcja (i)** i **alternatywa (lub)**.

Definicje

Koniunkcja (iloczyn logiczny) to zdania połączone spójnikiem logicznym i. Koniunkcja jest prawdziwa, gdy wszystkie jej zdania są prawdziwe.

Alternatywa (suma logiczna) to zdania połączone spójnikiem **lub**. Alternatywa jest prawdą, gdy przynajmniej jedno z jej zdań jest prawdziwe.

koniunkcja i alternatywa



Jeżeli użyjemy koniunkcji (i) w filtrze **Między**, oba warunki muszą być spełnione. Liczby muszą więc być **jednocześnie** większe lub równe 100 i mniejsze lub równe 165 (rys. 4.3.).

Pokaz wiersze, w których:
TDP

Jest większe niż lub równe

☒ I ☐ LUB

Jest mniejsze niż lub równe

→

Pełna nazwa	L2 Cache + L3 Cache	TDP
Procesor INTEL 2	18MB + 24.8MB	165
Procesor AMD 2	18MB + 24.8MB	165
Procesor INTEL 4	2.5MB + 20MB	125
Procesor INTEL 1	8MB + 64MB	105
Procesor INTEL 3	6MB + 64MB	105

Rys. 4.3. Użycie koniunkcji w filtrze **Między**

Jeśli zastosujesz alternatywę (lub), filtr pozostawi wszystkie dane, które spełniają przynajmniej jedno z ograniczeń, czyli wszystkie większe od 100 lub mniejsze od 165. W przypadku takich liczb w naszym przykładzie zostaną wybrane wszystkie pola (rys. 4.4a.). Użycie alternatywy ma sens, jeżeli odpowiednio dobierzemy parametry, np. TDP jest większe od 165 lub mniejsze od 100, czyli pozostawimy na ekranie wszystkie procesory poza tymi z rysunku 4.3. W tym przypadku tabela wyświetli dane jak na rysunku 4.4b.

a)

Pokaz wiersze, w których:
TDP

Jest większe niż lub równe

☐ I ☒ LUB

Jest mniejsze niż lub równe

→

Pełna nazwa	L2 Cache + L3 Cache	TDP
Procesor AMD 1	12MB + 64MB	250
Procesor INTEL 2	18MB + 24.8MB	165
Procesor AMD 2	18MB + 24.8MB	165
Procesor INTEL 4	2.5MB + 20MB	125
Procesor AMD 3	8MB + 32MB	180
Procesor INTEL 1	8MB + 64MB	105
Procesor AMD 4	6MB + 64MB	65
Procesor INTEL 3	6MB + 64MB	105
Procesor INTEL 5	8MB + 32MB	180
Procesor AMD 5	6MB + 32MB	180

b)

Pełna nazwa	L2 Cache + L3 Cache	TDP
Procesor AMD 1	12MB + 64MB	250
Procesor AMD 2	18MB + 24.8MB	165
Procesor AMD 3	8MB + 32MB	180
Procesor AMD 4	6MB + 64MB	65
Procesor AMD 5	6MB + 32MB	180

Rys. 4.4. Użycie alternatywy w filtrze **Między**

Po analizie opisanego filtru z łatwością użyjesz pozostałych związanych z filtrowaniem liczb. Pamiętaj jednak o umiejętnym wyborze pomiędzy koniunkcją a alternatywą.

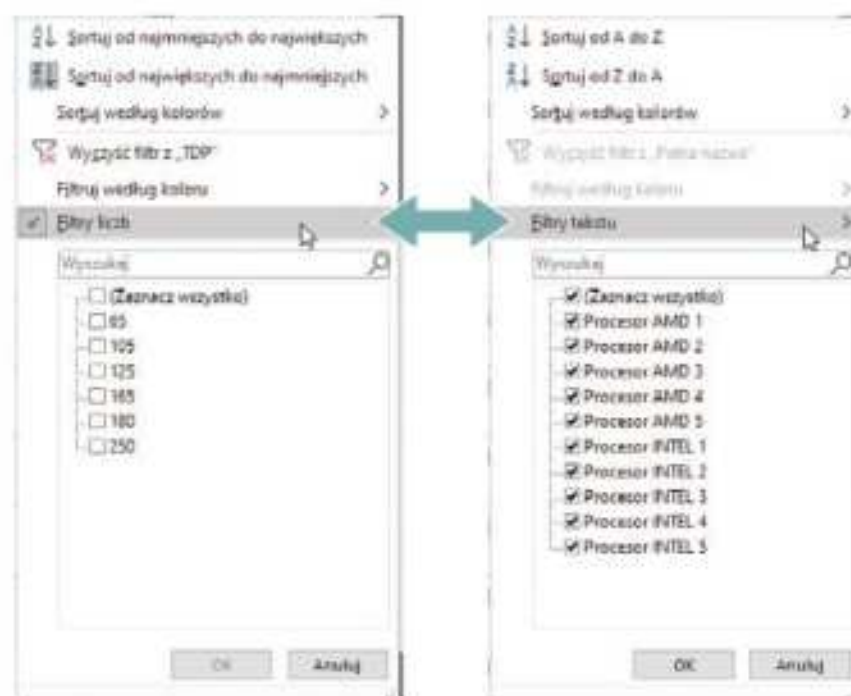


Uwaga!

Koniunkcja i alternatywa w wyborze parametrów filtrów występuje w filtrach dotyczących zarówno liczb, jak i tekstów.

4.3. Nie tylko liczby, czyli odnajdujemy filtry tekstu i koloru

W przykładzie pierwsza lista zawiera nazwy procesorów wraz z ich marką. Jeśli chcemy wyświetlić dane układów tylko jednego producenta, możemy użyć filtrów tekstu. Zauważ, że arkusz rozpoznał zawartość listy i w menu kontekstowym w miejscu filtrów liczb pojawiły się filtry tekstu (rys. 4.5.).



Rys. 4.5. Menu kontekstowe filtrów w zależności od zawartości listy

Jeśli interesuje cię wynik i parametry, np. procesorów danego producenta, użyj takiego filtru, który wykryje w nazwie odpowiedni tekst. Należy założyć, że może ona zawierać także inne wyrazy (w naszym przypadku słowo *Processor* i cyfrę). By żaden procesor danej marki nie zniknął nam z pola widzenia po zastosowaniu filtru, użyjemy opcji **Zawiera** (rys. 4.6.).

Pełna nazwa	L2 Cache + L3	TDP
AMD Ryzen Threadripper 2970WX	12MB + 64MB	250
AMD Ryzen 9 3950X	8MB + 64MB	105
AMD Ryzen Threadripper 2990X	8MB + 32MB	180
AMD Ryzen 9 3900X	8MB + 64MB	105
AMD Ryzen 9 3900	8MB + 64MB	65
AMD Ryzen Threadripper 1950X	8MB + 32MB	180
AMD Ryzen Threadripper 2920X	8MB + 32MB	180
AMD Ryzen 9 3700X	4MB + 32MB	65

Rys. 4.6. Przykład użycia **Filtru** tekstu z opcją **Zawiera**



filtrowanie kolorem

Pozostałe opcje tego filtru nie będą ci sprawiały trudności. W razie problemów możesz skorzystać z funkcji **Pomoc** (F1).

Zapewne zauważyłeś/zauważyłaś, że w naszej tabeli niektóre pola wypełniono danymi, używając czerwonych znaków. W praktyce może to oznaczać, że dana wartość np. wykracza poza normę. Kolorem można także wyróżniać informacje z innych przyczyn. Należy jednak w opisie tabeli podać, co oznaczają poszczególne barwy. Skoro można wyróżniać w ten sposób niektóre dane, powinna istnieć możliwość ich filtrowania. W naszej tabeli zaznaczono kolorem procesory z największą pojemnością pamięci cache i najwyższą częstotliwością taktowania. Użyjmy odpowiedniego filtru, by odnaleźć te najszybciej taktowane (rys. 4.7.).

TDP	MHz - Turbo	Rdzenie / Wątki
250		24/48
165		18/36
165		
125		
180		
105		16/32
65		12/24
105		12/24
180		16/32
180		12/24

W menu wystąpią kolory użyte w danej liście.

Pełna nazwa	L2 Cache + L3 Cache	TDP	MHz - Turbo
Procesor INTEL 4	2.5MB + 20MB	125	3700 - 5300
Procesor INTEL 3	6MB + 64MB	105	3800 - 4600

Rys. 4.7. Przykład użycia filtrowania według koloru



4.4. Fragmentator, czyli filtrowanie z różnymi kryteriami

Użycie **Fragmentatora** jest ciekawym sposobem filtrowania, bardzo przydatnym podczas analizy danych z tabel, w których musimy wybierać dane według wielu kryteriów.

Opcja pomocy w arkuszu Excel tak definiuje ten sposób filtrowania:

fragmentator

„**Fragmentatory** udostępniają przyciski, które można kliknąć w celu filtrowania tabel lub tabel przestawnych. Oprócz szybkiego filtrowania fragmentatory wskazują też bieżący stan filtrowania, co ułatwia zrozumienie, co jest obecnie wyświetlane”.

Użyjmy fragmentatora w naszej tabeli. Odfiltrujmy wszystkie procesory AMD i INTEL poniżej numeru 3 i mające 10 lub 12 rdzeni (rys. 4.8.). Spójnik i odgrywa w naszym zadaniu dużą rolę. Oznacza, że wyświetlane wiersze muszą spełniać wymagania obu filtrów **jednocześnie**.

Kliknij dowolne pole tabeli, a następnie na wstążce **Wstawianie** wybierz **Fragmentator**.

Zaznacz nagłówki list, według których nastąpi filtrowanie Fragmentatorem.

Aby użyć więcej niż jednego przycisku filtrowania, wciśnij **Wybór wielokrotny**.

Po wybraniu naszego filtru dla listy **Rdzenie/Wątki** odfiltrowany zostanie także wiersz z Pełną nazwą **Procesor INTEL 3**. Nie spełnia on bowiem wymagań jednego z filtrów.

Pełna nazwa	L2 Cache + L3 Cache	TDP	MHz - Turbo	Rdzenie / Wątki
Procesor AMD 1	12MB + 64MB	250	3000 - 4200	24/48
Procesor INTEL 2	18MB + 24.8MB	165	3000 - 4600	18/36
Procesor AMD 2	18MB + 24.8MB	165	3000 - 4400	18/36
Procesor AMD 3	8MB + 32MB	180	3500 - 4400	16/32
Procesor INTEL 1	8MB + 64MB	105	3500 - 4700	16/32

Pełna nazwa	Rdzenie / Wątki
Procesor AMD 1	12/24
Procesor AMD 2	18/36
Procesor AMD 3	18/36
Procesor INTEL 1	16/32
Procesor INTEL 2	24/48
Procesor INTEL 5	10/20
Procesor AMD 4	
Procesor AMD 5	

Rys. 4.8. Przykład zastosowania **Fragmentatora**



Używając Fragmentatora, można filtrować jednocześnie według różnych parametrów, różnych list. Należy pamiętać, że filtry będą działać zgodnie z zasadami **koniunkcji**.

ZADANIA DO WYKONANIA

1. Wykorzystaj treść poprzednich rozdziałów i pobierz z sieci tabelę zawierającą ranking wybranych elementów komputera, np. kart grafiki. Użyj filtrów, by wśród nich znaleźć najbardziej wydajne, najtańsze, najdroższe, kosztujące nie mniej niż określona kwota. W trakcie wykonywania zadania wykonaj zrzuty ekranowe, np. narzędziem **Wycinanie**, i umieść je z krótkim opisem w dokumencie tekstowym. Zapisz go w formacie PDF.
2. Wykonaj prezentację w dowolnej formie (np. PowerPoint lub film) opisującą działanie poszczególnych opcji filtrowania liczb.
3. Wykonaj prezentację w dowolnej formie (np. PowerPoint lub film) opisującą działanie poszczególnych opcji filtrowania tekstu.
4. Dla tabeli z zadania 1 ulóż własne filtry. W trakcie wykonywania zadania wykonaj zrzuty ekranowe, np. narzędziem **Wycinanie**. Opisz w dokumencie zastosowanie twoich filtrów i sens ich użycia. Dokument zapisz w formacie PDF.
5. Wykorzystaj tabelę z zadania 1 lub dowolną z twoich dokumentów i użyj **Fragmentatora**. Opisz w dokumencie tekstowym cel jego zastosowania. Wykonaj zrzuty ekranowe w celu zilustrowania wykonanego ćwiczenia.

PODSUMOWANIE LEKCJI

s. 29

- Narzędzia do filtrowania są uniwersalne i mogą służyć do wyszukiwania danych zapisanych w obszernych tabelach według różnych parametrów. **Przykładem mogą być** liczby lub ich zakresy, tekst, kolor lub dowolnie wpisany filtr (liczba lub tekst).

s. 30

- **Po filtrowaniu wiersze pozostają niezmiennione**, a odfiltrowane dane pozostają jedynie niewidoczne – nie znikają z tabel. Po usunięciu filtru zostanie wyświetlona pełna, zapisana w arkuszu tabela.

s. 30

- Menu kontekstowe udostępnia opcje **filtrowania zgodnie z zawartością listy danych** (kolumny tabeli).

s. 34

- Filtrowanie nie musi dotyczyć liczb lub tekstu. Można filtrować **według kolorów**.

s. 35

- **Fragmentatory** udostępniają przyciski do szybkiego filtrowania w tabelach. Wyświetlają także aktualny stan filtrowania.

s. 31

- W filtrowaniu z użyciem kilku filtrów należy zwrócić baczność na wybór pomiędzy **koniunkcją a alternatywą**.



5. Z eksperymentu do arkusza, czyli analiza danych z doświadczenia

NA TEJ LEKCJI:

- wykonasz doświadczenie z symulacją rzutu kostką;
- wpiszesz wyniki pomiarów doświadczeń do arkusza i z jego pomocą dokonasz ich analizy.

Wiele doświadczeń z różnych dziedzin wymaga zbierania sporej ilości informacji, które później muszą zostać przetworzone. Gdyby nie arkusz, byłaby to bardzo żmudna czynność, obarczona dużym prawdopodobieństwem popełnienia błędu. Arkusz może także posłużyć symulacji zjawisk i doświadczeń nie tylko ekonomicznych. W tym rozdziale dowiesz się, jak rzucić kostką 1000 razy w ciągu ułamka sekundy i jak można te wyniki wykorzystać.

5.1. Oszczędzamy czas, czyli symulujemy rzuty kostką do gry

W czasie przygotowań do matury możesz spotkać się z zadaniem polegającym na rzucaniu kostką do gry i obliczeniu najbardziej prawdopodobnej liczby sukcesów (wyrzucenia określonej liczby oczek) w n próbach. Oczywiście nie istnieje metoda na dokładne podanie wyniku, który sprawdziłby się w praktyce. Jednak przy odpowiednio dużej liczbie prób będzie on zbliżony do wyniku doświadczenia. Aby odkryć, czy tak jest, należy wykonać wiele rzutów, wynik każdego zapisać, a następnie policzyć, ile razy wystąpiła wcześniej określona liczba oczek i sprawdzić, czy będzie ona w przybliżeniu równa obliczeniom. Aby eksperyment miał sens, musimy wykonać co najmniej kilkaset prób.

Czy może to za nas zrobić arkusz? Jeśli wykorzystamy generator liczb pseudolosowych, to oczywiście tak. Przy okazji wyniki można zwizualizować wykresami.

symulacja zjawisk losowych

Zacznijmy od przygotowania arkusza (rys. 5.1.).

W wybrane pole wpisz formułę losującą liczbę całkowitą z zakresu od 1 do 6.

Tę samą formułę skopiuj do 999 kolejnych pól kolumny, by uzyskać tysiąc symulacji rzutu kostką.

Rys. 5.1. Przygotowanie arkusza symulującego 1000 rzutów kostką



Jeśli arkusz symuluje już odpowiednią liczbę rzutów, można przystąpić do opracowania pozostałych elementów doświadczenia. Podstawą będzie wzór na najbardziej prawdopodobną liczbę sukcesów w n próbach. W naszym przypadku sukcesem będzie wyrzucenie określonej liczby oczek, np. 6, a n to liczba wykonanych prób, czyli 1000. Wzór jest oparty na schemacie Bernoulliego (wym. bernulego).

Najbardziej prawdopodobna liczba sukcesów w n próbach jest określona jako $(n + 1)p$; p oznacza tu prawdopodobieństwo wystąpienia sukcesu w pojedynczej próbie. Ponieważ rzut kostką może dać jedynie 6 wyników, z których każdy jest tak samo prawdopodobny, to $p = 1/6$. Wynik powinien być całkowity. Jeśli nie będzie, należy przyjąć największą liczbę całkowitą mniejszą od obliczonej wartości. W naszym przypadku:

$$(n + 1)p = 1001 \div 6 \approx 166,83$$

Przyjmujemy więc, że przy 1000 próbach rzutu kostką najbardziej prawdopodobną liczbą zdarzeń polegających na wyrzuceniu danej liczby oczek jest **166**.

Uzupelnijmy arkusz o formułę wyświetlającą liczbę wystąpień danej liczby wśród wszystkich prób (rys. 5.2.). Do policzenia wystąpień danej liczby oczek w przykładzie użyjmy formuły:

`=LICZ.JEŻELI(B2:B1002;E3)`

Dla kolejnej liczby oczek formuła będzie analogiczna, liczenie odbywa się wśród tych samych danych w polach od B2 do B1002, np. dla dwójek:

`=LICZ.JEŻELI(B2:B1002;E4)`

W formule można zastosować także adresowanie mieszane. W takim przypadku, jeśli chcemy wymusić blokowanie kolumny E w formule, a pozwolić na zmianę wierszy, można użyć formuły:

`=LICZ.JEŻELI(B2:B1002;$E4)`

W tym przypadku nie jest to konieczne, warto jednak przypomnieć sobie, że istnieją inne formy adresowania komórek niż zastosowane pierwotnie adresowanie względne, np. mieszane.

=LICZ.JEŻELI(B2:B1002;E3)						
	A	B	C	D	E	F
1						
2		Wynik rzutu kostką				Liczba trafień
3		1	Wylosowano	1		152
4		4	Wylosowano	2		170
5		2	Wylosowano	3		186
6		1	Wylosowano	4		165
7		2	Wylosowano	5		169
8		3	Wylosowano	6		157
9		6				

Rys. 5.2. Obliczanie liczby wystąpień poszczególnych wyników doświadczenia z rzuca-
niem kostką do gry



Jak widać, liczba trafień oscyluje wokół obliczonej. Zilustrujmy to doświadczenie na wykresie (rys. 5.3.).



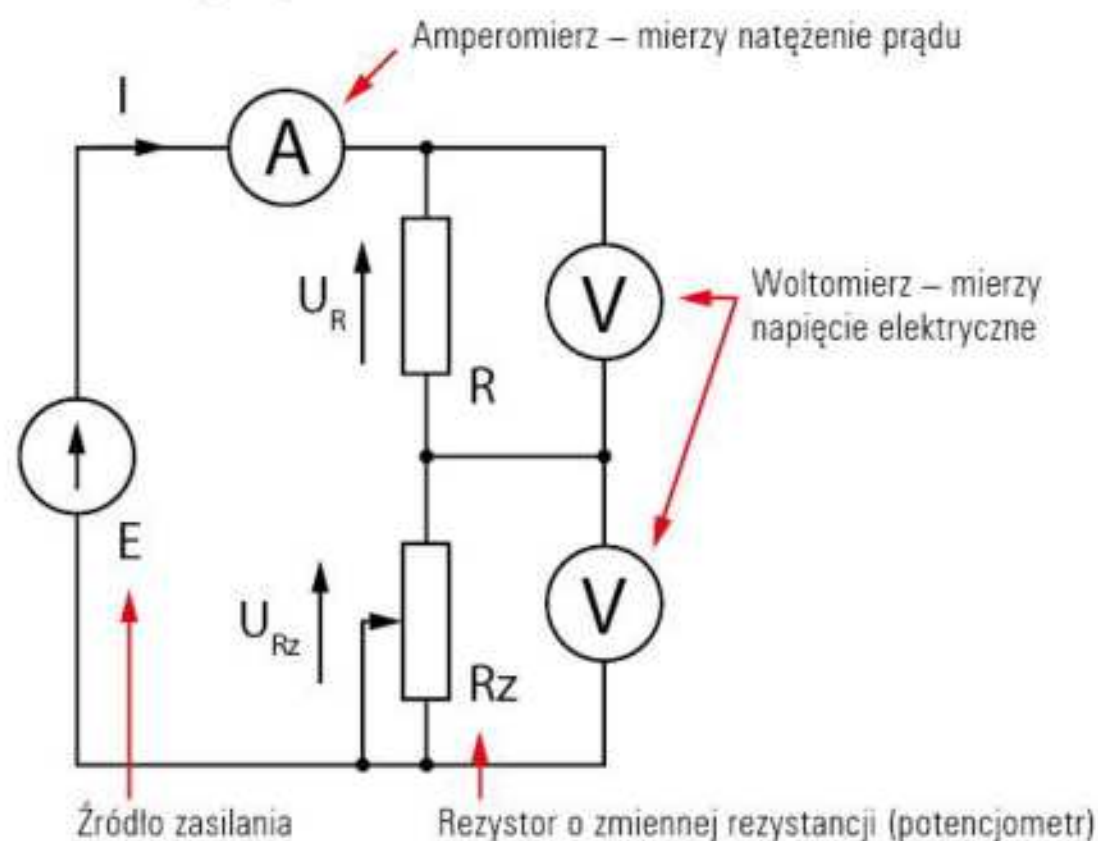
Rys. 5.3. Graficzne przedstawienie wyniku eksperymentu

Oczywiście eksperyment można powtarzać wielokrotnie, używając klawisza F9.

5.2. Pomiary elektryczne, czyli sprawdzamy prawo Ohma (wym. oma)

Przeanalizuj proste doświadczenia z elektrotechniki i zaprojektuj odpowiedni arkusz do gromadzenia danych pomiarowych.

Na rysunku przedstawiono schemat elektryczny układu pomiarowego (rys. 5.4.). Źródło napięcia E (siła elektromotoryczna) zasila obwód składający się z dwóch szeregowo połączonych rezystorów. Jeden z nich to potencjometr, w którym można regulować opór elektryczny. Na każdym z rezystorów dokonuje się pomiaru spadku napięcia za pomocą woltomierzy. Natężenie prądu w obwodzie mierzy amperomierz.



Rys. 5.4. Schemat elektryczny układu pomiarowego doświadczenia



Zapewne znasz wzór zwany prawem Ohma. Mówi on, że spadek napięcia na rezystorze równa się iloczynowi prądu przez niego płynącego i rezystancji.

$$I = \frac{U}{R} \rightarrow U = I \times R$$

W naszym doświadczeniu użyto źródła o napięciu 9 V, rezystora R o rezystancji 100Ω i potencjometru o regulowanej wartości od 0 do 400Ω .

Wyniki pomiarów natężenia prądu w obwodzie i spadki napięć na rezystorach można zapisać w tabeli arkusza (rys. 5.5.).

$$E6=B6/(C6+D6)$$

zgodnie z prawem Ohma

$$F6=E6*C6 \text{ zgodnie z prawem Ohma}$$

$$G6=E6*D6 \text{ zgodnie z prawem Ohma}$$

$$E6=B6/(C6+D6) \text{ zgodnie z prawem Ohma}$$

	A	B	C	D	E	F	G
3							
4							
5		E [V]	R [Ohm]	R₂ [Ohm]	I [A]	U_R [V]	U_r [V]
6		9	100	0	0,090	9,000	0,000
7		9	100	20	0,075	7,500	1,500
8		9	100	40	0,064	6,429	2,571
9		9	100	60	0,056	5,625	3,375
10		9	100	80	0,050	5,000	4,000
11		9	100	100	0,045	4,500	4,500
12		9	100	120	0,041	4,091	4,909
13		9	100	140	0,038	3,750	5,250
14		9	100	160	0,035	3,462	5,538
15		9	100	180	0,032	3,214	5,786
16		9	100	200	0,030	3,000	6,000
17		9	100	220	0,028	2,813	6,188
18		9	100	240	0,026	2,647	6,353
19		9	100	260	0,025	2,500	6,500
20		9	100	280	0,024	2,368	6,632
21		9	100	300	0,023	2,250	6,750
22		9	100	320	0,021	2,143	6,857
23		9	100	340	0,020	2,045	6,955
24		9	100	360	0,020	1,957	7,043
25		9	100	380	0,019	1,875	7,125
26		9	100	400	0,018	1,800	7,200
27							

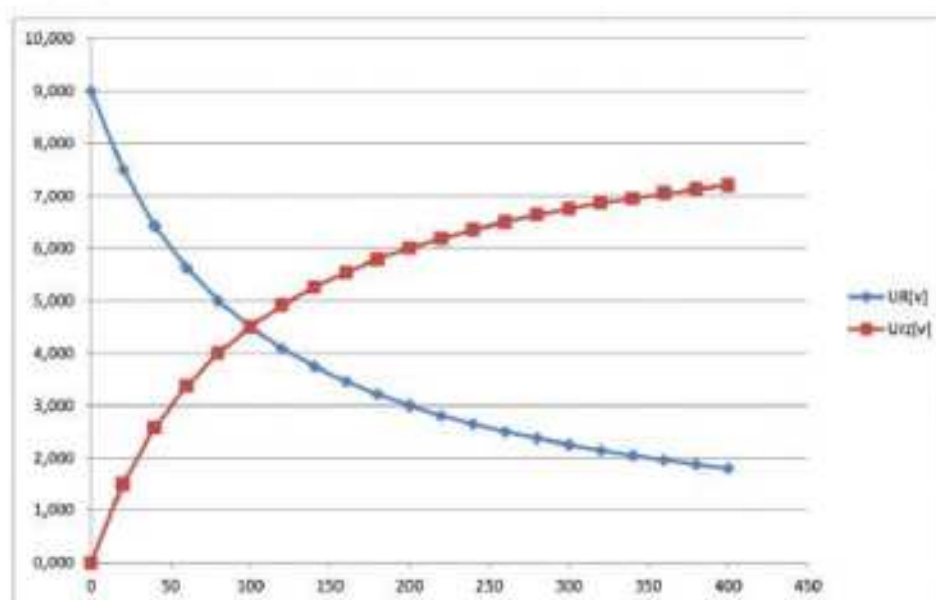
Rys. 5.5. Arkusz do symulacji pomiarów w obwodzie elektrycznym



wyniki
doświadczeń
w arkuszu

Zauważ, że w przykładowym arkuszu ograniczono liczbę cyfr po przecinku w wynikach obliczeń do trzech. Dla dwóch miejsc po przecinku wiele wyników byłoby identycznych. W celu osiągnięcia takiego efektu zaznacz komórki z danymi od E6 do G26, kliknij prawym przyciskiem i z menu wybierz **Formatuj komórki**.

Wykonajmy teraz wykresy, które pozwolą wizualnie określić, czy w obwodzie pomiarowym zostało zachowane prawo Ohma. Do sporządzenia wykresów wykorzystaj typ liniowy.

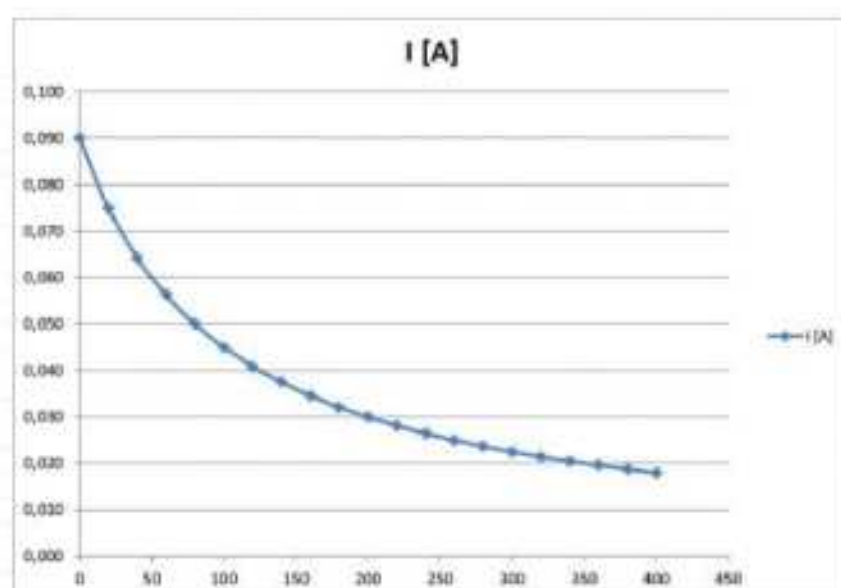


Rys. 5.6. Wykres spadków napięć na rezystorach w zależności od zmian rezystancji R_2

Zgodnie z prawami dotyczącymi obwodów elektrycznych suma spadków napięć zmierzonych na szeregowo połączonych opornikach (w naszym przypadku wskazania woltomierzy) jest równa napięciu źródła zasilającego układ (drugie prawo Kirchhoffa). Widać to w tabeli oraz na wykresie (rys. 5.6.). Zgodnie z tym prawem sumy danych w wierszach dla kolumn F i G muszą być równe 9 V. Jak widać, nie zawsze tak jest. Wynika to z niedoskonałości elementów rezystancyjnych i błędów pomiaru. W doświadczeniu użyto elementów i mierników laboratoryjnych. W przypadku wykonania doświadczenia przy użyciu amatorskich mierników i elementów gorszej jakości różnice mogą być większe.

Na wykresie (rys. 5.6.) widać zmiany spadków napięć na obu rezystorach. Dla wartości rezystancji równej $100\ \Omega$ spadki napięć są równe i wynoszą 4,5 V. Dzięki wykresowi łatwo można zauważyć, że wartości zmierzone potwierdzają prawo Ohma i drugie prawo Kirchhoffa.

Korzystając z danych z tabeli, możesz także sprawdzić, jak na natężenie prądu w obwodzie wpływa zmiana rezystancji. W tym celu można utworzyć kolejny wykres (rys. 5.7.).



Rys. 5.7. Zależność natężenia prądu w obwodzie od jego rezystancji na podstawie wyników pomiarów

wykresy
w dokumentacji

Tak sporządzoną tabelę pomiarów i wykresy możesz umieścić w dokumentacji doświadczenia i oprzeć na nich swoje wnioski. Powyższy przykład dotyczy prostego doświadczenia, jednak ilustruje możliwości arkusza. W rzeczywistości jego właściwości można wykorzystać do opracowania znacznie większych ilości danych pochodzących z obserwacji lub doświadczeń. Ułatwią to wykresy, a także poznane przez ciebie wcześniej metody sortowania i filtrowania oraz bogaty zestaw formuł.

ZADANIA DO WYKONANIA

1. Przeanalizuj jeszcze raz doświadczenie z rzutami kostką do gry. Wykonaj doświadczenie, w którym zmniejszysz do 50 liczbę rzutów w danej próbie. Oblicz najbardziej prawdopodobną liczbę sukcesów dla poszczególnej liczby oczek. Przeanalizuj wyniki z 10 kolejnych prób i porównaj z obliczoną wartością, a następnie z wynikami dla $n = 1000$ z przykładu. Jakie wnioski można wyciągnąć z takiego porównania?
2. Ułóż arkusz symulujący wielokrotny rzut monetą. Ustal $n = 50$, a następnie $n = 2000$. Policz liczbę wystąpień orłów i reszek. Dla jakiej wartości n będą one podobne?
3. Przeanalizuj doświadczenia opisane w podręcznikach do fizyki lub na stronach edukacyjnych. Wybierz jedno, którego wyniki można opracować za pomocą arkusza. Napisz, jak utworzyłbyś/utworzyłabyś taki arkusz i na co powinieneś/powinnaś zwrócić baczniejszą uwagę.

PODSUMOWANIE LEKCJI

s. 37

• Arkusz może służyć jako narzędzie do **symulacji różnych zjawisk, np. losowych**, których wyniki zostaną wykorzystane w dalszych badaniach.

s. 41

• **Wyniki doświadczeń** można wpisać do tabeli arkusza i analizować.

s. 42

• **Tworzenie wykresów** znacznie ułatwia analizę danych uzyskanych w czasie doświadczenia. Wzbogaca także **dokumentację** doświadczeń.



6. Edytor grafiki w pracy zawodowej, czyli tworzymy reklamę

NA TEJ LEKCJI:

- przypomnisz sobie, na czym polega stosowanie warstw w edytorach grafiki;
- prześledzisz proces umieszczania elementów graficznych w projekcie reklamy;
- prześledzisz proces przygotowania fragmentu grafiki do wklejenia do zdjęcia.

W swojej przyszłej działalności zawodowej będziesz musiał/musiła zadbać o reklamę. Można to zrobić na wiele sposobów. Twój potencjalny klient powinien dostać prosty i czytelny przekaz zachęcający do zainteresowania się tym, co masz do zaoferowania. Jednym ze środków przekazu jest odpowiednio wykonane i opracowane zdjęcie z nałożonymi elementami grafiki i napisami, które można wydrukować w ulotce reklamowej lub umieścić na stronie internetowej, w mediach społecznościowych itp. Do dzieła.

6.1. Podstawy są ważne, czyli utrwalamy wiedzę o warstwach

Zanim przystąpisz do opracowywania grafiki reklamowej, przypomnij sobie działanie mechanizmu warstw, bez którego tworzenie twojego dzieła byłoby znacznie utrudnione.

Wyobraź sobie kilka przezroczystych folii, na których kolejno narysowano tło i fazy ruchu postaci. Jeśli na tło położymy warstwę (folię) z pierwszą fazą ruchu i sfotografujemy całość, otrzymamy początkowy kadr filmu animowanego (rys. 6.1.). Teraz wystarczy powtórzyć to dla następnych etapów poruszania się bohatera. Dokładnie takiej metody używali twórcy filmów rysunkowych przed wprowadzeniem animacji komputerowych.

warstwy



Rys. 6.1. Kadr z filmu wykonany z wykorzystaniem warstw



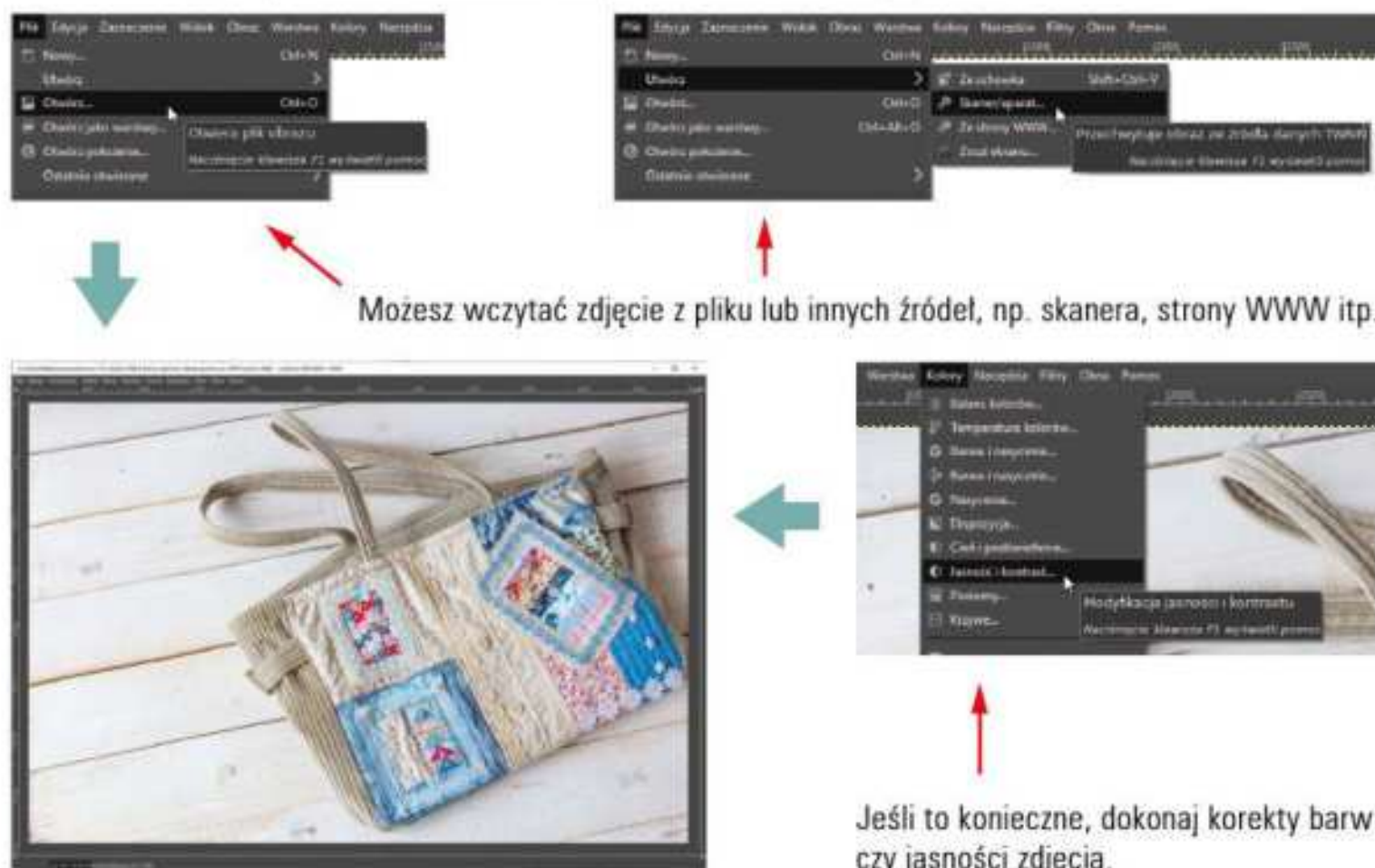
Wyobraź sobie teraz, że tłem jest zdjęcie wyrobu lub tło dla reklamy usługi. Dzięki znajomości mechanizmu warstw możesz nakładać na nie napisy czy elementy graficzne, np. logo firmy itp.

6.2. Klika warstw, czyli jak przygotować grafikę reklamową

Do wykonania ulotki lub grafiki reklamowej będzie ci potrzebny edytor grafiki, zdjęcie i dobry pomysł. Oczywiście najłatwiej opanować narzędzie. Trudniej opracować projekt reklamy. Przejrzyj strony internetowe i przypatrz się uważnie statycznym reklamom lub banerom reklamowym. Z jakich elementów się składają? Co jest istotne? Co zwraca twoją uwagę? Co zapamiętujesz po jej obejrzeniu? Wynotuj swoje spostrzeżenia i postaraj się je wykorzystać w tworzeniu własnej grafiki reklamowej. Przykład wykorzystania narzędzi edytora GIMP do realizacji projektu znajdziesz poniżej.

Licencja **GNU** pozwala na korzystanie z programu **GIMP** całkowicie bezpłatnie nawet do celów komercyjnych.

Rozpocznij od wczytania zdjęcia do edytora. W razie potrzeby popraw kadrowanie, jasność, kontrast i nasycenie kolorami (rys. 6.2.).



Rys. 6.2. Wczytywanie tła przykładowej reklamy do edytora GIMP

Jeżeli obraz został wczytany i okno warstw jest aktywne, możesz przystąpić do pracy. Jeśli dysponujesz znakiem firmowym, logo lub dodatkową fotografią, nałóż ten element na zdjęcie. Wykorzystaj do tego warstwy (rys. 6.3.; okno z warstwami otworzysz, wybierając kolejno z menu **Okna/Dokowalne okna dialogowe/Warstwy**). Najlepszy efekt uzyskasz, jeżeli tło dla nieregularnego kształtu nakładanej grafiki będzie przezroczyste.



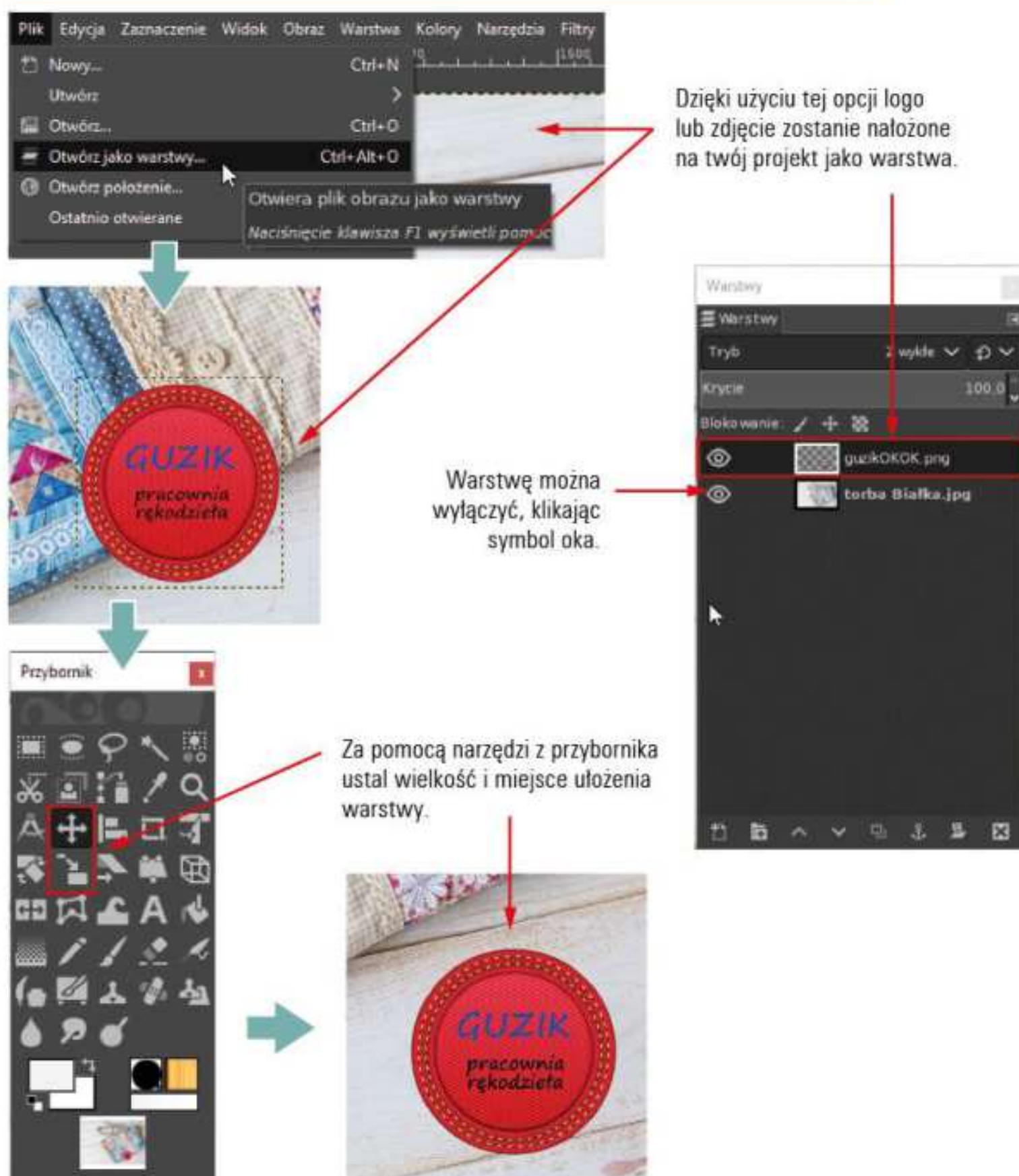
kanal alfa

Uzyskasz to dzięki zastosowaniu kanału alfa (przezroczystość) i zapisaniu pliku w formacie zachowującym przezroczystość, np. PNG. W opisany sposób możesz umieszczać na zdjęciu wszystkie rodzaje elementów graficznych.

W czasie pracy precyzyjnie operuj myszką.

Uwaga!

Jeśli przez niedokładność wykonasz niechcianą czynność, użyj **Ctrl+Z**. Wszystkie czynności są wykonywane na **aktywnej warstwie**.



Rys. 6.3. Wklejanie elementów graficznych z pliku jako warstw w GIMP

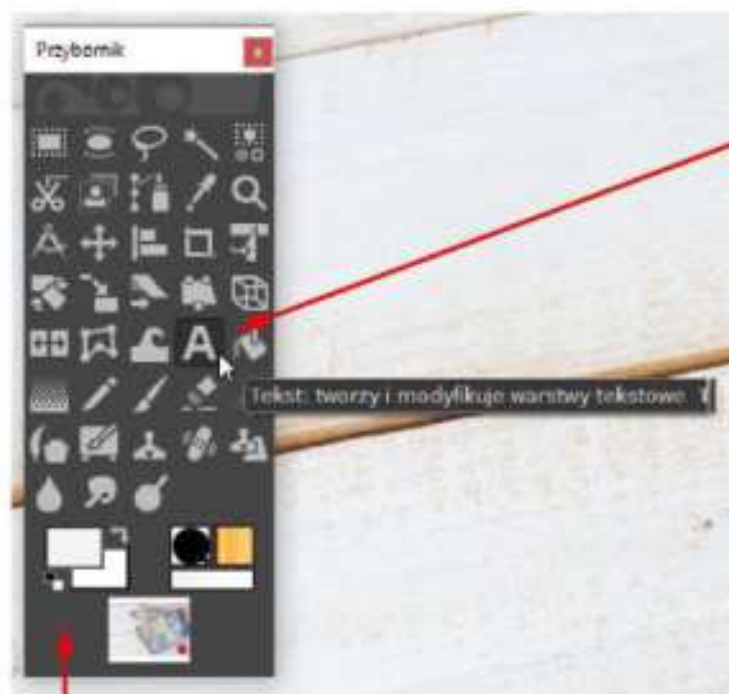


Uwaga!

W trakcie wykonywania projektu możesz używać wielu warstw. Te, które okażą się niepotrzebne w danej wersji, możesz wyłączyć, klikając symbol oka. Nie oznacza to, że warstwa została usunięta. W każdej chwili możesz ją przywrócić. W ten sposób można projektować równocześnie wiele wersji grafiki.

elementy
w oddzielnych
warstwach

Skoro wszystkie dodatkowe elementy graficzne są już na miejscu, możesz przystąpić do nakładania napisów. Każdy z nich będzie tworzył oddzielną warstwę, więc będzie podlegał tym samym prawom, co wklejony wcześniej element graficzny. Oprócz tego możesz użyć metod formatowania typowych dla edycji tekstu (rys. 6.4.).



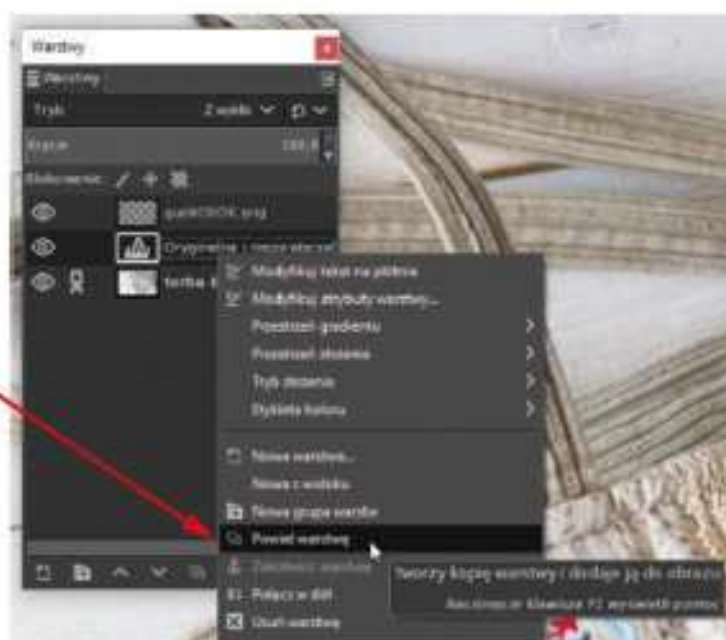
Wybierz narzędzie **Tekst** (ikona z literą A).

Wpisz tekst. Po zaznaczeniu znaków możesz formatować tekst. Dopasuj wielkość, krój i kolor czcionki.



Ustaw tekst w odpowiednim miejscu i pod odpowiednim kątem, używając narzędzi do przesuwania, powiększania i obracania warstw.

Jeśli planujesz więcej napisów, użyj mechanizmu powielania warstw. Zachowasz krój i formatowanie nowego tekstu.



Rys. 6.4. Wstawianie warstw z napisami w GIMP



Ukończone dzieło może zostać użyte w prezentacji, na stronie internetowej lub do drukowania ulotek reklamowych. Aby było to możliwe, powinno być zapisane w jednym z formatów graficznych powszechnie stosowanych (np. jpg lub png). Jeśli ulotki mają być drukowane w drukarni, uzgodnij format pliku wymaganego przez daną firmę (rys. 6.5.). Należy przy tym zachować odpowiednią jakość grafiki. Do publikacji na stronie można użyć kompresji, np. do pliku JPG, i zmniejszyć rozdzielczość. Do druku w większości przypadków przydatne będą formaty bezstratne, np. PNG lub TIF, i maksymalna możliwa rozdzielczość.

Uwaga!

Użycie opcji **Zapisz** spowoduje zapisanie pliku projektu z rozszerzeniem zgodnym z edytorem – dla **GIMP** jest to **xcf**. Do zapisu projektu w postaci pliku graficznego użyj opcji **Wyeksportuj**.



Rys. 6.5. Gotowa grafika reklamowa



6.3. Z dwóch jedno, czyli montujemy zdjęcia

Założmy, że masz do wykonania tło dla reklamy agencji turystycznej. Klient określił, że na tle egzotycznej plaży ma się znaleźć postać młodego człowieka. Oczywiście wyjazd zagraniczny w celu zrobienia zdjęć jest mało opłacalny. Jak sobie poradzić?

kupowanie zdjęć

Po pierwsze, należy legalnie zakupić zdjęcia w jednym z serwisów internetowych (rys. 6.6.). Po drugie, trzeba się posłużyć edytorem grafiki i odpowiednio zmontować ilustrację.



Rys. 6.6. Przykładowe zdjęcia zakupione w serwisie internetowym

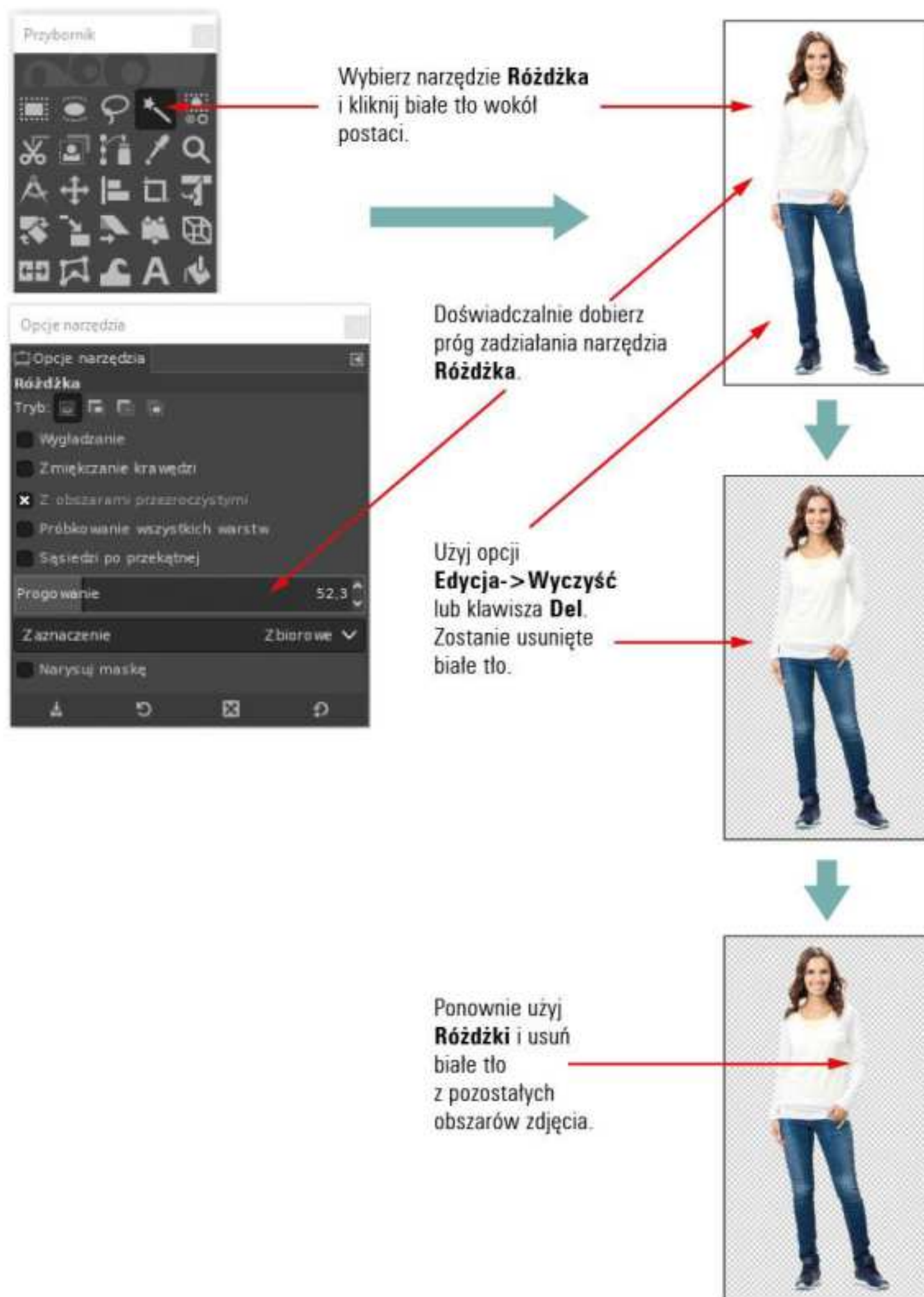
Do ćwiczenia możesz użyć plików pobranych z serwisów z darmową grafiką, np. pixabay.com. Pobierz obraz z obiektem, który został sfotografowany na białym, jednolitym tle. Dzięki temu możliwe będzie szybsze wycięcie postaci za pomocą jednego z narzędzi, np. **Różdżki** w GIMP. W przypadku typowego zdjęcia postaci na innym tle w wycinanie trzeba włożyć o wiele więcej pracy.

Opisany poniżej przykład pozwoli ci prześledzić lub przećwiczyć proces montażu zdjęcia lub grafiki.

Otwórz pobrane zdjęcie z postacią w edytorze i postępuj według opisu (rys. 6.7.).

Uwaga!

Kanał alfa dodaje do rysunku przezroczystość. Pliki graficzne, np. zdjęcia, zazwyczaj zawierają informację na temat kolorów, np. w modelu RGB jest to 8 bitów na każdy z nich. Po dodaniu kanału alfa każdy punkt zyskuje dodatkowe 8 bitów informacji o poziomie jego przezroczystości. Mamy wtedy do czynienia z 32-bitową głębią kolorów.



Rys. 6.7. Usuwanie tła wokół postaci



Uwaga!

Jeśli po użyciu opcji czyszczenia zaznaczonego obszaru zostanie wycięta postać zamiast tła, użyj opcji **Zaznaczenie/Odwróć**.

Jeśli po użyciu opcji czyszczenia zaznaczonego obszaru tło przybierze kolor biały zamiast wypełnienia szarą kratką (przezroczystość), dodaj **Kanał alfa (Warstwa/Przezroczystość)**.

Jeśli tło nie jest jednorodne lub zdecydowanie jaśniejsze (ciemniejsze) od wycinanego obiektu, odręcznie zaznacz dane obszary. Po operacji wycinania postaci musisz usunąć pozostałości tła, np. wokół włosów (rys. 6.8.).



Pozostałości tła są widoczne po powiększeniu podglądu.

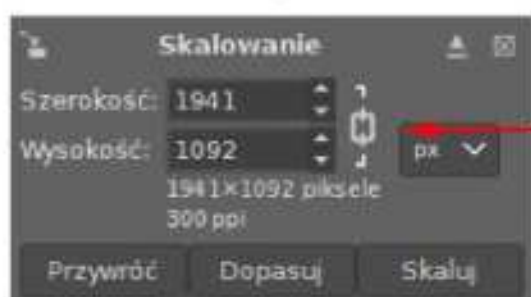
Użyj narzędzia **Gumka**.



Rys. 6.8. Poprawianie niedoskonałości wycinania **Różdżką**

Pracuj dokładnie i cierpliwie, aż do uzyskania dobrego rezultatu. Parametry Gumki możesz zmieniać. Dobierz je tak, by usuwanie niedoskonałości nie zniekształcało postaci.

Czas na montaż. Wczytaj do edytora GIMP zdjęcie z plażą, a następnie, jako nową warstwę, plik z wyciętą postacią. Wczytana postać zapewne nie będzie proporcjonalnie dopasowana do tła. Wystąpi efekt jej zawieszenia nad piaskiem plaży. Dopasuj zdjęcie, używając narzędzia **Skalowanie** (rys. 6.9.).



Konieczn ie ustaw skalowanie proporcjonalne, by unikn ąć zniekształcenia postaci.



Postać wklejona w całości, pozbawiona cienia, sprawia wrażenie unoszącej się w powietrzu. Montaż jest zbyt widoczny.



Wklejenie postaci w innym planie (powiększenie lub pomniejszenie i przemieszczenie) pozwala uniknąć problemów z cieniem i nienaturalnością położenia.

Rys. 6.9. Efekt końcowy montażu zdjęcia



Zauważ, że postać jest równo oświetlona ze wszystkich stron. Nie jest to naturalne na nasłonecznionej plaży. Ten efekt możesz usunąć, nakładając maskę lub warstwę z półprzezroczystym tłem. Mimo tych niedoskonałości i niewielkiej ilości pracy włożonej w przygotowanie zdjęcia, efekt jest zadowalający.

Zdjęcie jest gotowe do nanoszenia innych elementów związanych z reklamą, np. napisów, logotypów itp.

ZADANIA DO WYKONANIA

1. Znajdź w internecie serwisy oferujące zdjęcia i grafikę komputerową. Porównaj je pod względem zawartości, bogactwa oferty i ceny. Sprawdź, jakie są warunki licencyjne zakupionych plików, np. ile razy można ich użyć, w jakiej rozdzielczości itp. Zwróć uwagę na prawa związane z RODO.
2. Wykonaj baner reklamowy szkoły zapraszający na jej drzwi otwarte. Użyj mechanizmów opisanych w rozdziale. Wykorzystaj swoje umiejętności i doświadczenia z poprzednich lat nauki. Wielkość banera i rozdzielczość dostosuj do wymagań strony internetowej, na której zostanie umieszczony.
3. Korzystając z edytora grafiki z mechanizmem operowania na warstwach, zaprojektuj okładkę do płyty swojego ulubionego muzyka lub zespołu. Weź pod uwagę, że okładka ma co najmniej dwie strony.
4. W edytorze tekstu opisz proces wycinania fragmentu zdjęcia za pomocą różnych narzędzi programu GIMP. Zwróć uwagę na parametry tych narzędzi, które możesz zmieniać w oknie dialogowym. W dokumencie umieść zrzuty ekranowe.

PODSUMOWANIE LEKCJI

s. 43

- **Warstwy** są podstawowym mechanizmem tworzenia grafiki, a także korekty i montażu zdjęć. Na warstwie można umieścić każdy element graficzny.

s. 45

- Zastosowanie elementów graficznych z **kanalem alfa** (przezroczystością) zapewni zachowanie kształtu wklejanego detalu. Kanał alfa jest znamieny dla 32-bitowej głębi kolorów, w której 8 bitów opisuje poziom przezroczystości każdego punktu.

s. 46

- **Skalowanie i przemieszczanie** warstw to podstawowe czynności edycyjne grafiki, skuteczne po umieszczeniu ich w **oddzielnych warstwach**.

s. 48

- Przygotowując elementy graficzne, należy pamiętać o prawach autorskich. W razie potrzeby można **zakupić zdjęcie lub rysunek w serwisie internetowym**.



7. Reklama jest ważna, czyli jak wykonać atrakcyjną prezentację

NA TEJ LEKCJI:

- poznasz darmowe programy do tworzenia prezentacji;
- prześledzisz proces tworzenia prezentacji w zależności od jej przeznaczenia;
- prześledzisz proces tworzenia prezentacji w chmurze Prezi.

Jednym z najpopularniejszych programów do tworzenia prezentacji jest MS PowerPoint. W programie szkoły podstawowej było wiele ćwiczeń wykorzystujących bogate możliwości tego programu. Programów o podobnej funkcjonalności jest więcej. Część z nich akceptuje format dokumentów PowerPoint. W tym rozdziale poznasz darmowy program LibreOffice Impress, który jest częścią dużego, darmowego pakietu biurowego LibreOffice, oraz aplikację sieciową działającą w chmurze Prezi.

7.1. Projekt jest ważny, czyli jak rozpocząć pracę nad prezentacją

Od tego, jak wykonasz swoją prezentację, może zależeć twój sukces. Bez względu na to, czy będzie przygotowana na lekcję w szkole, konferencję naukową czy dla działu handlowego firmy, musi być wykonana według przemyślanego scenariusza i z zastosowaniem odpowiednich środków. **Kluczem do sukcesu jest dostosowanie jej do grupy docelowej odbiorców i użycie atrakcyjnej formy.** O ile forma zależy od możliwości programu, który wybierzesz, to scenariusz jest już wynikiem pracy autora. Rozpoczynając pracę nad prezentacją, możesz wziąć pod uwagę kilka czynników (rys. 7.1.).



Rys. 7.1. Czynniki wpływające na jakość scenariusza prezentacji



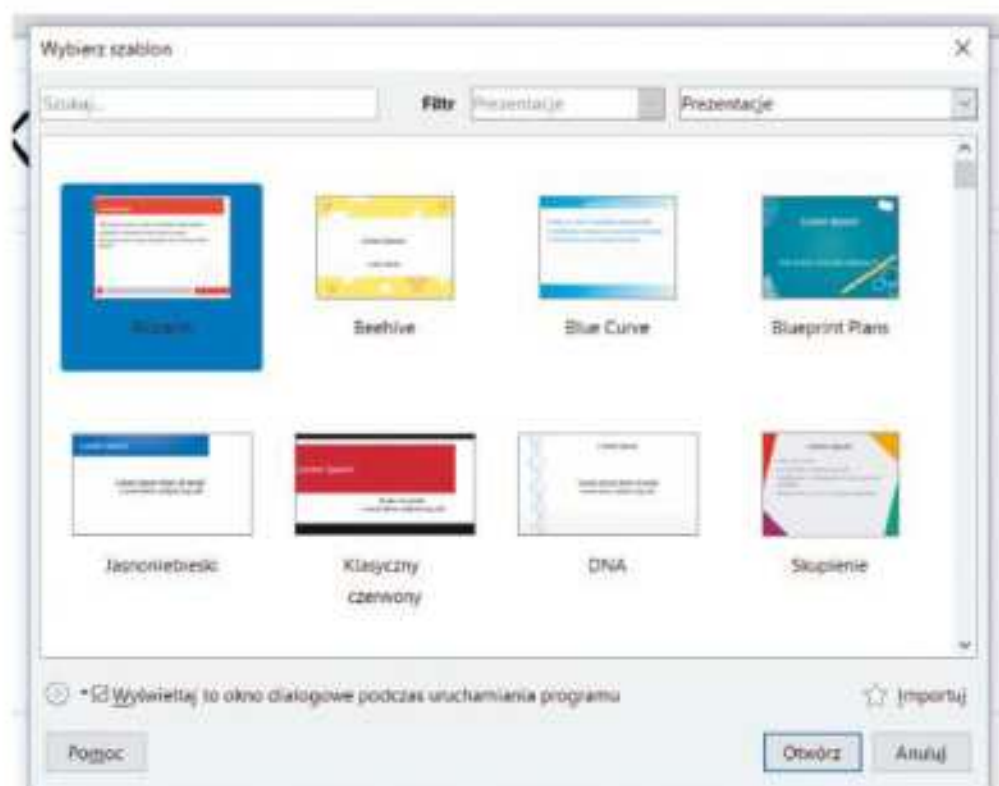
Prezentacja przygotowana do **ilustrowania** wykładu powinna być inaczej zbudowana niż przygotowana na ten sam temat, ale przeznaczona do samodzielnej nauki.

Pierwsza z nich powinna zawierać grafiki, zdjęcia, rysunki i filmy ilustrujące to, o czym mówi prelegent. W prezentacji dotyczącej architektury antycznej, w czasie której pokazujesz zdjęcia budowli z odpowiednio zaznaczonymi fragmentami, nie musisz opowiadać o kształtach czy materiałach użytych do budowy. W przypadku przygotowywania materiału do samokształcenia (np. zamiast notatek w zeszycie) nie można poprzestać na zdjęciach i filmach. Konieczne jest także użycie tekstu. Bez odpowiedniej informacji słownej taka prezentacja nie będzie skuteczna. Odbiorca jedynie przejrzy zdjęcia. Ten sam temat, ci sami odbiorcy, ale różne przeznaczenie, a więc potrzebne są różne scenariusze prezentacji.

7.2. Alternatywne rozwiązania, czyli nie tylko PowerPoint

Jeśli chcesz korzystać z darmowych programów do tworzenia prezentacji, musisz wybrać takie, które spełnią twoje wymagania i jednocześnie pozwolą zapisywać slajdy w popularnych formatach (np. pptx, pdf, jpg). Jednym z nich jest LibreOffice Impress z pakietu biurowego LibreOffice. Znajdziesz go na stronie pl.libreoffice.org.

Do wykonania prezentacji możesz wykorzystać jeden z wielu szablonów (rys. 7.2.) lub utworzyć własny i zapamiętać go w programie.



Rys. 7.2. Szablony dostępne w programie LibreOffice Impress

Po wybraniu szablonu możesz przystąpić do tworzenia prezentacji według ustalonego scenariusza. Dzięki ikonom na przyciskach z łatwością odnajdziesz podstawowe opcje niezbędne do rozpoczęcia edycji (rys. 7.3.).



Rys. 7.3. Podstawowe menu programu Impress

Zanim przystąpisz do edycji, zdecyduj o pewnych jej cechach, np. o proporcjach obrazu. Jeśli twoje urządzenie wyświetlające ma proporcje 16:9, wybierz właśnie taki format (rys. 7.4.). Okna dialogowe umieszczone z prawej strony ekranu pozwalają na ustalenie pozostałych parametrów przygotowywanych prezentacji, np. przejść między slajdami, tła, animacji itp. (rys. 7.5.).



Rys. 7.4. Okno do ustalania parametrów prezentacji

Wybierz **Właściwości** dla właściwości obiektu. Jeśli żaden z obiektów nie jest zaznaczony, zmienisz właściwości slajdu.

Jedna z ważniejszych opcji okna dialogowego, pozwala na edycję tekstu.

Służy do określenia przejść pomiędzy slajdami oraz ich parametrów.

W trakcie edycji możesz zmienić szablon i wygląd pierwszego slajdu.



Zaznacz obiekt i dodaj parametry animacji.

Tu możesz wybrać elementy graficzne, np. strzałki, i dodać je do slajdu.

Rys. 7.5. Menu okna dialogowego programu Impress

Menu programu jest intuicyjne i tylko od pomysłowości i doświadczenia autora zależy końcowy efekt.



7.3. Dla klienta i na lekcję, czyli różne scenariusze na ten sam temat

Założmy, że pracujesz w firmie produkującej komponenty do budowy samochodów. Jesteś menadżerem zajmującym się skrzyniami biegów. Oprócz tego prowadzisz także zajęcia z praktykantami ze szkoły branżowej. Dla swoich klientów musisz przygotować prezentację reklamującą części do skrzyni biegów. Druga prezentacja powinna wyjaśnić praktykantom zasady działania przekładni mechanicznych. Mimo użycia tych samych rysunków i zdjęć, scenariusze obu prezentacji będą znacznie się różniły.

Scenariusz prezentacji oferty handlowej

Slajd 1 startowy:

- tło związane z techniką,
- najważniejsze informacje o firmie oraz jej logo,
- tytuł prezentacji.

Slajd 2:

- po kolei, po kliknięciu, pojawiają się nazwy firm, z którymi współpracuje nasze przedsiębiorstwo.

Slajd 3:

- prezentacja wyrobu, zdjęcie przekroju skrzyni biegów, kolejno, po kliknięciu, pojawiają się parametry.

Slajd 4:

- po kliknięciu pojawiają się po kolei argumenty, dlaczego klient powinien kupić części właśnie u nas,
- na przykładzie przekładni rowerowej pokazane przyspieszenie, jakie osiągnie klient dzięki współpracy z nami.

Scenariusz prezentacji na zajęcia z praktykantami

Slajd 1 startowy:

- tło związane z techniką,
- najważniejsze informacje o firmie oraz jej logo.

Slajd 2:

- działanie przekładni rowerowej,
- zdjęcie i krótki opis.

Slajd 3:

- przekładnia rowerowa na filmie (odnośnik do YouTube).

Slajd 4:

- wzór i przykład obliczeń przekładni,
- rysunek pomocniczy.

Slajd 5:

- ogólna budowa samochodowej skrzyni biegów,
- ilustracja.

Slajd 6:

- oferta pracy w firmie.

7.4. On-line i na dysku, czyli realizujemy scenariusze w różnych środowiskach

Wybór programu do realizacji prezentacji można oprzeć na kilku kryteriach:

1. Koszty, jakie należy ponieść.

Istnieją dobre programy, za które nie musisz płacić, nawet jeśli używasz ich w firmie. Przed zainstalowaniem należy sprawdzić dokładnie, z jaką licencją masz do czynienia. Istnieją bowiem bezpłatne programy, które są takie tylko dla użytkowników domowych.



2. Musisz zdecydować, czy będziesz projektować prezentację on-line, czy lokalnie na swoim komputerze. Wersja on-line zazwyczaj zapewnia także na dysku chmury miejsce dla twoich prezentacji.
3. Kompatybilność z systemem operacyjnym lub innymi edytorami prezentacji. Jeśli zależy ci na zgodności plików (np. z MS PowerPoint), wybierz program, który ma możliwość importu i eksportu plików w tym formacie.
4. Jeśli potrzebujesz prezentacji, która działa także na smartfonie, przed zakupem lub instalacją sprawdź, czy działa niezawodnie.

W trakcie ćwiczeń możesz użyć edytora Impress z pakietu LibreOffice i Prezi. Pierwszy jest całkowicie darmowy i należy go zainstalować w systemie. Drugi to darmowy jedynie do zastosowań domowych program działający on-line.

Zrealizujemy prezentację według scenariusza nr 1.

Pobierz i zainstaluj program LibreOffice.

Po uruchomieniu programu Impress pojawi się okno z szablonami. Możesz wybrać któryś z nich, jeśli będzie pasował do tematu twojej prezentacji. Do opisanego wyżej scenariusza prezentacji handlowej utworzymy własny (rys. 7.6. i rys. 7.7.).

Kliknij przycisk **Właściwości** i wybierz opcję **Właściwości**.

Ustal proporcje slajdu.

Wybierz **Wstaw obraz**, aby wczytać własne tło slajdu.

Ustaw obraz tła

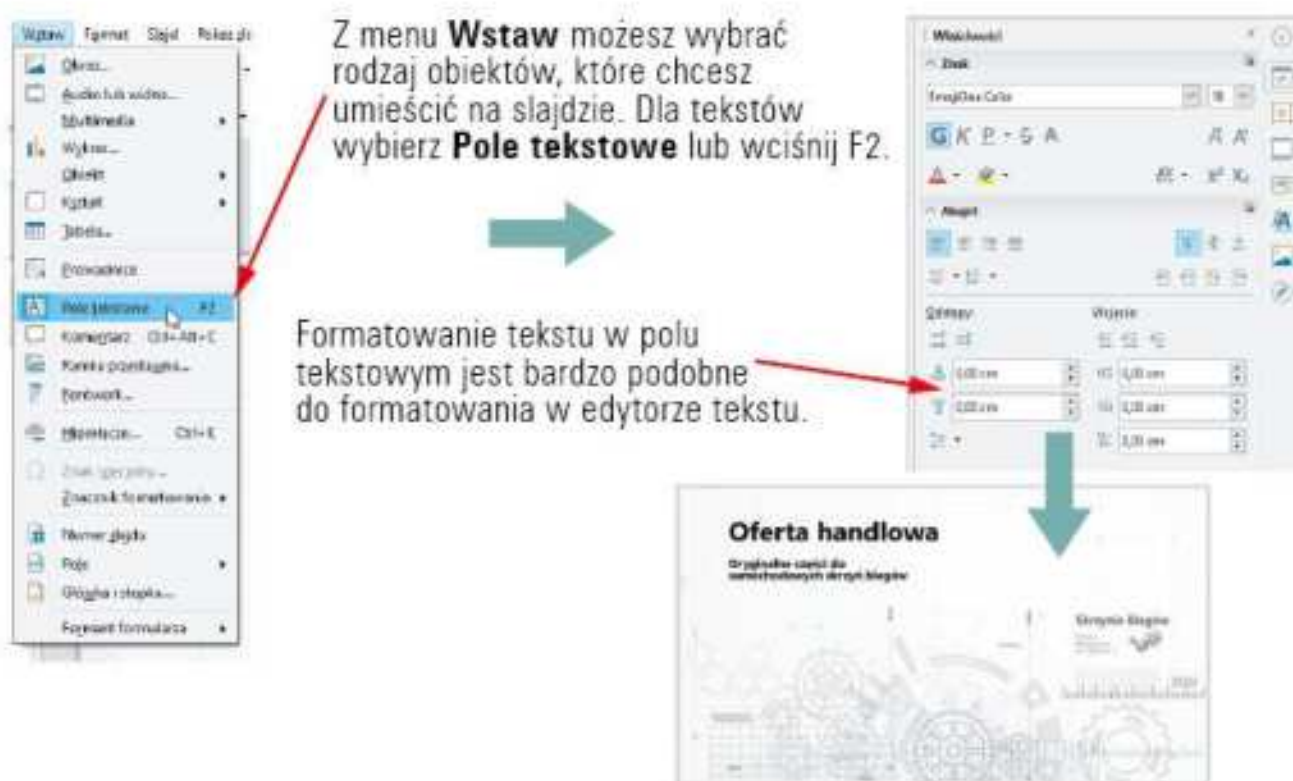
Wybierz plik graficzny z tłem. Kliknij dwukrotnie miniaturę.

Rys. 7.6. Ustalanie własnego tła slajdów



I. ROZWIĄZYWANIE PROBLEMÓW Z WYKORZYSTANIEM KOMPUTERA...

Korzystając z narzędzi, umieść obiekty na pierwszym slajdzie według scenariusza (rys. 7.7.).



Rys. 7.7. Wstawianie pola tekstowego i formatowanie tekstu

Teraz możesz dodać nowy slajd (rys. 7.8.) i nadać mu tło.



Rys. 7.8. Dodawanie nowego slajdu do prezentacji

Taki układ warto zapisać jako szablon (rys. 7.9.).



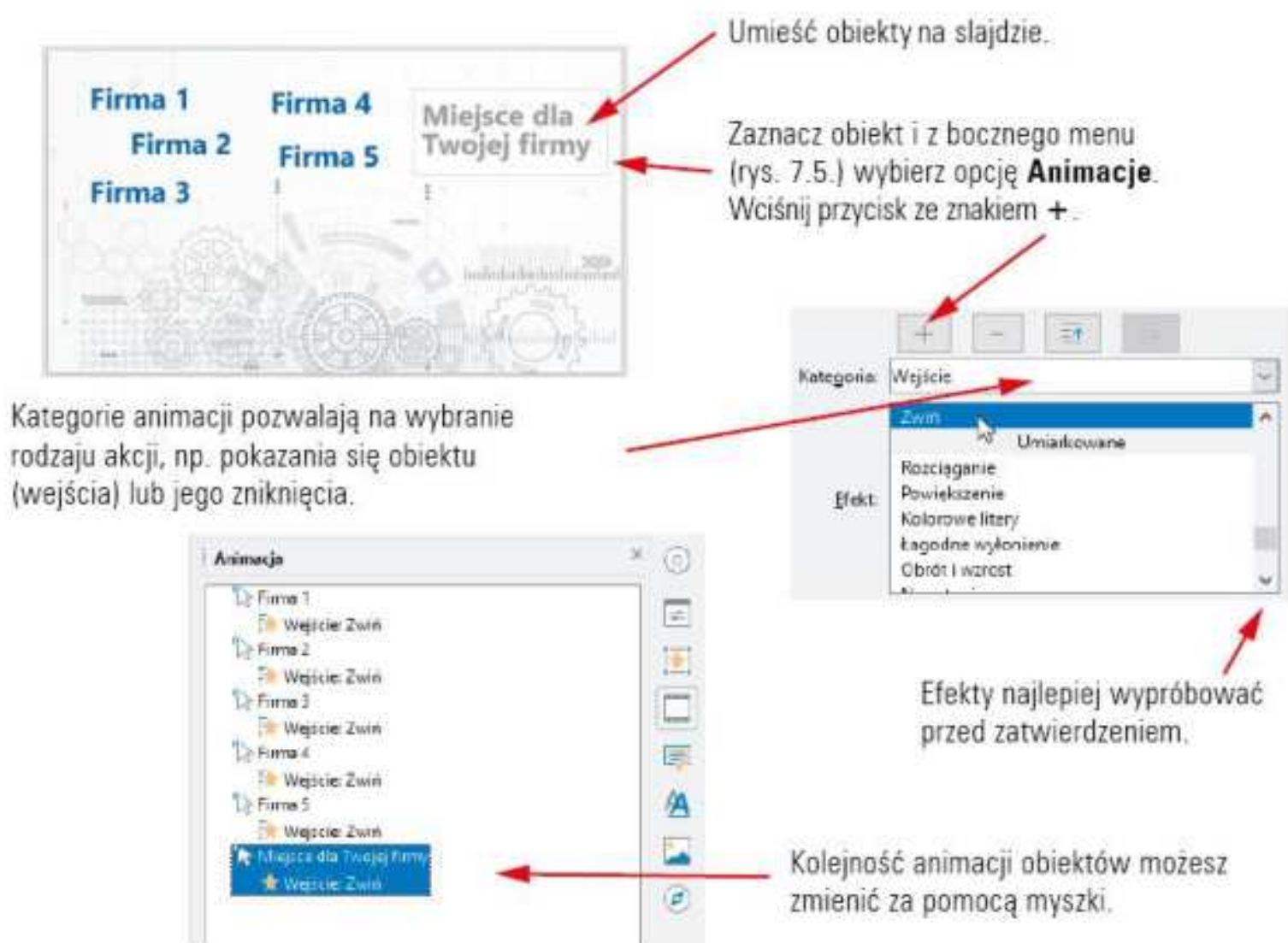
Rys. 7.9. Tworzenie nowego szablonu



Tworzenie szablonu nie jest niezbędne do projektowania prezentacji. Przyda się jednak, jeśli chcesz w firmie ujednolicić materiały multimedialne, którymi posługują się pracownicy. To ważne szczególnie w przypadkach częstego tworzenia prezentacji firmowych. Dzięki własnemu formatowi będą oryginalne i jednocześnie utrzymane w tym samym stylu.

Czas na realizację drugiego punktu scenariusza. Mamy tu do czynienia z jednym slajdem, a obiekty (napisy) pojawiają się po wciśnięciu klawisza.

Pracę rozpocznij od dodania nowego slajdu. Następnie rozmieść na jego powierzchni nazwy firm lub ich logo. Do każdego z obiektów dodaj animację (rys. 7.10.).



Rys. 7.10. Dodawanie animacji obiektów

Trzeci i czwarty slajd ze scenariusza można wykonać w taki sam sposób.

Uwaga!

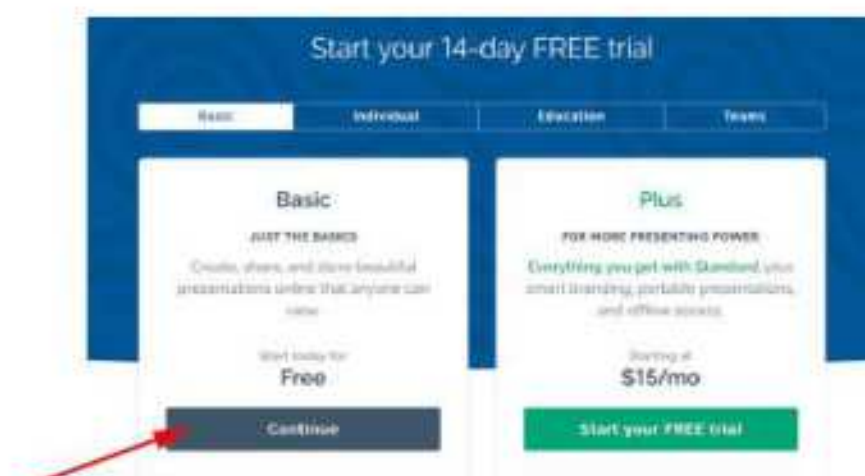
Nie wszystkie efekty animacji i przejść są kompatybilne z podobnymi w PowerPoint. W przypadku eksportu mogą nie działać.

Pamiętaj, by zapisywać co pewien czas wyniki swojej pracy.

Jeśli swoją prezentację chcesz umieścić na stronie internetowej, możesz ją wyeksportować do postaci HTML.



Drugi scenariusz zrealizujemy w chmurze prezī.com. Pierwszą czynnością jest założenie konta. Do ćwiczeń i prywatnego użytku wybierz darmową opcję (rys. 7.11.).

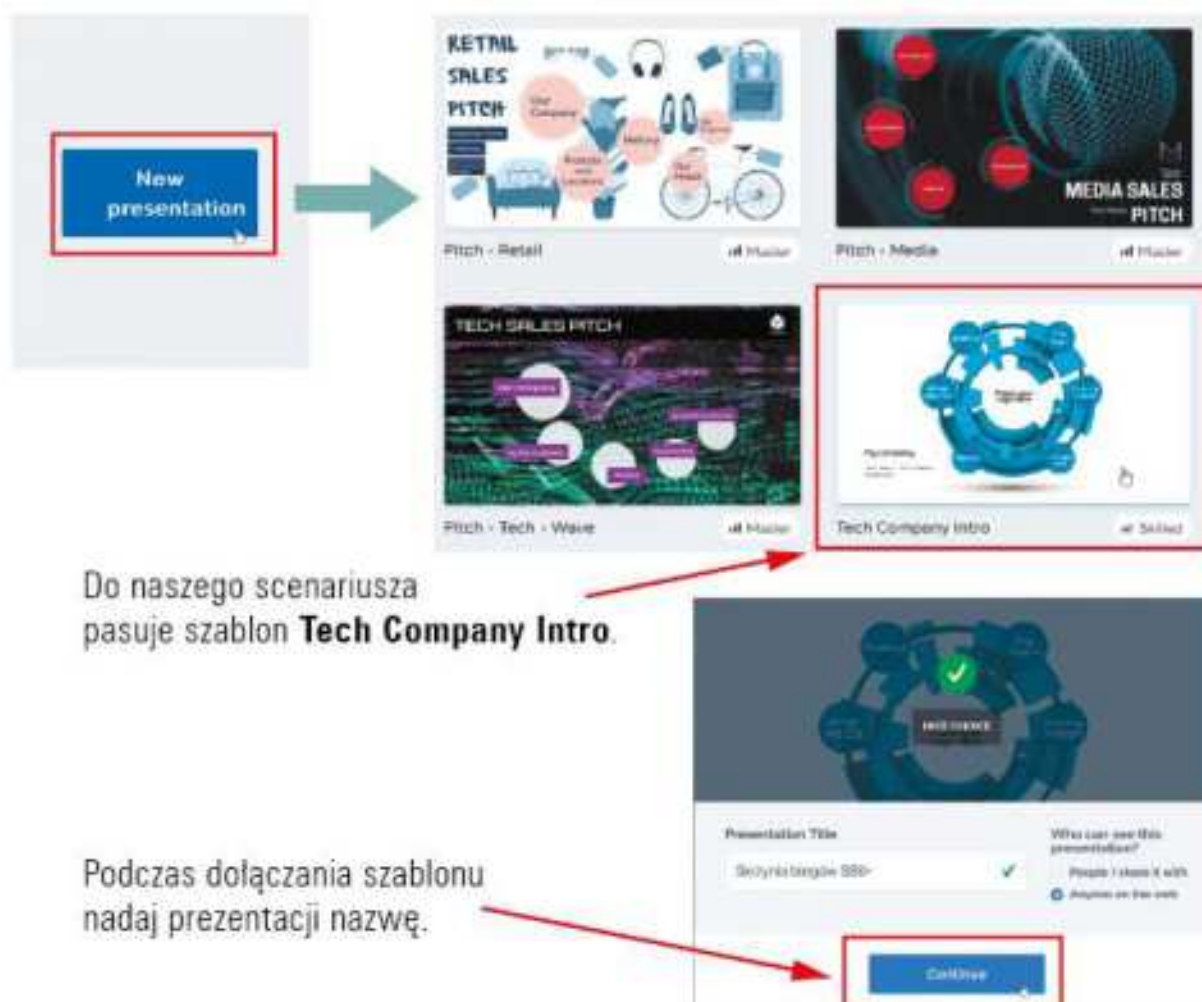


Do ćwiczeń i użytku domowego wybierz opcję **Basic**.
Dalej postępuj zgodnie z opisem.

Rys. 7.11. Wybór opcji programu prezī.com

Masz już konto. Jego zasobami możesz zarządzać podobnie jak każdym dyskiem. Jeśli planujesz wykonanie wielu prezentacji, utwórz foldery, nazwij je i przechowuj w nich swoje prace. Unikniesz chaosu.

Prace nad nową prezentacją rozpocznij od przycisku **New presentation**, a następnie zdecyduj, czy wykorzystasz gotowy szablon, czy stworzysz go od podstaw (rys. 7.12.).



Do naszego scenariusza
pasuje szablon **Tech Company Intro**.

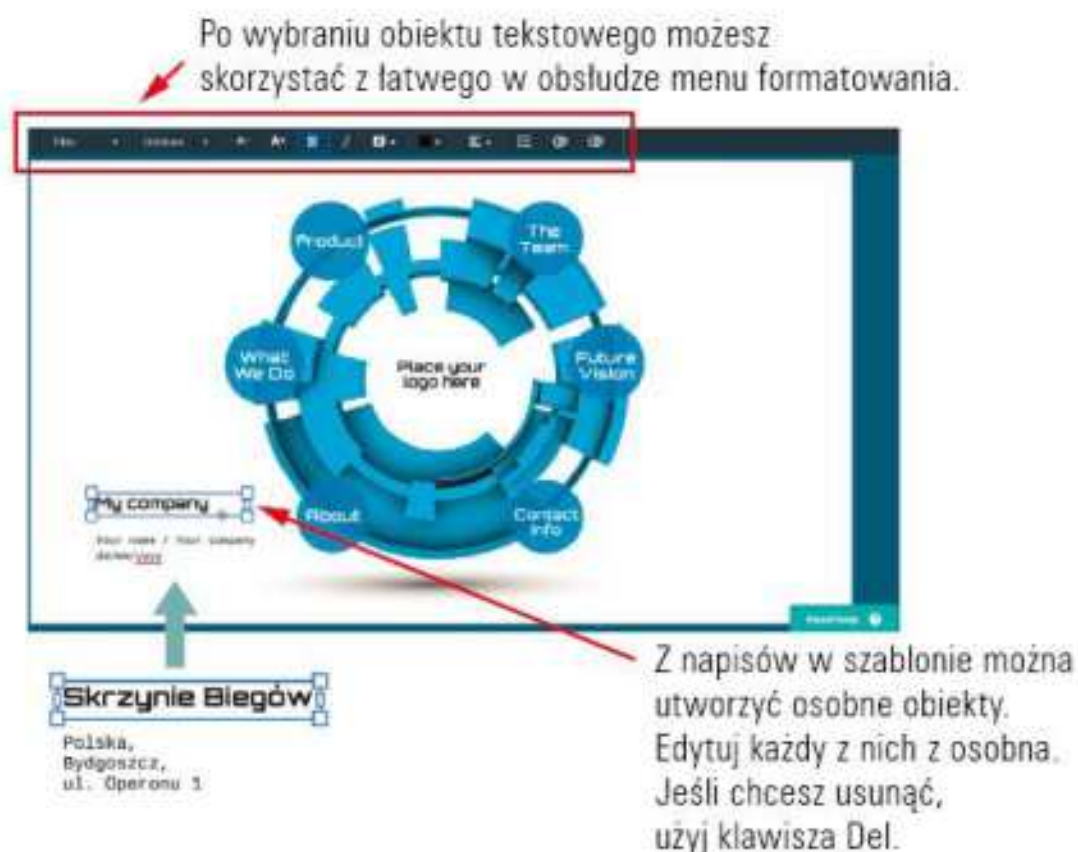
Podczas dołączania szablonu
nadaj prezentacji nazwę.

Rys. 7.12. Tworzenie nowej prezentacji na podstawie szablonu



Ponieważ tytuły poszczególnych obiektów nie odpowiadają naszemu scenariuszowi (nr 2), musisz je zmienić. Edytor działa intuicyjnie. Wystarczy kliknąć obiekt, by został wybrany i ukazało się menu kontekstowe.

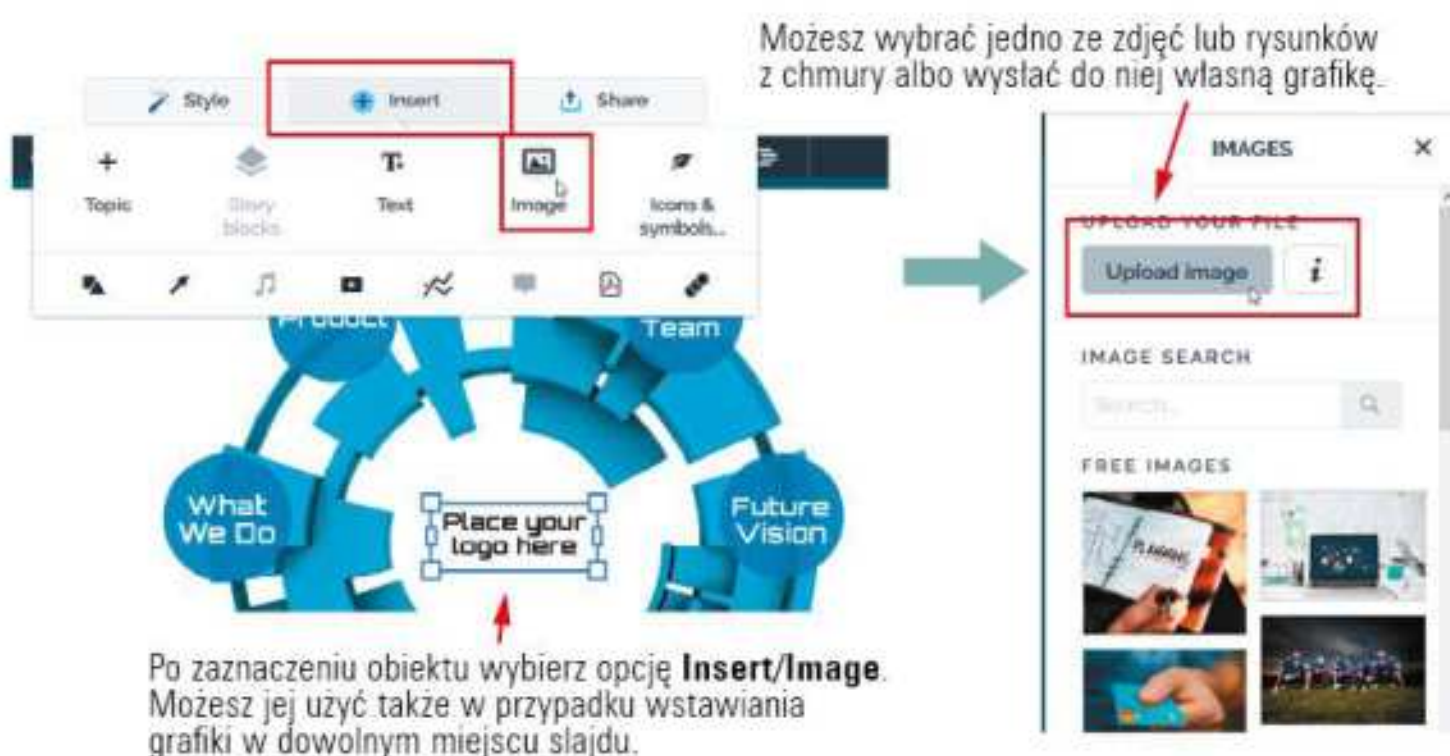
Zmiany rozpocznij od nazwy firmy (rys. 7.13.).



Rys. 7.13. Edytowanie obiektu tekstowego

Każdy z obiektów możesz oczywiście przemieszczać, naprowadzając na niego kursor myszy, a następnie przytrzymując lewy jej przycisk.

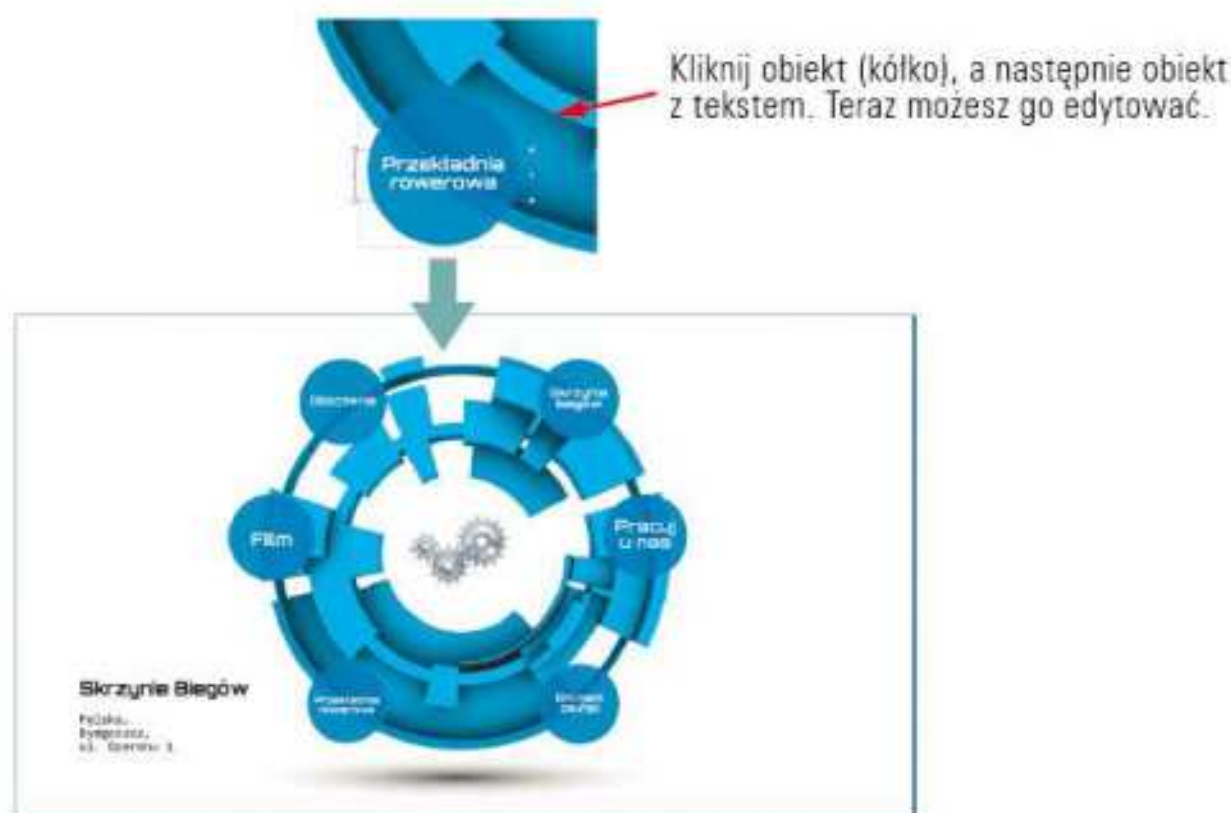
W centrum planszy możesz umieścić logo firmy. Kliknij obiekt, a następnie wykorzystaj opcję wstawiania grafiki (rys. 7.14.).



Rys. 7.14. Wstawianie grafiki do slajdu w chmurze prezio.com



Teraz możesz zmienić tytuły poszczególnych slajdów umieszczonych w kołach szablonu. Postępuj podobnie jak na rysunku. Po tych czynnościach główny slajd będzie już gotowy (rys. 7.15.).



Rys. 7.15. Edycja tytułów slajdów

Czas na edycję pozostałych slajdów i realizację scenariusza nr 2.

Kliknij dwukrotnie obiekt **Przekładnia rowerowa**. Twój slajd zostanie animowany i powiększony na cały ekran. Użyj menu **Insert** do umieszczenia na nim obiektów zgodnych ze scenariuszem (rys. 7.16.).



Rys. 7.16. Slajd z obiektami tekstowymi i zdjęciem



Slajdy można animować. Wypróbuj wszystkie rodzaje animacji. Według scenariusza ten slajd ma pomóc w zrozumieniu działania przekładni rowerowej. Należy więc zwrócić uwagę odbiorcy na niektóre szczegóły. Jeśli wybierzesz typ animacji Zoom, to wirtualna kamera wykona zbliżenie danego obiektu (rys. 7.17.). Wypróbuj inne rodzaje animacji obiektu.



Rys. 7.17. Dodawanie animacji obiektu na slajdzie

Zapoznaj się z tabelą 7.1. dotyczącą porównania programów.

Lp.	Impress	prezi.com
1.	Aplikacja instalowana w systemach operacyjnych (Windows, Linux, Mac OS). Wymaga odpowiedniej ilości miejsca na dysku, a korzystanie z niej jest ograniczone do jednego komputera.	Aplikacja internetowa działająca w chmurze pod adresem prez.com. Nie obciąża zasobów komputera. Możliwa jest praca na każdym komputerze z przeglądarką i łączem internetowym.
2.	Prezentacje można eksportować do innych formatów, np. PowerPoint.	Prezentacje w wersji darmowej można prezentować wyłącznie on-line.
3.	Prezentacje można uruchamiać na innych komputerach oraz zapisywać w formatach tekstowych, np. PDF. Jest to przydatne w procesie przygotowywania się do wystąpień z użyciem danej prezentacji.	Prezentacje są dostępne dla innych użytkowników Prezi (dla wersji darmowej). Można także udostępniać bezpośredni adres prezentacji niezalogowanym użytkownikom (także użytkownikom smartfonów). Pobieranie prezentacji wraz z odtwarzaczem jest możliwe wyłącznie w wersji komercyjnej.



Lp.	Impress	prezi.com
4.	Wyświetlanie prezentacji polega na liniowym (jeden za drugim) wyświetlaniu slajdów ze wszystkimi zastosowanymi opcjami, np. animacjami, przejściami itp.	Wyświetlanie prezentacji jest podobne do filmu, w którym kamera porusza się po planie, wykonuje zbliżenia, dodaje ruch.
5.	Oba edytory obsługują się intuicyjnie. Impress jest aplikacją bardzo rozbudowaną i poznanie wszystkich jej opcji jest czasochłonne. Jednak poznanie podstawowych opcji pozwala na tworzenie atrakcyjnych prezentacji. Prezi nie ma rozbudowanych narzędzi. Przygotowanie prezentacji odbiega nieco od zasad stosowanych w większości edytorów. Wymaga intuicyjnego wykorzystania języka filmowego, a w kreowaniu kolejnych slajdów pomaga zbliżenie myślenia do pracy kamery w konkretnych ujęciach.	

Tab. 7.1. Porównanie edytorów prezentacji

ZADANIA DO WYKONANIA

1. Dokonaj internetowego przeglądu programów do edycji prezentacji. Te, które uznasz za najciekawsze, mogą stanowić inspirację. Wykonaj prezentację z użyciem Impress lub Prezi, w której przedstawisz wybrane edytory. Użyj różnych form przekazu, np. film z sieci, zrzuty ekranu, odnośniki itp.
2. Utwórz szablon prezentacji w LibreOffice Impress, który posłuży ci do tworzenia tutoriali informatycznych. Tło i inne grafiki możesz wykonać samodzielnie lub użyć grafiki znalezionej w sieci. Pamiętaj o tym, by przestrzegać prawa. Upewnij się, czy są one udostępnione na zasadzie licencji CC. Szablon zapisz w pliku i umieść w swojej chmurze. Sprawdź, czy możesz go użyć w edytorze chmury Google.
3. Wypróbuj opcje edytorów, które nie zostały omówione w podręczniku, a są warte użycia. Wykonaj prezentację na ich temat.
- 4*. W chmurze prez.com wykonaj szablon prezentacji. Zachowaj go. Utwórz prezentację (tutorial) opisującą przepis na wykonanie własnego szablonu.
- 5*. Po zalogowaniu się do Prezi uruchom jedną ze swoich prezentacji. Skopiuj do schowka jej adres z paska adresów przeglądarki. Wyszukaj w sieci stronę, która oferuje darmowe generowanie kodów QR. Wygeneruj kod dla adresu prezentacji. Zapisz go na dysku w formacie pliku graficznego. Wyświetl na ekranie monitora i zeskanuj za pomocą aplikacji w telefonie, np. QR Droid. Sprawdź, czy prezentacja zostanie uruchomiona w wybranej przez siebie przeglądarce.

PODSUMOWANIE LEKCJI

s. 53

- **Prezentację** zawsze należy dostosowywać do grupy docelowej odbiorców, a także używać atrakcyjnej formy graficznej.



8. Prezentacja wideo, czyli jak przygotować prezentację filmową

NA TEJ LEKCJI:

- przypomnisz sobie podstawowe zasady montażu wideo;
- wykorzystasz zasady tworzenia prezentacji i scenariusza do planowania prezentacji wideo;
- poznasz programy do rejestracji obrazu z ekranu komputera.

Zapewne aktywnie korzystasz z portali internetowych. Większość z nich osiąga zyski dzięki reklamom. Część reklam to banery, które już umiesz wykonać. Inne to krótkie filmy reklamowe. Przygotowują je profesjonalne pracownie, ale postęp technologiczny i rozwój narzędzi informatycznych spowodował, że i ty możesz spróbować swoich sił i zrealizować ciekawą reklamę wideo. W ten sam sposób można wykonać także prezentację lub tutorial.

8.1. Warto przypomnieć, czyli zasady pomagają w realizacji filmu

Filmowcy znają wiele zasad, których przestrzeganie pomaga w realizacji projektów filmowych. Oczywiście nie są one obowiązkowe i w uzasadnionych przypadkach się je łamie. Zaczynając pracę z filmem, warto jednak je poznać i na nich budować pierwsze sceny.

1. Po ułożeniu scenariusza opracuj storyboard, czyli narysuj każde ujęcie sceny. Zaznacz na nim ruch, ustawienie bohaterów i inne istotne elementy. Dobrze opracowanie storyboardu gwarantuje sprawną realizację zdjęć.

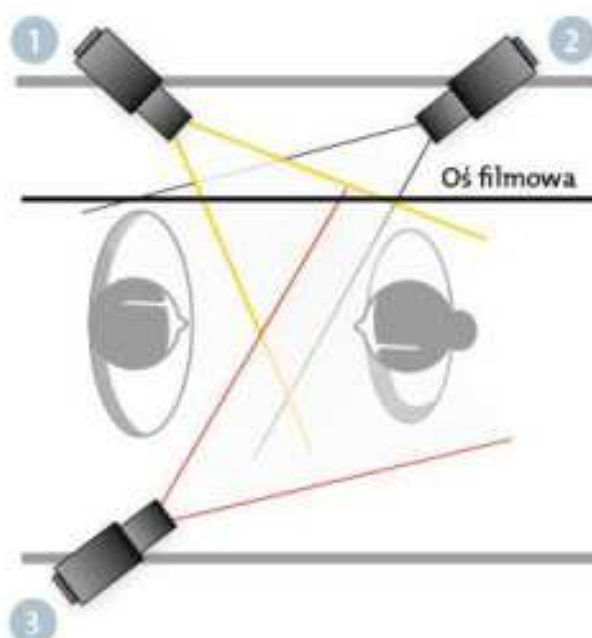
scenariusz
i storyboard

2. Nie łam osi filmowej i zachowuj kierunki akcji.

oś filmowa

Jeśli nie będziesz przestrzegać tej zasady, w filmowaniu niektórych scen może dojść do błędnego odczytania przez widza całej sytuacji. Przykładem może być dialog dwojga bohaterów (rys. 8.1.).

Rys. 8.1. Przykład złamania osi filmowej podczas realizacji sceny dialogu



Oś filmowa nie jest przekroczona. W kolejnych ujęciach widz widzi, że postaci są zwrócone do siebie twarzami i toczą ze sobą dialog.

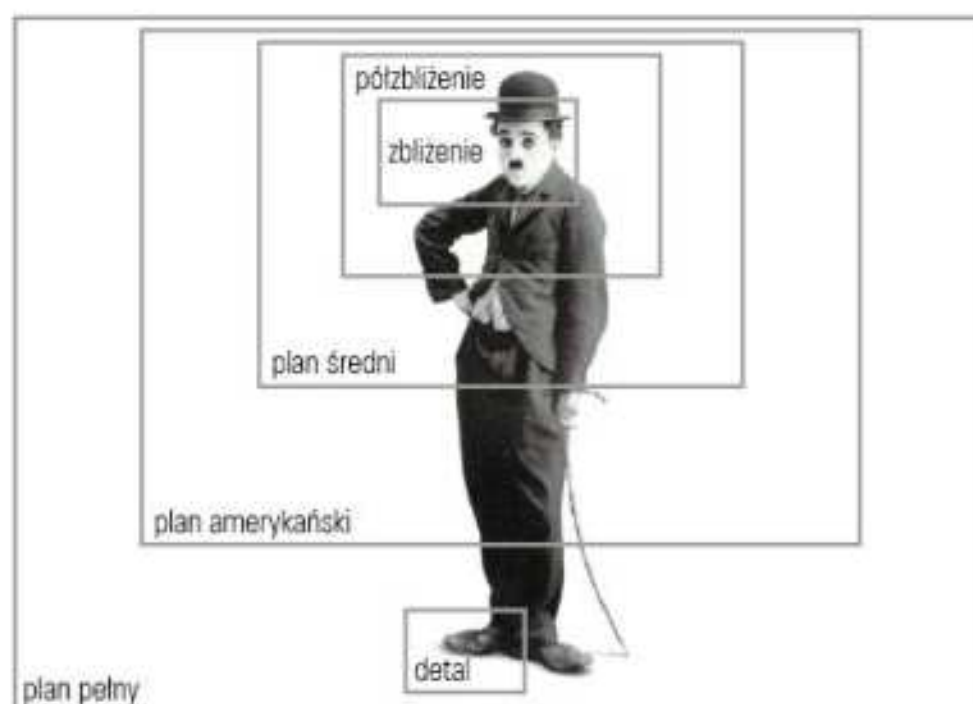


Oś filmowa jest przekroczona (kamera 3). Widz odnosi wrażenie, że w scenie uczestniczy trzecia osoba, a bohaterowie nie prowadzą dialogu ze sobą.



Jeżeli oś filmową złamiesz w scenie pościgu, to widz odniesie wrażenie, że bohaterowie poruszają się naprzeciw siebie i dojdzie do ich spotkania.

3. W ujęciach staraj się stosować złoty podział. Dzięki temu pomożesz widzowi zidentyfikować najważniejsze obiekty. Starannie dobieraj tło sceny, by nic nie rozpraszało skupienia widza na głównych bohaterach. Unikaj tzw. bohaterów drugiego planu, jeśli nie przewidziałeś/przewidziałaś tego w scenariuszu.
4. Starannie dobieraj sposób oświetlenia planu. Światło w ujęciach pełni jedną z najważniejszych funkcji. Jeśli wykorzystasz je kreatywnie, możesz zmienić nastrój i klimat ujęcia, a w konsekwencji wpłynąć na odbiór sceny.
5. Podczas realizacji zdjęć weź pod uwagę późniejszy montaż sceny. Dbaj nie tylko o nieprzekraczanie osi filmowej, lecz także o umiejętny dobór planów filmowych. Staraj się, by kolejne ujęcia dzieliła odległość przynajmniej dwóch planów filmowych (rys. 8.2.).



Rys. 8.2. Plany filmowe w odniesieniu do postaci

Te i inne zasady są szeroko opisywane w internecie. Jeśli jesteście zainteresowani tworzeniem filmów, znajdziecie w nich wiele ciekawych i pożytecznych porad. Korzystajcie z wiarygodnych i sprawdzonych źródeł.

8.2. Zamiast kamery, czyli rejestrujemy zawartość ekranu narzędziem systemu Windows

Coraz częściej w prezentacjach, instrukcjach lub samouczkach wykorzystuje się nagrania z czynności wykonywanych na ekranie komputera. O ile do zapisania nieruchomych obrazów można użyć narzędzia **Wycinanie** z Windows lub klawisza **PrintScreen** (obraz jest zapamiętany w schowku), to zapis wideo wymaga od nas więcej zaangażowania.

System Windows 10 zawiera aplikację ułatwiającą nagrywanie przebiegu gry komputerowej. Domyślnie ta opcja może być wyłączona, dlatego należy to



sprawdzić w **Ustawienia > Granie > Pasek gry** (rys. 8.3.). Uruchamianie nagrywania następuje po użyciu kombinacji klawiszy **Win+Alt+R**.

Pasek gry

Określ, jak ma być otwierany pasek gry i jak ma rozpoznawać grę

Rejestruj klipy z gry oraz zrzuty ekranu i emituj je za pomocą paska gry

☒ Włączone

Rys. 8.3. Aktywowanie paska gry w Windows 10

Rejestrator nagrywa wyłącznie akcję w oknie danej aplikacji. Jego obsługa jest intuicyjna (rys. 8.4.). Program ten ma bardzo wiele opcji i możliwości konfiguracji. Więcej informacji na jego temat znajdziesz w pomocy systemu Windows.



Rys. 8.4. Menu rejestratora obrazu z systemu Windows

Innym rozwiązaniem jest użycie łatwych w obsłudze programów, które pozwalają także na zapis akcji na pulpicie, wyznaczonego obszaru czy wybranego okna. Możliwe jest także zapisanie ruchu kursora i sygnalizowanie wciśnięcia klawisza myszy. Niektóre rejestrują także obraz z zewnętrznych źródeł, np. kamery.

8.3. Inne narzędzia dla PC, czyli wykorzystujemy darmowe narzędzia rejestrujące obraz

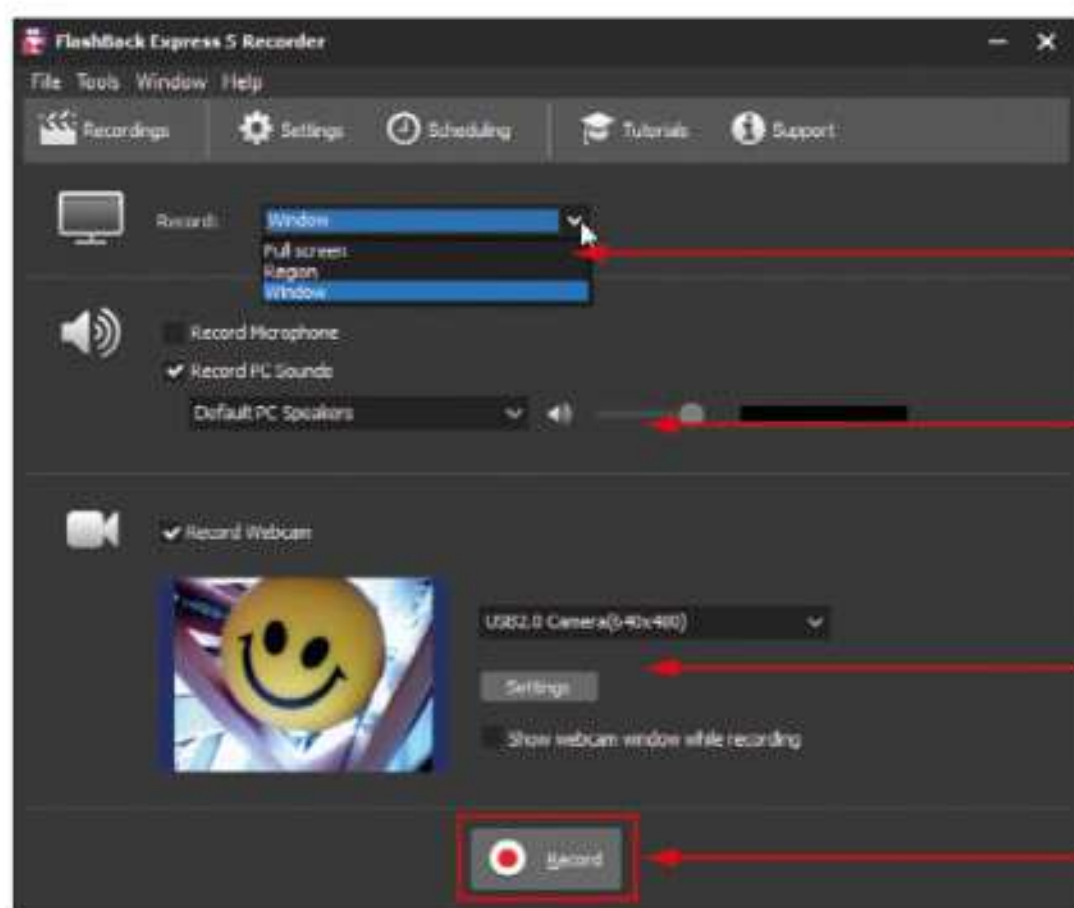
Wykorzystując wyszukiwarki, można znaleźć wiele darmowych narzędzi do rejestracji obrazu pochodzącego z różnych źródeł. Do wykonania samouczka, prezentacji itp. najbardziej przydatne będą te z opcjami płynnej rejestracji zawartości ekranu lub danego okna aplikacji. Przykładem mogą być **FlashBack Express Recorder** oraz **Nagrywanie ekranu** z systemu Android. Metody pracy z nimi są typowe dla bardzo wielu podobnych aplikacji, dlatego nie będziesz mieć problemu z korzystaniem z innych programów tego typu.



Uwaga!

Podczas pobierania i instalacji darmowych programów zwróć uwagę na propozycje instalowania także aplikacji dodatkowych, np. programów antywirusowych. Zazwyczaj domyślnie jest zaznaczona zgoda. Możesz to zmienić. W czasie użytkowania darmowych programów możesz także spotkać reklamy lub zachęty do zakupu ich pełnej wersji. Jeśli nie chcesz z tych ofert korzystać, z rozważą wybieraj opcje.

Po uruchomieniu programu **FlashBack Express Recorder** możesz od razu przystąpić do nagrywania (rys. 8.5.). Podstawowe opcje ustawień obrazu i dźwięku znajdziesz w okienku głównym aplikacji. Możesz rejestrować zawartość ekranu, okienka, zaznaczonego obszaru lub obraz z kamery.



Tu wybierz, jaki obszar ekranu ma być rejestrowany.

Możesz wybrać źródło dźwięku i poziom jego zapisu. Warto przeprowadzić kilka prób, by był on ustawiony optymalnie.

Jeśli planujesz użycie kamery, ustaw rozdzielczość, z jaką chcesz rejestrować obraz. Możesz także zmienić ustawienia zaawansowane.

Kliknij i rozpocznij rejestrację.

Rys. 8.5. Okno programu FlashBack Express Recorder w wersji 5

Zanim jednak przystąpisz do pracy z jakąkolwiek aplikacją rejestrującą obraz, warto przejrzeć dokładnie opcje ustawień (**Settings**). Wśród wielu z nich znajdziesz takie, które mogą poprawić jakość twojej rejestracji lub w pełni osiągnąć zamierzony efekt.

Każdy program rejestrujący musi działać w tle, by jego okno nie było widoczne na filmie. Warto więc zapamiętać kombinację klawiszy kończącą nagrywanie. Dla FlashBack jest to **Ctrl+S**.

Jeśli opanujesz pracę z opisanym programem, obsługa innych rejestratorów nie będzie już dla ciebie problemem.



8.4. Puk, puk, czyli rejestrujemy obraz z ekranu telefonu

Skąd taki tytuł podrozdziału? W wielu komórkach z systemem Android zastosowano specyficzny mechanizm otrzymywania zapisu zawartości ekranu. Należy dwukrotnie pukać palcem w ekran. Plik JPG zostanie automatycznie zapisany w odpowiednim folderze. Łatwo go znaleźć w systemie danego telefonu.

W starszych smartfonach można np. jednocześnie przytrzymać **Home** i przycisk zasilania lub przycisk zasilania i dolny klawisz regulacji głośności.

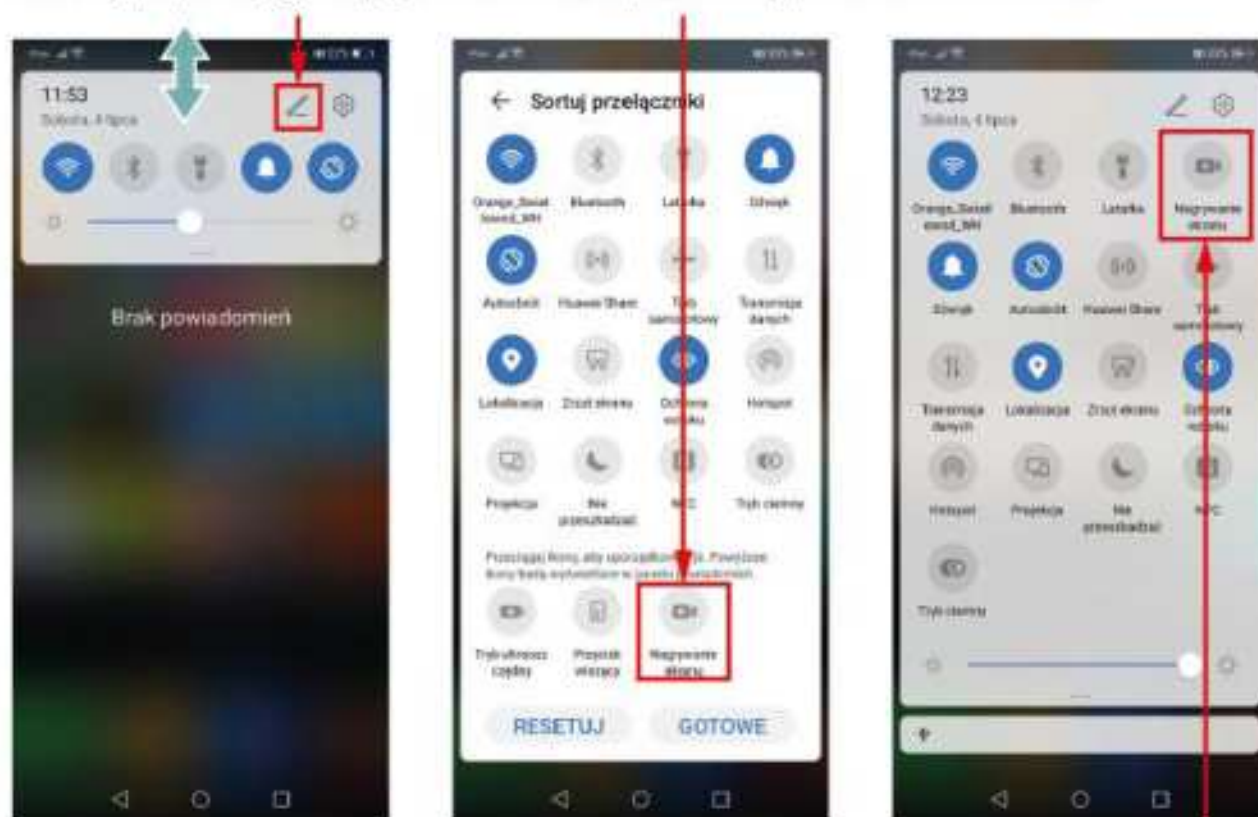
Jeśli żaden z powyższych sposobów nie zadziała, rozwiązania problemu należy poszukać w instrukcji obsługi telefonu lub na forach dyskusyjnych, blogach itp.

Co zrobić, aby zarejestrować film pokazujący działania na ekranie? Wielu użytkowników instaluje w tym celu zewnętrzne programy. Często żądają one dostępu do bardzo wielu zasobów telefonu. Mogą więc być źródłem potencjalnego zagrożenia. Systemy Android od wersji 7 mają wbudowaną aplikację, która zapisze w pliku wszystko, co się dzieje na ekranie. Na próżno jednak szukać dostępu do niej wśród ikon aplikacji. Na szczęście, jest inny sposób (rys. 8.6.).

rejestracja obrazu
w komórce

Rozwiń menu podręczne i dotknij przycisk jego edycji.

Zgodnie z sugestią systemu przeciągnij ikonę **Nagrywanie ekranu** na wybrane miejsce wśród ikon menu.



Rejestrację uruchomisz, wybierając ikonę **Nagrywanie ekranu**.

Rys. 8.6. Dostęp do aplikacji **Nagrywanie ekranu** w systemie Android

Natychmiast po uruchomieniu aplikacji rozpoczyna się rejestracja. W lewym górnym rogu ekranu jest wyświetlany czas nagrania i przycisk do jego zatrzymania. Plik z filmem jest dostępny w pamięci telefonu.



Jeśli korzystasz z telefonu z system iOS, postępuj podobnie. Przejdź do ustawień systemu i otwórz **Centrum Sterowania**, a następnie **Dostosowanie Narzędzi**. Tam znajdziesz ikonę aplikacji **Nagrywanie ekranu** i znak **+** (plus). Użyj go, by dodać aplikację do menu podręcznego.

Nagrane filmy możesz rozpowszechniać w sieci lub pobrać do komputera i montować z innymi klipami.

Jeśli opanujesz sposoby nagrywania zawartości ekranu komputera lub telefonu, możesz tworzyć prezentacje filmowe lub uzupełniać nimi inne dokumenty, np. PowerPoint lub Prezi.

ZADANIA DO WYKONANIA

1. Ułóż scenariusz prezentacji filmowej (tutoriala) omawiającej proces instalacji wybranej aplikacji rejestrującej obraz. Zapisz go w postaci punktów. Prezentacja przeznaczona jest do publikacji w sieci.
2. Przetestuj działanie aplikacji rejestracji obrazu z systemu Windows 10. Zapisz na dysku przykładowy krótki film zarejestrowany tym programem.
3. Wykonaj prezentację wideo opisującą proces przygotowania systemu do użycia wbudowanego w system Windows 10 oprogramowania do rejestracji obrazu.
4. Sprawdź w swoim telefonie, jaki zastosowano w nim sposób uruchamiania rejestracji zrzutów ekranowych i rejestracji wideo z ekranu. Zarejestruj krótki film, np. pokazujący fragment rozgrywki w twojej ulubionej grze. Jeśli korzystasz z portali społecznościowych, prześlij go w to miejsce. Pamiętaj o zasadach bezpieczeństwa i zachowania prywatności.
- 5*. Ułóż scenopis (scenariusz zapisany jako opis kolejnych scen) prezentacji filmowej (tutoriala) omawiającej proces instalacji wybranej aplikacji rejestrującej obraz. Prezentacja przeznaczona jest do publikacji w sieci. Zapisz scenopis w postaci storyboardu, czyli rysunków pokazujących kolejne ujęcia scen filmu. Jeśli chcesz dowiedzieć się więcej o tworzeniu storyboardu, poszukaj informacji w sieci, np. na stronach dla filmowców amatorów.

PODSUMOWANIE LEKCJI

s. 65

- Przed przystąpieniem do realizacji nagrania prezentacji wideo należy przemyśleć i ułożyć **scenariusz**, a na jego podstawie – **storyboard**.

s. 65

- Podczas nagrywania i montażu powinno się stosować ogólnie przyjęte zasady tworzenia filmu, w tym dbałość o **nieprzekraczanie osi filmowej**.

s. 69

- Zanim zainstalujesz zewnętrzny program, sprawdź, czy przeznaczona do rejestracji aplikacja dostarczona z systemem spełnia twoje wymagania. Jeśli tak, instalacja nie będzie konieczna. Do rejestracji **obrazu z telefonu komórkowego** możesz użyć aplikacji wbudowanej lub zewnętrznej.



9. Multimedia w prezentacji, czyli dźwięk i film na slajdach

NA TEJ LEKCJI:

- dowiesz się, jak umieszczać filmy w prezentacjach z uwzględnieniem dostępu do sieci;
- poznasz sposoby rejestracji komentarzy do slajdów prezentacji.

Twoje prezentacje są coraz lepsze i skuteczniejsze w swoim przekazie. Dzięki umiejętności układania scenariuszy i komponowania slajdów (np. zgodnie z zasadami trójpodziału), możesz tworzyć pomoce do wykładów, prelekcji i referatów, a także do pokazów, reklamy i e-learningu na coraz wyższym poziomie. Będą jeszcze atrakcyjniejsze, gdy dodasz do nich elementy multimedialne, np. filmy, których proces realizacji opisano w poprzednim rozdziale. Dołączenie dźwięków, filmów i animacji pozwala nie tylko uatrakcyjnić pokaz, lecz także zwiększyć skuteczność jego oddziaływania na odbiorcę. Duży wpływ na przygotowanie prezentacji ma także dostęp do sieci i możliwość zagnieżdżania multimediów pobieranych z internetu.

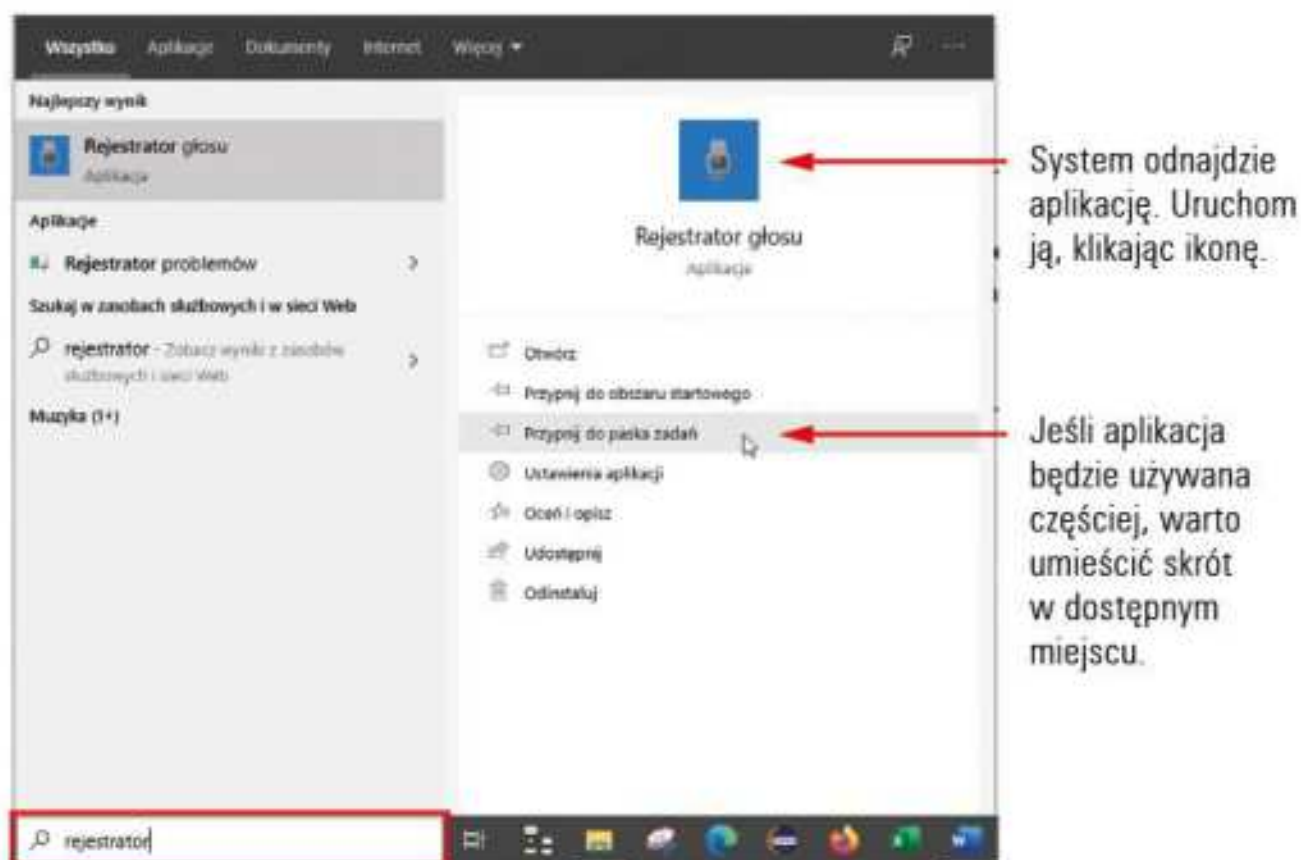
9.1. Komentarze do slajdów, czyli rejestrujemy dźwięk

W wielu aplikacjach do tworzenia prezentacji każdemu slajdowi można przypisać dźwięk. Uruchamianie odsłuchu może następować także po kliknięciu obiektu, np. ikony, fragmentu tekstu lub ilustracji. Zazwyczaj są to utwory muzyczne. Przy ich doborze musisz pamiętać o poszanowaniu praw autorskich i majątkowych. Zdarza się jednak, że chcemy dodać komentarz słowny do slajdu lub jego elementu. W tym celu musimy mieć możliwość zarejestrowania dźwięku. Może się także okazać, że konieczny będzie także montaż. Na szczęście nie jest to trudne.

Założmy, że zobowiązałeś/zobowiązałaś się do wykonania projektu polegającego na prezentacji jednego utworu literackiego. Jego omówienie planujesz ilustrować prezentacją. Plan przewiduje, że rozpoczniesz od zaprezentowania fragmentu utworu. Czytanie go w czasie lekcji wiąże się ze stresem. Dlatego postanowiłeś/postanowiłaś nagrać swoją interpretację i dołączyć ją do slajdu z tekstem. W tym celu możesz użyć **Rejestratora głosu** z systemu lub zewnętrznej aplikacji, np. darmowej **Audacity**.

Pierwsze rozwiązanie wydaje się łatwiejsze, ale jak to zwykle bywa, jest ono pozbawione możliwości edycji, np. miksowania z muzyką i dobierania parametrów nagrania. Do krótkich nagrań nadaje się jednak bardzo dobrze. Aplikację znajdziesz, używając wyszukiwania (rys. 9.1.).

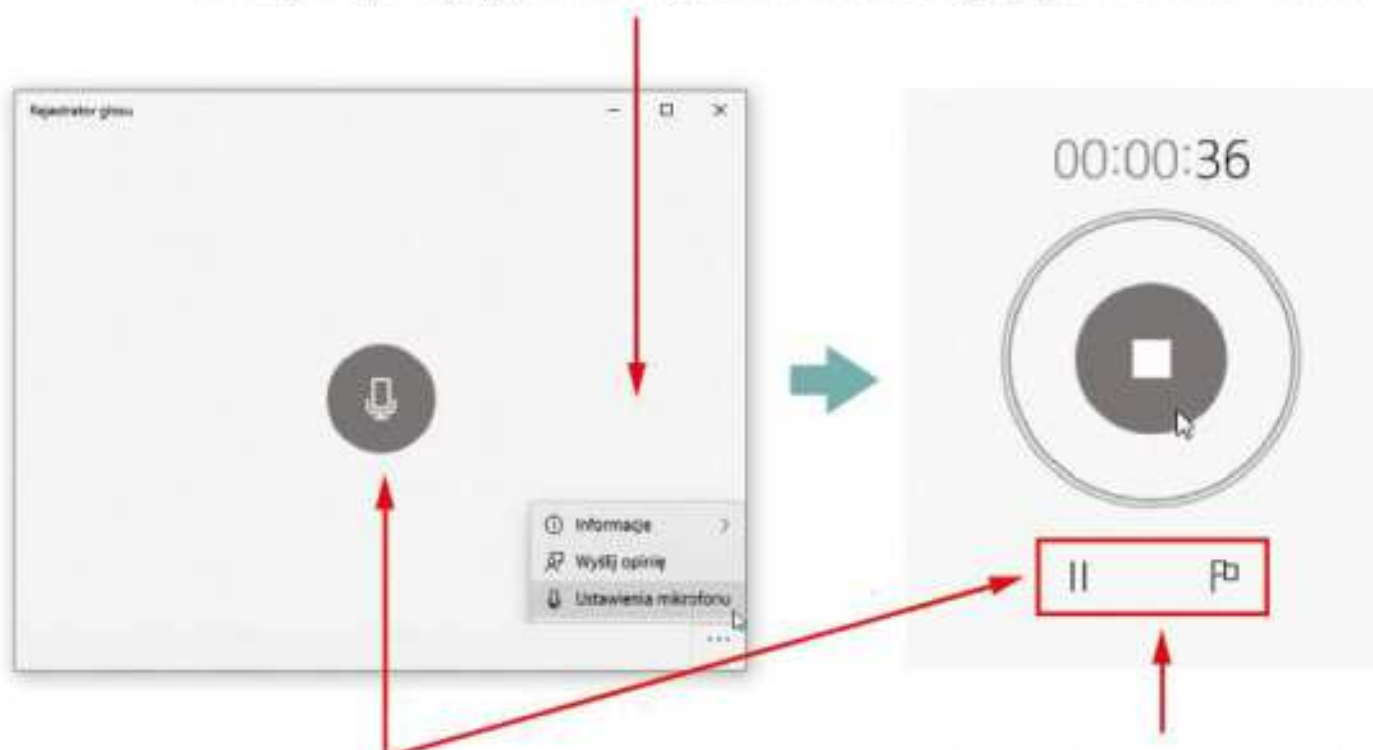
Rejestrator głosu



Rys. 9.1. Uruchamianie aplikacji **Rejestrator głosu** w Windows 10

Po uruchomieniu Rejestratora głosu możesz rozpocząć rejestrację, zatrzymać na chwilę lub zakończyć zapis (rys. 9.2.). Aplikacja pozwala także na przycięcie zapisanego dźwięku (rys. 9.3.). Plik z zapisem ma format m4a. Nie wszystkie odtwarzacze prawidłowo go odtwarzają. Informacje o formacie zapisu dźwięku m4a znajdziesz w internecie.

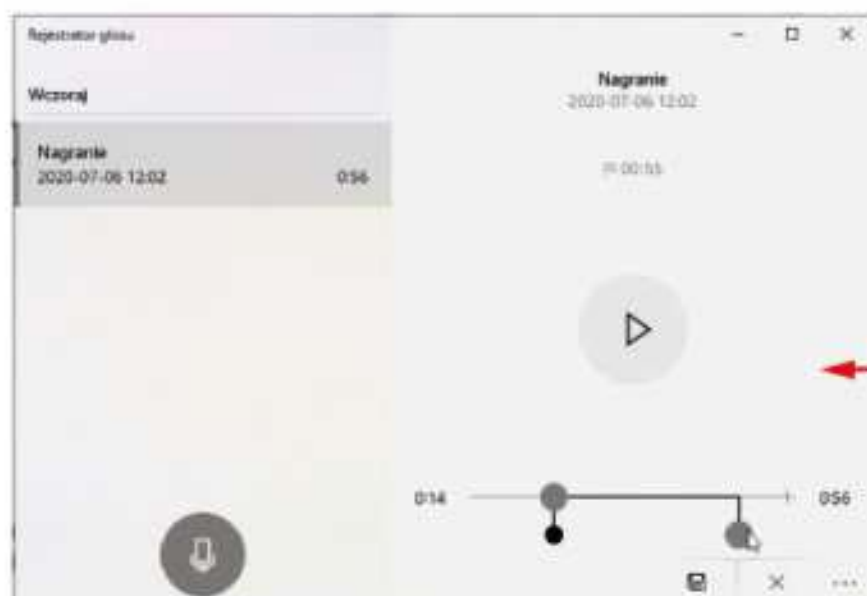
Jeśli aplikacji nie przypisano dostępu do mikrofonu, użyj opcji **Ustawienia mikrofonu**.



Rozpoczynanie i kończenie nagrania.

W każdej chwili możesz zrobić przerwę w nagrywaniu lub dodać znacznik w celu szybkiego odnajdowania danego fragmentu.

Rys. 9.2. Używanie **Rejestratora głosu**



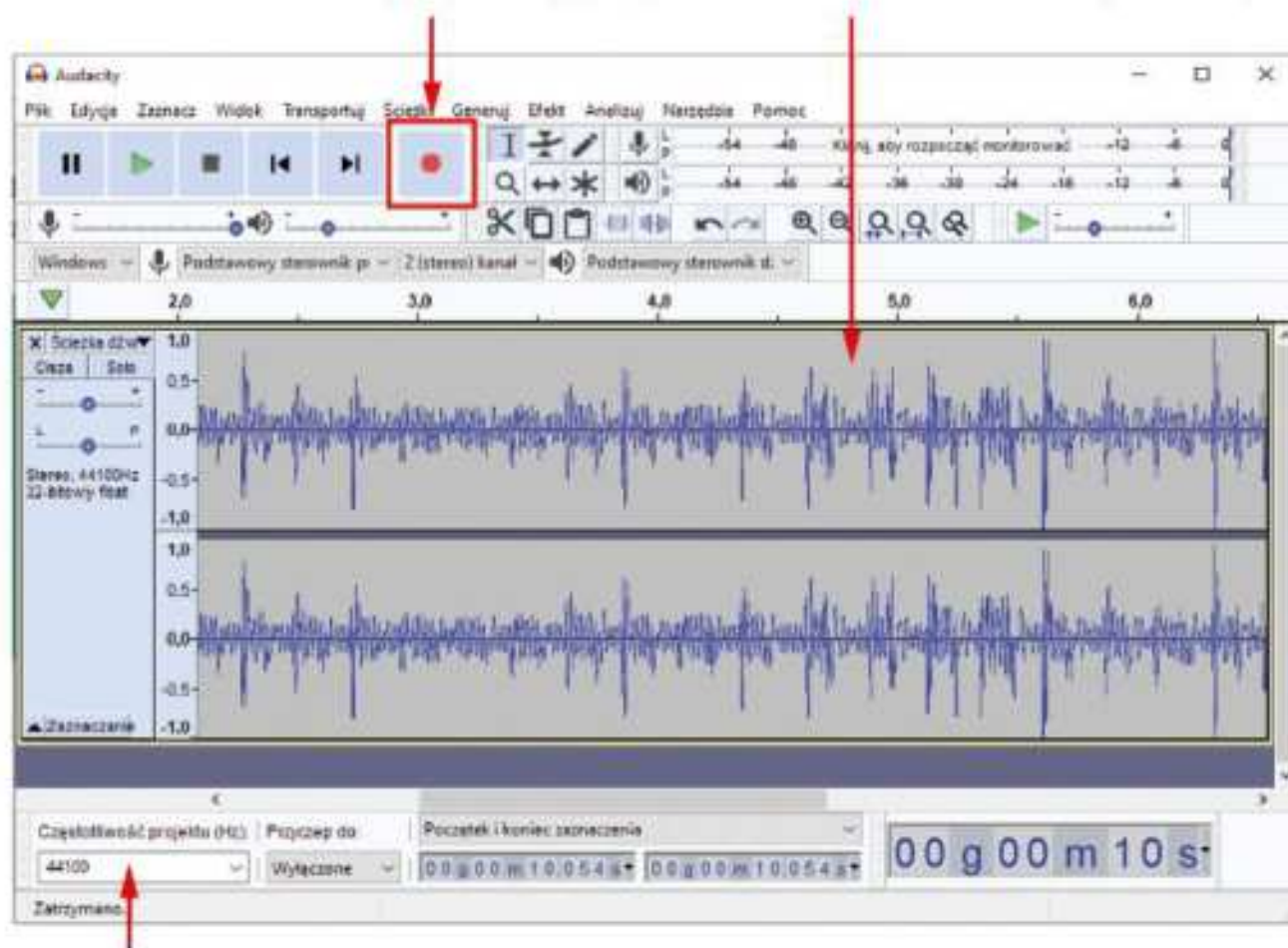
Po zarejestrowaniu nagrania Rejestrator głosu umożliwi jego odsłuchanie i przycięcie nagrania.

Rys. 9.3. Odsłuchanie i przycinanie nagrania dźwiękowego w **Rejestratorze głosu**

O wiele większe możliwości daje program Audacity. Rejestracja dźwięku z różnych źródeł jest jedną spośród wielu jego opcji. Mimo iż narzędzie to jest darmowe, stanowi pełnowartościowy edytor dźwięku z praktycznie nieograniczoną liczbą ścieżek i sporymi możliwościami wpływania na brzmienie dźwięku. Warto przetestować je wszystkie. W opisie ograniczymy się do rejestrowania dźwięku i prostej edycji (rys. 9.4.).

Audacity

Po uruchomieniu programu i ustaleniu parametrów nagrywania wciśnij przycisk **Nagrywaj**. Powstanie nowa ścieżka dla zapisu dźwięku.



Częstotliwość próbkowania ma istotne znaczenie dla jakości nagrania. 44,1 kHz jest wartością znamioną dla jakości płyty CD audio.

Rys. 9.4. Rejestracja dźwięku za pomocą **Audacity**



Częstotliwość próbkowania to liczba pomiarów (zamiast wartości napięcia sygnału analogowego, np. z mikrofonu, na liczbę) w ciągu sekundy. Ma ona istotne znaczenie dla jakości dźwięku.

Jakość dźwięku zależy także od **liczby bitów**, na których zostanie zapisany wynik pojedynczego pomiaru – 16 jest zazwyczaj wystarczające do zapisu mowy.

Jeśli w programie Audacity dodasz następne ścieżki, np. z muzyką wczytaną z pliku mp3, i ustalisz odpowiedni jej poziom, uzyskasz efekt zmiksowania dźwięku. W tle nagranych komentarza będzie słychać muzykę.

Gotowy projekt możesz zapisać w pliku dźwiękowym, np. w popularnym formacie mp3 (rys. 9.5.).



Rys. 9.5. Eksport projektu dźwiękowego programu Audacity do pliku w formacie mp3

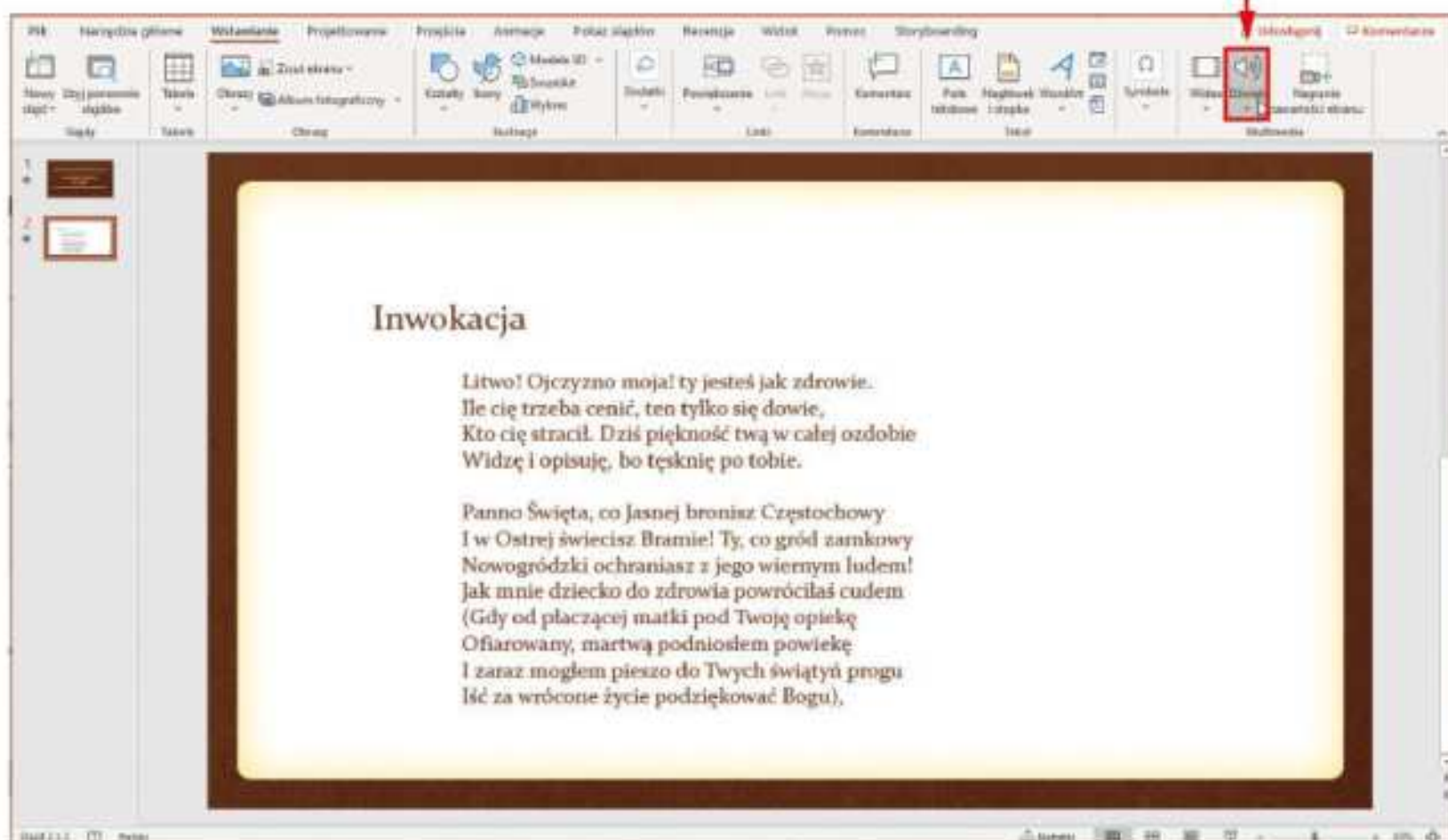
Edytor dźwięku Audacity może także posłużyć do tworzenia muzyki z sampli (próbek dźwięku), własnych dzwonek do telefonów i innych ciekawych projektów.

9.2. Prezentacja multimedialna, czyli dźwięk i film w PowerPoint

Załóżmy, że twoja interpretacja tekstu literackiego jest już zarejestrowana i zapisana w pliku dźwiękowym. Czas na umieszczenie jej w prezentacji (rys. 9.6.).



Z wstążki **Ustawienia** wybierz **Dźwięk**.



Odnajdź na dysku i wybierz przygotowany wcześniej plik dźwiękowy.

Możesz także wykorzystać opcję nagrywania narracji bezpośrednio w PowerPoint – **Nagraj dźwięk**.



Na slajdzie pojawi się symbol głośnika uruchamiający odtwarzanie, w trakcie którego jest wyświetlany także pasek czasu oraz przyciski pauzy i regulacji głośności. Ikonę głośnika możesz umieścić w dowolnym miejscu slajdu.



Rys. 9.6. Umieszczanie dźwięku na slajdzie PowerPoint

Wiesz już, jak umieścić dźwięk na slajdzie. Czas na podobne działanie z filmem (rys. 9.7.).

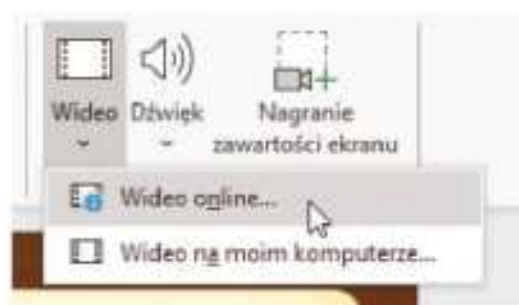
PowerPoint umożliwia umieszczenie filmu z pliku z twojego komputera, internetu lub bezpośredniego nagrania za pomocą wbudowanego rejestratora.

umieszczanie
multimediów
na slajdzie

źródła filmów
w PowerPoint



Jaka będzie różnica? Plik wgrany bezpośrednio do prezentacji zwiększy rozmiar pliku, ale zapewni odtwarzanie bez dostępu do internetu. Jeśli skorzystasz z opcji **Wideo online...**, plik prezentacji nie zwiększy objętości, ale do odtworzenia filmu konieczna będzie możliwość pobrania strumienia wideo bezpośrednio ze źródła, np. serwisu YouTube.



Rys. 9.7. Opcja wstążki **Wstawianie** służąca do umieszczania filmu na slajdzie

Oczywiście w trakcie edycji okienko odtwarzania filmu możesz dowolnie skalować i przemieszczać na slajdzie. Po wstawieniu filmu możliwe jest także formatowanie wideo (rys. 9.8.).



Rys. 9.8. Formatowanie filmu wstawionego do prezentacji PowerPoint

W poprzednim rozdziale opisano, jak zapisywać zawartość ekranu komputera w pliku filmowym. W PowerPoint umieszczono opcję o podobnym działaniu (rys. 9.9.).



Rys. 9.9. Nagranie zawartości ekranu za pomocą narzędzia programu PowerPoint

Prezentacja przygotowana z użyciem elementów multimedialnych może być bardzo atrakcyjna i znacznie wzmocnić przekaz treści. Należy jednak pamiętać, że przygotowana dla ilustrowania prelekcji powinna wspierać prelegenta, a nie dominować.

ZADANIA DO WYKONANIA

1. Opracuj projekt i wykonaj prezentację na temat wykorzystania Audacity do nagrania głosu narratora oraz podłożenia pod niego muzyki na odpowiednim poziomie.
2. Wykonaj prezentację zawierającą zrzuty ekranowe i komentarze audio narratora na temat ustawień jakości i parametrów nagrań w programie Audacity.
3. Wykonaj prezentację zawierającą filmy znalezione w sieci, odtwarzane on-line, dotyczące nagrywania dźwięku za pomocą telefonu komórkowego, np. opisujące aplikację, mikrofony zewnętrzne itp.
4. Wykonaj prezentację składającą się z jednego slajdu zawierającego zapis twoich działań na ekranie zarejestrowanych podczas formatowania wideo w PowerPoint. Jeśli dysponujesz mikrofonem, zarejestruj również komentarz.
- 5*. Użyj opcji **Nagraj dźwięk** z programu PowerPoint i w ten sposób zmodyfikuj jedną z wykonanych przez siebie wcześniej prezentacji na dowolny temat.

PODSUMOWANIE LEKCJI

- Do rejestracji dźwięku można użyć narzędzi dostarczonych wraz z systemem operacyjnym (np. **Rejestrator głosu**) lub narzędzi zewnętrznych (np. **Audacity**). s. 73
- Program **Rejestrator głosu** rejestruje dźwięk z mikrofonu i zapisuje go w formacie m4a. s. 71
- Program **Audacity** może służyć do rejestracji dźwięku z dowolnych źródeł i zaawansowanej, jak na darmowe narzędzie, edycji dźwięku. s. 73
- Na wstążce **Wstawianie** w PowerPoint umieszczono opcje **umieszczania na slajdach dźwięków i filmów**. s. 75
- Możliwe jest **umieszczenie filmu z pliku** (powiększa plik prezentacji), **z odnośnika**, np. YouTube, lub **akcji zarejestrowanej bezpośrednio z ekranu**. s. 75



10. Skuteczne wsparcie, czyli jak przygotować pokaz prezentacji

NA TEJ LEKCJI:

- dowiesz się, jak skonfigurować odtwarzanie multimediów na slajdach w zależności od przeznaczenia prezentacji;
- przekonasz się, jak opracować gotową prezentację do użycia w trakcie prelekcji;
- poznasz narzędzia PowerPoint ułatwiające przygotowanie prelekcji z użyciem prezentacji.

Twoja prezentacja jest już gotowa i możesz jej użyć. Aby pokaz lub prelekcja były skuteczne i atrakcyjne, przygotuj je starannie. Pomoże ci w tym szereg opcji edytora PowerPoint. Niektóre mają bezpośredni wpływ na prezentowanie treści slajdów, np. sposób odtwarzania multimediów, a inne pomogą przygotować konspekt wystąpienia. Wysiłek włożony w przygotowanie prezentacji nie powinien być zmarnowany przez zaniechanie przygotowań do jej użycia.

10.1. Odpowiedni sposób, czyli uruchamianie multimediów w prezentacji

Najnowsze wersje edytorów prezentacji mają bardzo rozbudowane opcje odtwarzania i kształtowania multimediów. Dają użytkownikowi już nie tylko możliwość ustawienia obiektu w dowolnym miejscu i uruchamiania automatycznie lub na żądanie. Edytory pozwalają na kształtowanie okna odtwarzania, skalowanie w trzech płaszczyznach, dodawanie efektów i napisów, korektę obrazu, przycinanie itp. Oczywiście nie zastąpią one narzędzi, za pomocą których należy przygotować film, ale mogą uatrakcyjnić sam pokaz.

Uwaga!

Wstążki z opcjami dotyczącymi filmów pokazują się po wybraniu w trakcie edycji danego obiektu filmowego.

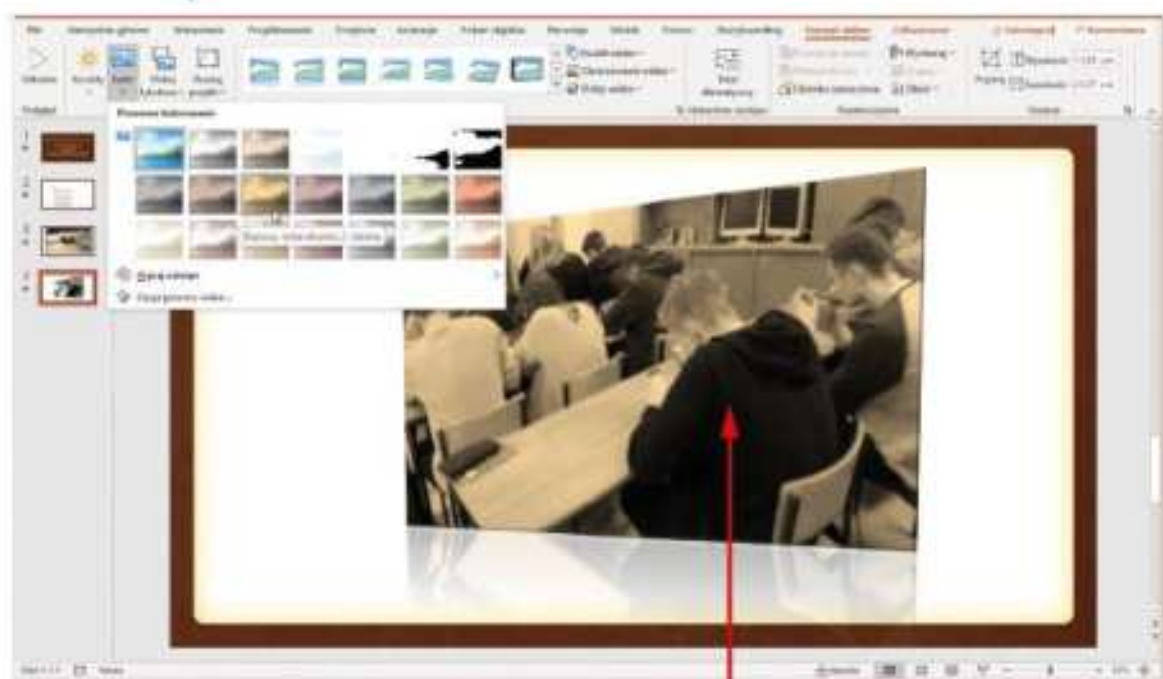
wstążka
z opcją filmów

Wstążki **Format wideo** i **Odtwarzanie** zawierają narzędzia pozwalające na zmianę bardzo wielu parametrów wyświetlania filmu na slajdzie (rys. 10.1.).



Korekty – pozwala na zmianę jasności filmu.

Klatką tytułową może być dowolne zdjęcie lub rysunek.



Ekran wyświetlający film na slajdzie może mieć różne kształty.

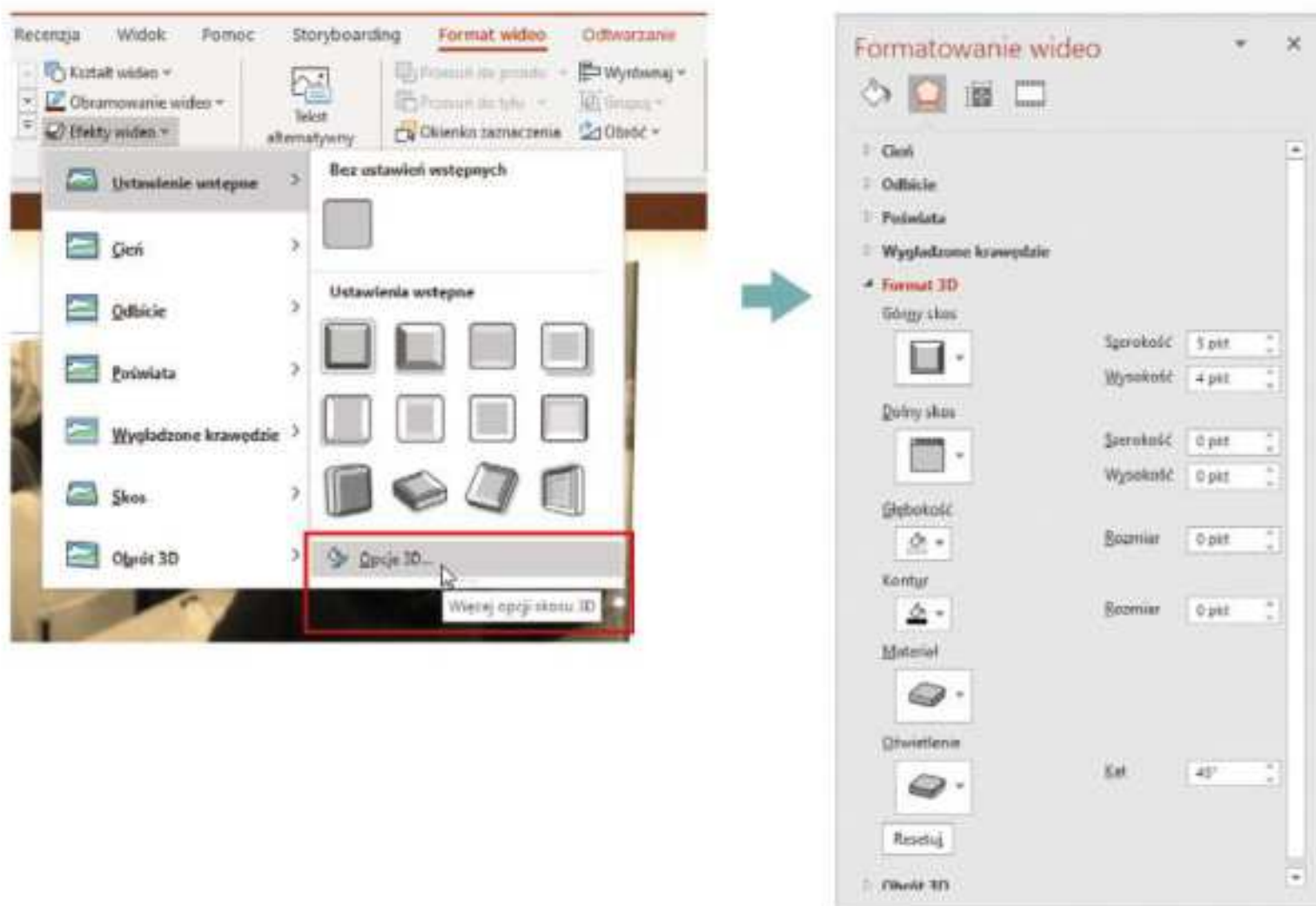


Opcja **Kolor** zmienia zabarwienie całego filmu.

Rys. 10.1. Niektóre z opcji formatujących wideo na slajdach w PowerPoint

Większości opcji formatowania wideo możesz używać, korzystając z ustawień domyślnych. Jeśli chcesz je zmienić (np. efekty 3D, cienie itp.), skorzystaj z możliwości ich edytowania (rys. 10.2.).

formatowanie
wideo



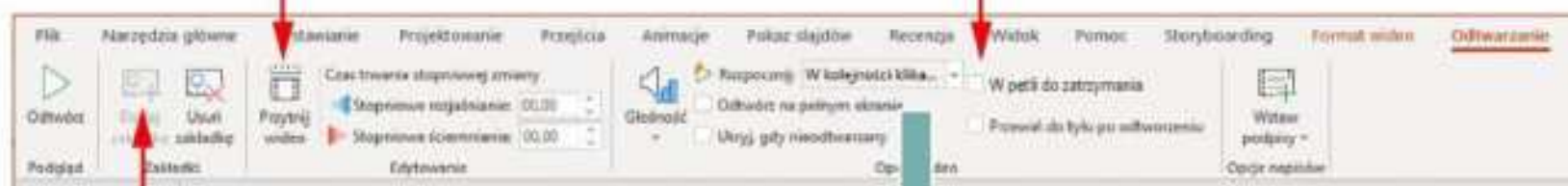
Rys. 10.2. Zmiana parametrów efektów 3D wyświetlania filmów w edytorze PowerPoint

wstążka Odtwarzanie

Gdy już dopracujesz szczegóły wyświetlania filmu na twoim slajdzie, możesz ustalić sposób jego odtwarzania, np. moment startu, głośność itp. (rys. 10.3.). W tym celu skorzystaj z opcji wstążki **Odtwarzanie**. Niektóre opcje mogą ułatwić prowadzenie prelekcji. Należy do nich między innymi stawianie znaczników na osi czasu filmu. Można nimi zaznaczać moment rozpoczęcia komentarza. Jeśli film lub animacja ma być ciągłą ilustracją prelekcji, warto ustawić opcję **W pętli do zatrzymania**.

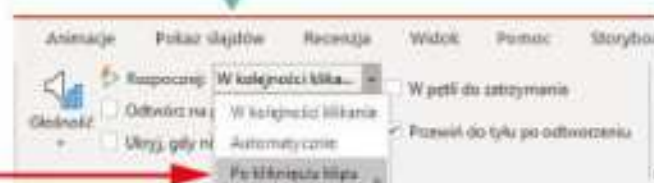
Przycięcie pozwala na wybór części filmu do odtwarzania i usunięcie niechcianych fragmentów.

Twój film może być wyświetlany w pętli lub za każdym razem, gdy uruchomisz odtwarzanie.



Możesz dodać zakładkę w filmie. Pomoże to w prelekcji.

Ważna opcja pozwalająca na dopasowanie uruchamiania filmu do konspektu prelekcji.



Rys. 10.3. Niektóre parametry odtwarzania filmu podczas prezentacji

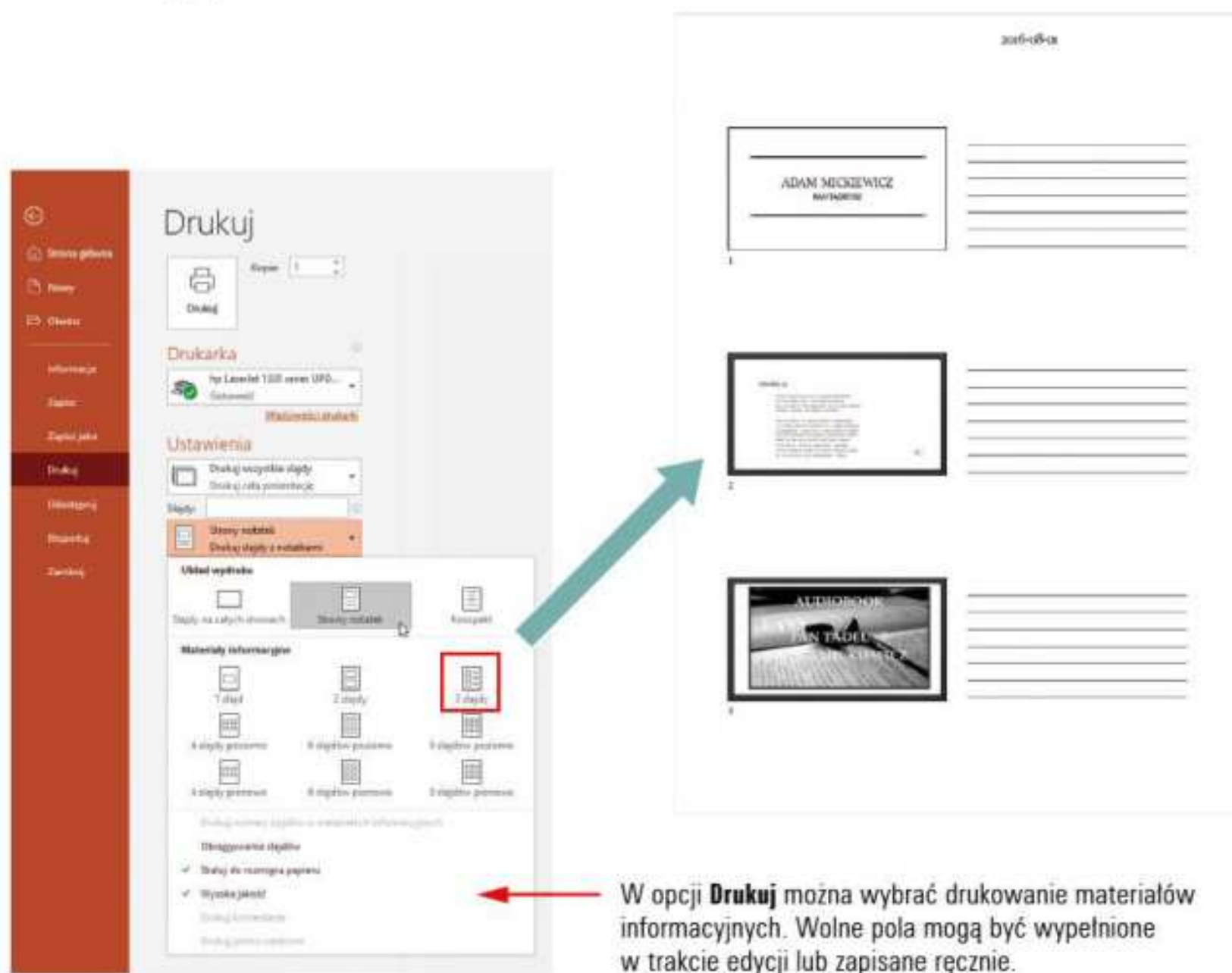


10.2. Konspekt, czyli przygotowujemy wystąpienie i materiały dla słuchaczy

Jeśli prezentacja ma zostać wykorzystana na lekcji, konferencji, prezentacji projektu itp., warto dobrze zaplanować wystąpienie i opracować materiały dla słuchaczy.

Wygodnym sposobem przygotowania się do prelekcji jest wydrukowanie konspektu składającego się z miniatur slajdów i treści do przekazania przez mówcę (rys. 10.4.). Jedną z opcji, którą można wykorzystać, jest drukowanie materiałów informacyjnych.

drukowanie
materiałów
informacyjnych



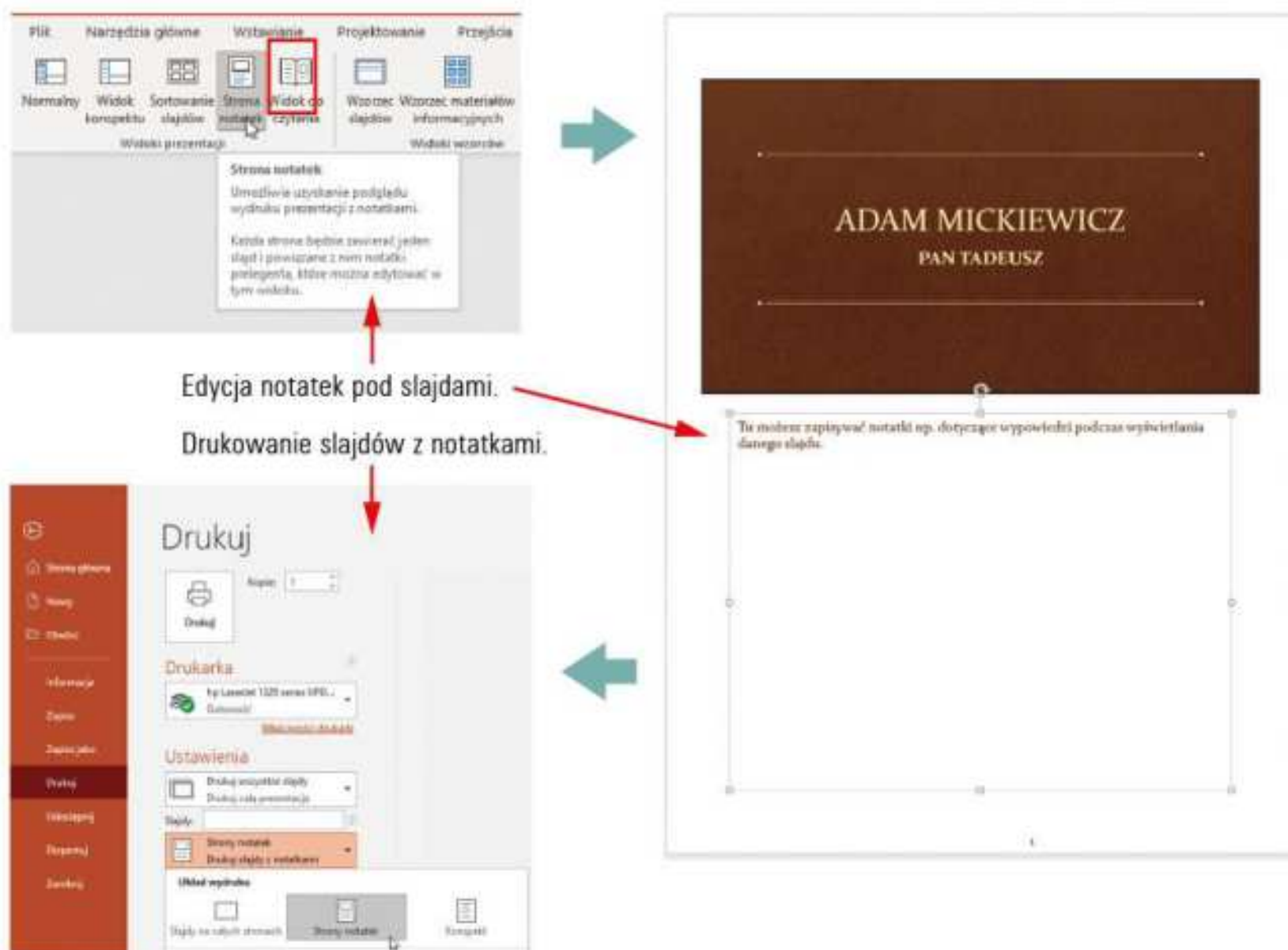
Rys. 10.4. Drukowanie materiałów informacyjnych w PowerPoint

W trakcie tworzenia prezentacji można opatrywać slajdy notatkami, np. na temat wypowiedzi podczas ich wyświetlania. Z takich notatek można korzystać podczas przygotowywania prelekcji. Dobre pomysły warto zapisywać, a edytor PowerPoint daje nam takie możliwości. Odpowiednią opcję znajdziesz na wstążce **Widok**. Zapisane notatki można także wydrukować w materiałach informacyjnych (rys. 10.5.).



Uwaga!

Notatki nie są wyświetlane podczas prezentacji slajdów.



Edycja notatek pod slajdami.

Drukowanie slajdów z notatkami.

Rys. 10.5. Edycja i wydruk notatek pod slajdami

10.3. Ostatnie szlify, czyli przygotowanie do prelekcji

Przygotowania do prelekcji są niezmiernie ważne, jeśli chcemy osiągnąć sukces. Każdy prelegent przygotowuje się indywidualnie. Jednak proces ten można ogólnie opisać w kilku punktach.

przygotowanie
prezentacji
do prelekcji

Przykład przygotowania prezentacji do użycia podczas prelekcji.

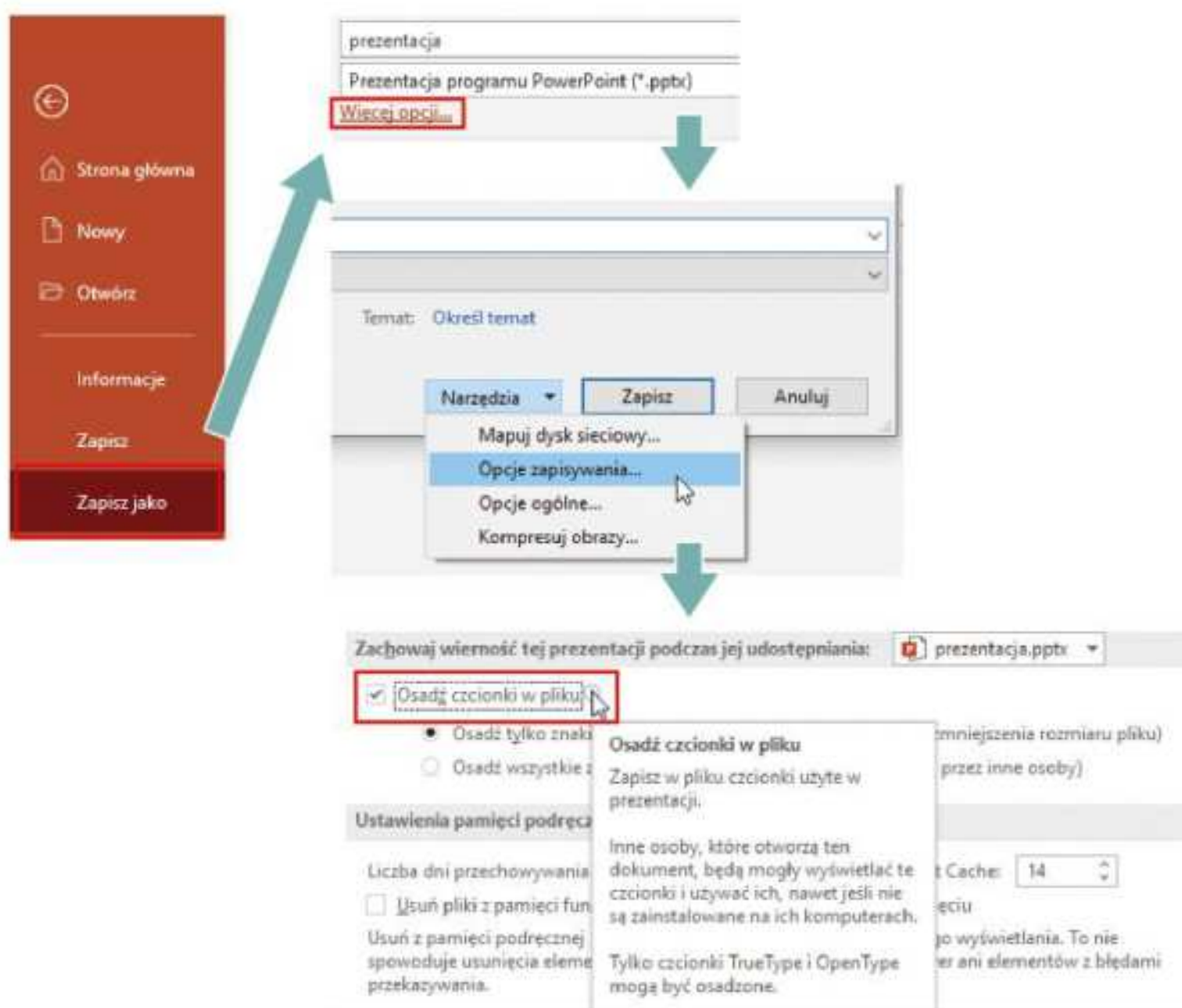
1. Opracowanie scenariusza prezentacji na podstawie analizy tematu i grona, do którego będzie skierowana (czy będą to uczniowie, studenci czy seniorzy itp.).
2. Przygotowanie materiałów do wykorzystania w prezentacji, np. zdjęć, ilustracji lub multimediów.
3. Wykonanie prezentacji w wybranym edytorze.
4. Ułożenie planu prelekcji i treści, które będzie przekazywał prelegent.



5. Przygotowanie **Materiałów informacyjnych** wspomagających przygotowania do prelekcji.
6. Przeprowadzenie testów działania prezentacji i prób prelekcji.
7. Sprawdzenie warunków, w jakich będzie się odbywać prelekcja, np. czy jest dostęp do internetu, jakie oprogramowanie może zostać użyte itp.
8. Ewentualne wydrukowanie materiałów informacyjnych dla odbiorców.

Punkt 7 może okazać się jednym z najważniejszych. Używanie innego komputera lub innej wersji oprogramowania czasami powoduje, że slajdy wyświetlane na ekranie znacznie się różnią od pierwotnej wersji. Może bowiem np. brakować czcionki użytej w prezentacji. Jak tego uniknąć? Wystarczy odpowiednio dobrać parametry zapisu pliku i osadzić czcionki w pliku (rys. 10.6.). Użyte czcionki zostaną zapisane w pliku z prezentacją.

osadzanie czcionek



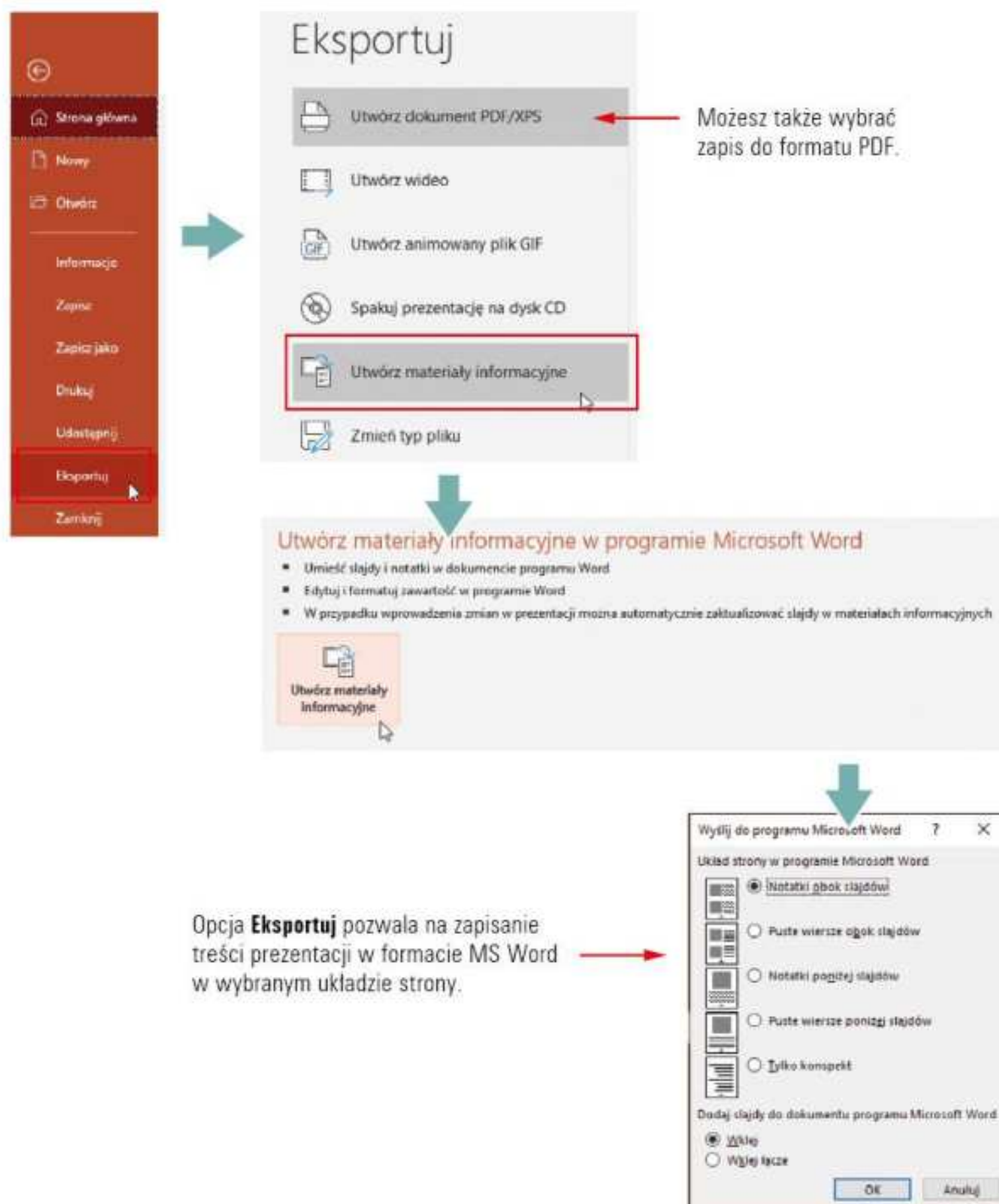
Rys. 10.6. Osadzenie czcionek w pliku prezentacji PowerPoint

Co zrobić, jeśli oprócz prelekcji musimy przygotować także materiały dla uczestników spotkania? Istnieje wiele możliwości. Możesz wykorzystać notatki do slajdów i tak je formułować, by były skrótem twoich wypowiedzi. Jeśli

eksport do innego formatu



przygotowujesz osobny dokument tekstowy, będzie potrzebny eksport do innych formatów, np. do pliku DOCX (rys. 10.7.).



Rys. 10.7. Eksportowanie prezentacji do postaci pliku tekstowego

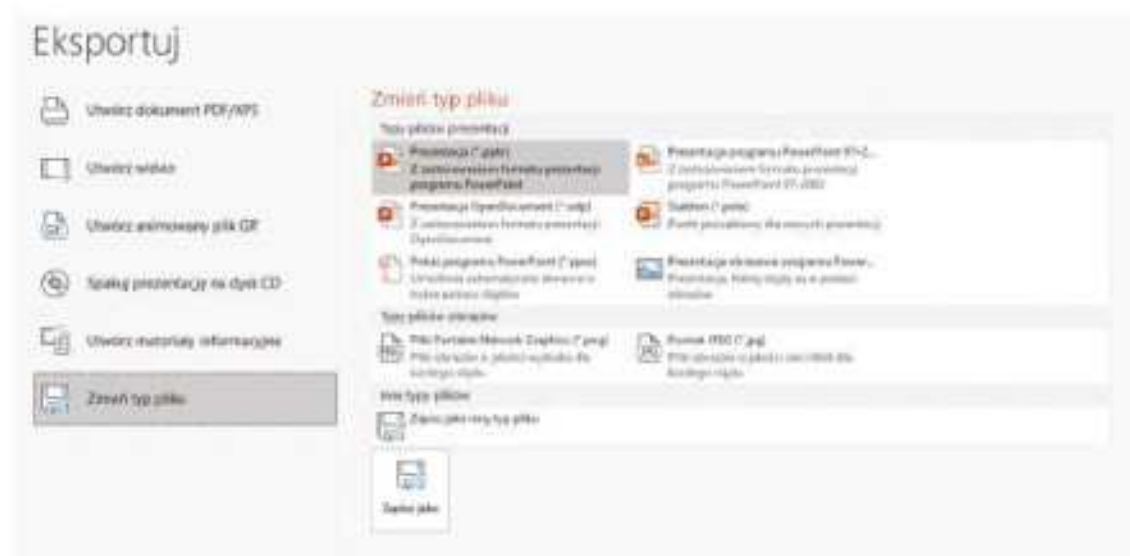
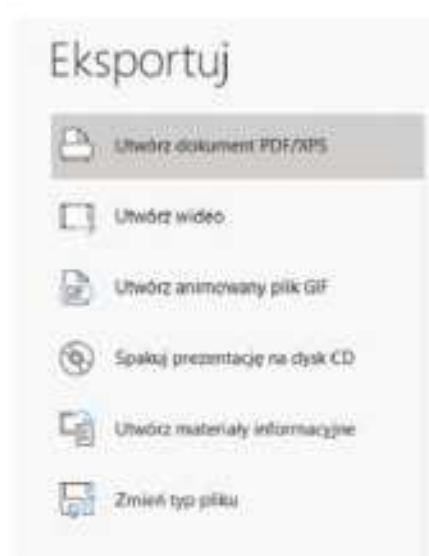


Uwaga!

Opcja **Eksportuj** może być też stosowana do przygotowania Materiałów informacyjnych opisanych w poprzednim podrozdziale.

Uzyskany w wyniku eksportu plik może być edytowany w edytorze tekstu oddzielnie lub wykorzystany jako część większego opracowania. Slajdy w takim dokumencie są wklejane jako obiekty. Nadal działa odtwarzanie multimedialnych i odnośniki.

Opcja **Eksportuj** umożliwia także zapis do innych formatów. Możliwe jest utworzenie ze slajdów filmu lub animowanego formatu pliku GIF. Można także zmienić typ pliku, w którym slajdy zostaną zapisane (rys. 10.8.).



Rys. 10.8. Opcje eksportu pliku prezentacji PowerPoint

Okazuje się, że przygotowanie dobrej prezentacji to nie tylko slajdy i multimedia. Profesjonalne podejście do jej tworzenia i planu wykorzystania stwarza duże szanse powodzenia prelekcji.



ZADANIA DO WYKONANIA

1. Przygotuj krótką prezentację na aktualnie omawiany temat na wybranym przedmiocie. Na jednym ze slajdów zamieść odpowiedni film i sformatuj wideo według własnego pomysłu. Uzasadnij użycie danego formatowania.
2. Dla dowolnej prezentacji, np. z zadania 1, przygotuj materiały dla słuchaczy. Dostosuj je do ilości, w jakich będzie użyta. Zamieść je w pliku PDF. Uzasadnij zastosowaną formę tych materiałów.
3. Na podstawie dowolnej prezentacji zawierającej multimedia przygotuj plik tekstowy będący jej streszczeniem. Taki plik będzie można wykorzystać do zareklamowania twojego wystąpienia. Wykorzystaj opcję **Eksportuj**.
- 4*. Zbadaj, jak program LibreOffice Impress odtworzy twoje prezentacje zawierające multimedia. Sprawdź, czy akceptuje on formatowanie wideo. Opisz swoje obserwacje w prezentacji PowerPoint.
- 5*. Dla dowolnej prezentacji (zapisanej bez osadzania czcionek) wykonaj doświadczenie wykazujące wpływ braku w systemie komputera czcionki użytej w czasie jej edycji. Wykonaj szereg zrzutów ekranowych. Ułóż prezentację przedstawiającą wyniki i przyczyny nieprawidłowości obserwowanych w czasie doświadczenia.

PODSUMOWANIE LEKCJI

s. 79,
80

- PowerPoint wzbogacono o wiele narzędzi służących do **formatowania wideo** i ustawiania opcji jego **odtworzenia**. Można je znaleźć we **wstążkach** pojawiających się po zaznaczeniu wideo na slajdzie.

s. 81

- W przygotowaniu do prelekcji z użyciem prezentacji przydatne są opcje służące do drukowania **materiałów informacyjnych**.

s. 82

- W **przygotowaniu prelekcji** warto zapisać w punktach czynności, od których będzie zależała jakość przekazu.

s. 83

- Jeśli prezentacja ma być odtwarzana na innym komputerze niż ten, na którym powstała, należy **osadzić czcionki** w jej pliku. Trzeba tak zrobić dlatego, że w systemie komputera, na którym ma być uruchamiana prezentacja, mogą nie być zainstalowane użyte czcionki.

s. 83

- Jeśli slajdy prezentacji mają być użyte do opracowania materiałów dla słuchaczy (odbiorców przekazywanych treści), można je **wyeksportować** do formatu tekstowego, np. DOCX lub PDF.



11. Atrakcyjnie i wygodnie, czyli jak upowszechniać prezentację

NA TEJ LEKCJI:

- przygotujesz automatyczny pokaz prezentacji;
- upowszechnisz prezentację zapisaną w różnych formatach;
- dowiesz się, jak prezentować slajdy na ekranach zewnętrznych, np. rzutnikach lub telewizorach.

Gdy prezentacja i materiały pomocnicze (informacyjne) są już przygotowane, pozostaje do rozwiązania jeszcze jeden problem – w jaki sposób przygotować jej pokaz. Zagadnienie wydaje się mało skomplikowane – wciskasz F5 lub udostępniasz plik i prezentacja upowszechniona. Gdy poznasz kilka ciekawych opcji edytora prezentacji, zrozumiesz, że możesz to zrobić na wiele sposobów w zależności od okoliczności, w jakich ma być ona użyta.

11.1. Różne formaty, czyli przygotowujemy prezentację do pokazu

PowerPoint umożliwia zapisanie prezentacji w wielu formatach. Ich wybór zależy od formy, w jakiej chcesz pokazywać swoje slajdy.

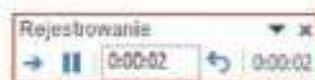
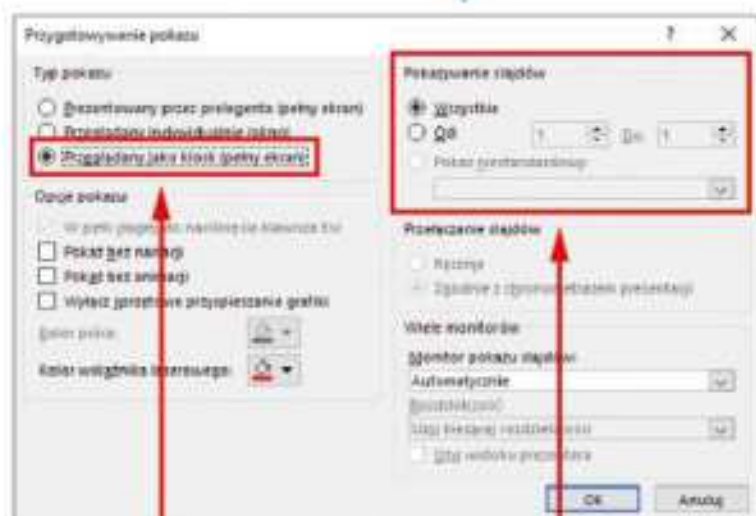
Przygotowując się do pokazu, zdecyduj, czy slajdy mają zmieniać się samoczynnie, czy ich zmianą kierować będzie prelegent. Upewnij się również, jaki sprzęt będzie dostępny.

Przykład 1.

Założmy, że slajdy mają się zmieniać samoczynnie. Ponieważ zawierają różne treści, czas ich wyświetlania może być różny. Konieczne będzie ustalenie chromatrazu (rys. 11.1.). Zapisanie prezentacji w formacie **PPSX** pozwoli na jej automatyczne uruchamianie z pominięciem edytora (rys. 11.2.).

Prezentacja zapisana w formacie **PPSX** uruchamia się automatycznie z pominięciem etapu wczytywania prezentacji do edytora.

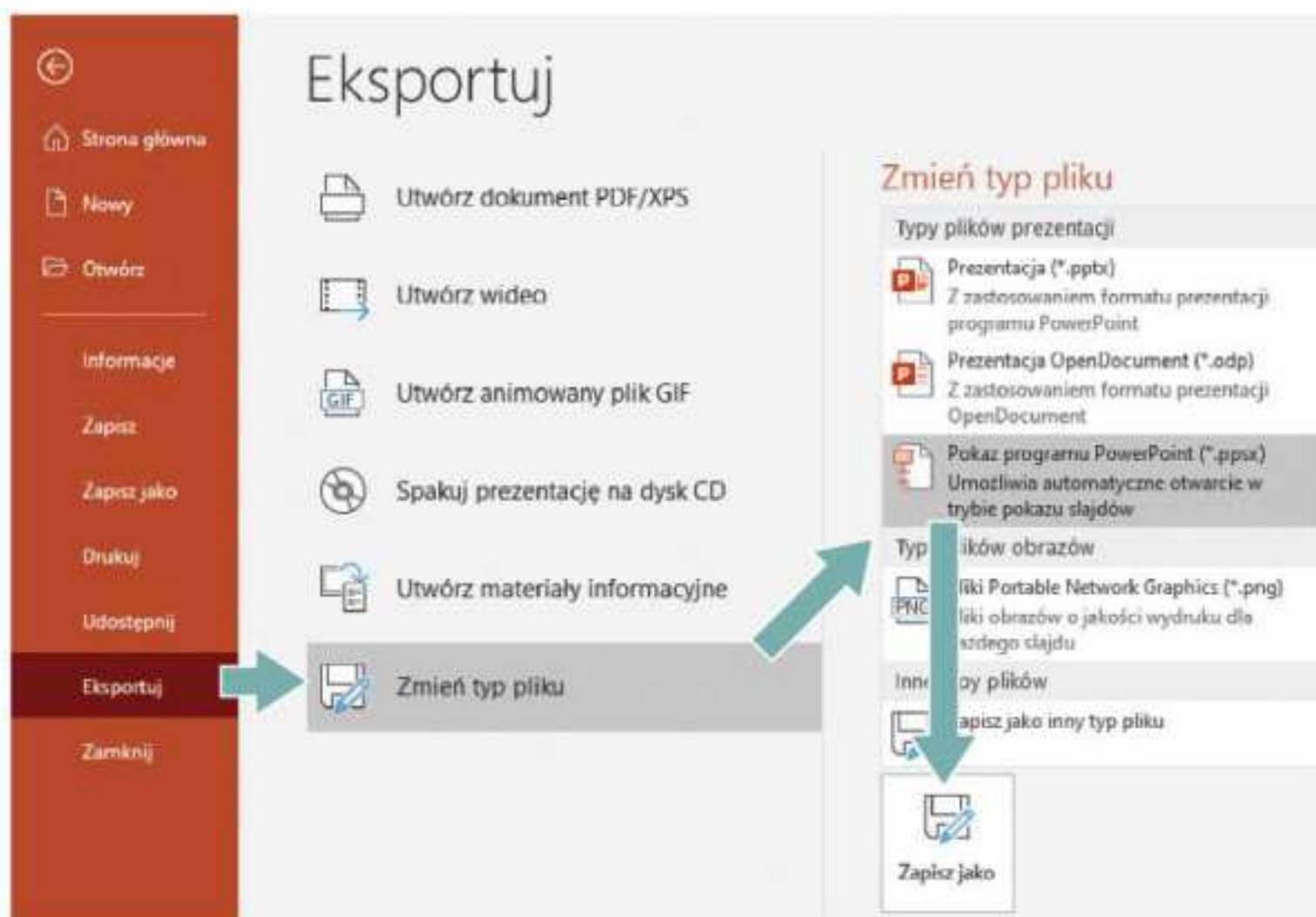
PPSX



Próba tempa pozwala na ustalenie czasu wyświetlania poszczególnych slajdów. Użycie tej opcji spowoduje uruchomienie pokazu z ręcznym przełączaniem slajdów, a czas ich wyświetlania będzie rejestrowany na żywo.

Możesz wybrać zakres wyświetlanych slajdów.

Rys. 11.1. Przygotowanie automatycznego pokazu slajdów z zastosowaniem chronometrażu



Rys. 11.2. Eksport pliku prezentacji do formatu .ppsx



Teraz wystarczy już tylko użyć klawisza F5 w trybie edycji lub uruchomić plik zapisany w formacie PPSX i można prezentować przygotowane slajdy według ustalonego **chronometrażu**.

Chronometraż jest metodą ustalania czasu potrzebnego do wykonania danej czynności. W przypadku prezentacji określa czas wyświetlania danego slajdu lub jego elementu podczas odtwarzania automatycznego, np. w trybie Kiosk.

chronometraż

Przykład 2.

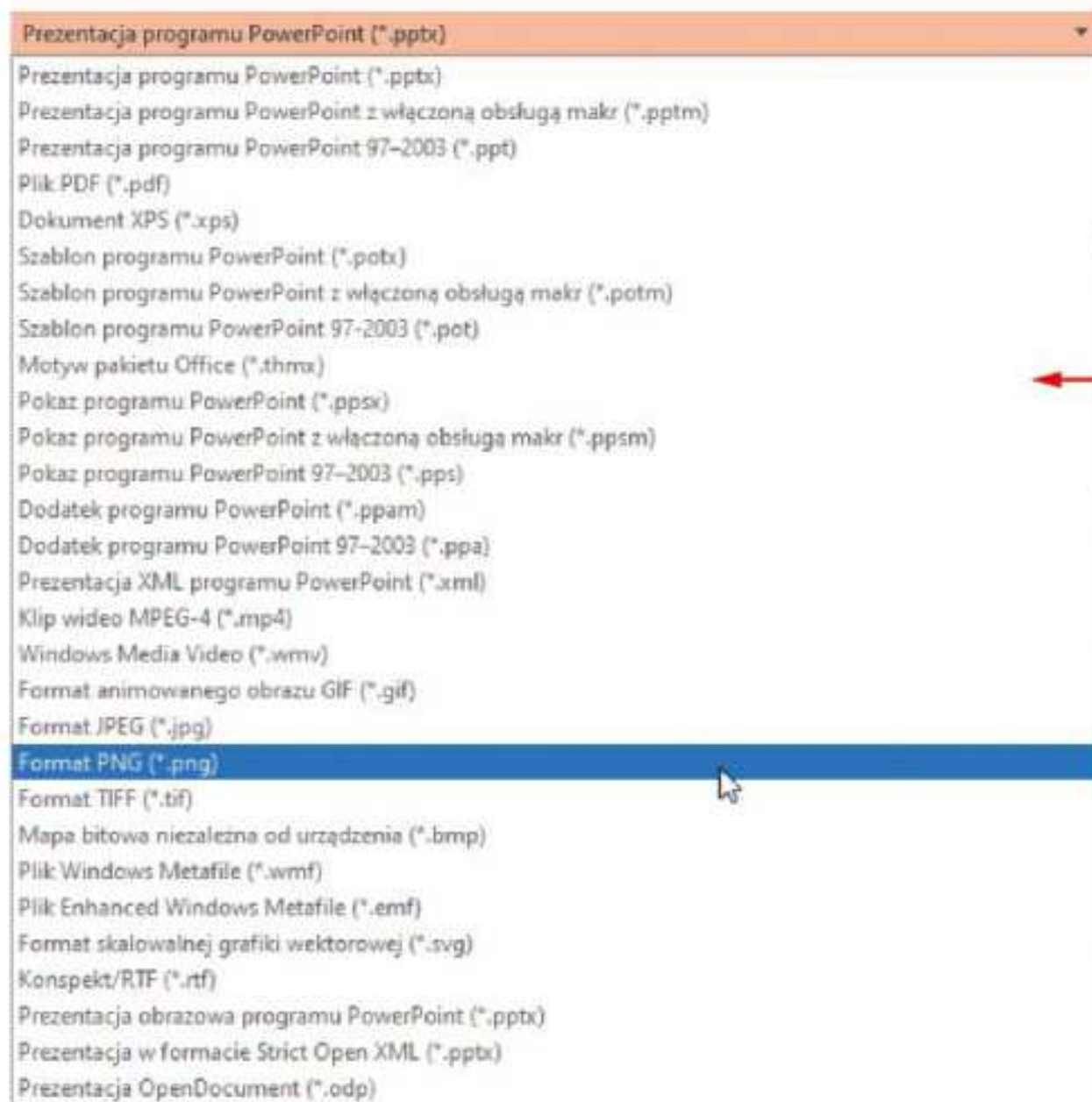
Automatyczny pokaz slajdów z chronometrażem wymaga komputera z zainstalowanym programem PowerPoint. Co zrobić, jeśli chcemy prezentować nasze slajdy w pętli na telefonie lub komputerze bez instalacji tego edytora? Istnieje kilka rozwiązań tego problemu. Można zapisać prezentację w formacie filmowym (rys. 11.3.). Taki plik z łatwością da się odtworzyć także w telefonie lub z zewnętrznej pamięci nowoczesnego telewizora czy rzutnika.



Rys. 11.3. Zapisywanie prezentacji PowerPoint w formacie filmowym

Przykład 3.

Slajdy mogą być także użyte w prezentacjach tworzonych w innych programach lub w filmach. Istnieje możliwość zapisu każdego z nich oddzielnie w formatach graficznych (rys. 11.4.), co sprawia, że ich użycie staje się łatwe.



Przeczytaj listę formatów zapisu prezentacji. Być może w przyszłości zastosujesz któryś z nich w swoim projekcie.

Rys. 11.4. Formaty plików, w których można zapisać prezentację PowerPoint

Wielość formatów plików, do których można wyeksportować prezentację PowerPoint, pozwala na użycie slajdów nie tylko do pokazów, lecz także do opracowań graficznych lub innych form prezentacji lub dokumentów.

11.2. Nie tylko kabel, czyli sposoby wyświetlania prezentacji

Założmy, że twoim zadaniem było przygotowanie prezentacji na targi edukacyjne lub drzwi otwarte w szkole. Być może z podobnym problemem spotkasz się w swojej przyszłej pracy lub na uczelni. Jeśli twoje treści, styl, grafika, zdjęcia i multimedia zostały już zaakceptowane, projekt może zostać zaprezentowany. Pozostaje do rozwiązania problem techniczny – jak dobrać odpowiedni sprzęt?

Rozpatrzmy przypadki, w których zostaną użyte urządzenia dostępne w szkole. Przykłady możliwych rozwiązań zapisano w tabeli (tab. 11.1.).



Lp.	Impress	prezi.com	
1.	Komputer połączony przewodem z rzutnikiem lub telewizorem	Plik prezentacji zapisany w pamięci masowej komputera, np. na dysku lub pamięci zewnętrznej	Komputer musi się znajdować w niewielkiej odległości od projektora lub telewizora. W przypadku zastosowania złącza VGA do odtwarzania dźwięku konieczne jest użycie oddzielnego przewodu łączącego wyjście karty dźwiękowej z rzutnikiem lub telewizorem. Użycie HDMI nie wymaga dodatkowych połączeń.
2.	Telewizor lub rzutnik z opcją odtwarzania filmów z pliku	Prezentacja zapisana w formacie filmowym, np. MP4 na nośniku zewnętrznym	Urządzenie musi mieć opcję odtwarzania filmów z nośnika zewnętrznego z opcją zapętlenia i możliwością zatrzymania odtwarzania. Plik z filmem należy zapisać na nośniku zewnętrznym ze złączem USB. Najwygodniejszym rozwiązaniem jest użycie pendrive'a. Rozwiązanie nie wymaga dodatkowych przewodów, zasilania ani miejsca.
3.	Telewizor lub rzutnik z opcją bezprzewodowego wyświetlania obrazu, np. w standardzie Miracast	Prezentacja zapisana w pamięci urządzenia z opcją udostępniania ekranu	Konieczne jest urządzenie nadawcze, np. smartfon lub komputer z systemem Windows 10 obsługującym Miracast. W tym rozwiązaniu można odtwarzać pliki zarówno z aplikacji PowerPoint, jak i w wersji wideo.

Tab. 11.1. Przykładowe rozwiązania wyświetlania prezentacji

Wybór sprzętu i rodzaju połączenia zależy od kilku czynników. Jeśli prezentacja ma być odtwarzana w pętli, np. w tle stanowiska targowego, na holu biura firmy lub w czasie drzwi otwartych w szkole, nie musisz dbać o wygodny dostęp prelegenta do sterowania pokazem. W takim przypadku wygodne będzie użycie każdego z wymienionych w tabeli przykładów rozwiązań. Jego wybór ograniczony będzie jedynie możliwościami posiadanego sprzętu i dostępnego miejsca na stanowisku.

Dla prelegenta atrakcyjne może być użycie rozwiązania nr 3. Da mu to możliwość wyświetlania slajdów z tabletu lub telefonu, czyli swobodę poruszania się w dużym promieniu od ekranu. W przypadku rozwiązania nr 1 konieczny będzie dostęp do bezprzewodowej myszki lub **bezprzewodowego prezentera slajdów** (rys. 11.5.). Jeśli prelegent ma **mieć pełną kontrolę nad pokazem**, powinien zainstalować aplikację PowerPoint zgodną z danym systemem telefonu i używać pliku w formacie PPTX.



Rys. 11.5. Bezprzewodowy prezenter slajdów

Bezprzewodowy prezenter slajdów to urządzenie pozwalające na zdalne sterowanie prezentacją – przełączanie slajdów, zamrażanie ekranu itp. Często ma także opcję emulacji myszy (jeśli nie pamiętasz, co to jest emulator, wyszukaj to hasło na stronie Wikipedii).

Rozwiązanie nr 2 jest doskonałym sposobem na wyświetlanie prezentacji na dużych ekranach, np. w holu szkoły, bez konieczności wykonywania dodatkowych połączeń i użycia komputera lub smartfona.

Przed ostateczną decyzją warto wypróbować różne warianty.

bezprzewodowy
prezenter slajdów



Użycie udostępniania ekranu z tabletu, telefonu lub komputera wymaga odpowiedniego skonfigurowania elementów zestawu. Po pierwsze, w większości rozwiązań muszą być one podłączone do tej samej sieci komputerowej. Tak jest w przypadku korzystania z popularnego standardu **Miracast**, który może być używany w smartfonach i tabletach, a także w komputerach z systemem Windows 10. W urządzeniu nadawczym konieczne jest użycie aplikacji lub odpowiedniej opcji systemu operacyjnego (rys. 11.6. oraz 11.7).

Miracast

Miracast to standard umożliwiający prezentowanie zawartości ekranu urządzenia mobilnego, np. smartfona, laptopa itp. na ekranach urządzeń stacjonarnych, np. rzutnikach, telewizorach lub ekranach komputerów.



Rys. 11.6. Przykład uruchamiania projekcji z telefonu komórkowego



Kliknij nazwę danego ekranu zewnętrznego.



Rys. 11.7. Przykład łączenia laptopa z ekranem bezprzewodowym

Uwaga!

Następne wersje systemów operacyjnych telefonów lub PC mogą różnić się od opisanych przykładów. Znasz zasadę wyświetlania obrazu na ekranie bezprzewodowym, więc z pewnością odnajdziesz odpowiednie opcje lub aplikacje.

Innym sposobem udostępniania ekranu ze smartfona lub tabletu jest użycie aplikacji zgodnej z technologiami MiracastDisplay, np. Screen Mirror.

Oczywiście możliwe jest także wyświetlanie obrazu z tabletu lub smartfona na ekranie komputera z Windows 10. Tu także wykorzystywany jest standard Miracast.



ZADANIA DO WYKONANIA

1. Wyszukaj w internecie lub sklepie zgodnym z systemem twojego telefonu darmowe aplikacje do wyświetlania ekranu smartfona lub tabletu bezprzewodowo na ekranie zewnętrznym. Zapoznaj się z opiniami na ich temat. Na tej podstawie wytypuj ten, który mógłbyś/mogłabyś zainstalować w swoim urządzeniu. Wymień najważniejsze jego zalety.
2. Wyszukaj w internecie oferty sprzedaży bezprzewodowych prezenterów slajdów. Przeczytaj uważnie ich opis. Zwróć uwagę na cechy funkcjonalne. Zapoznaj się z opiniami użytkowników i opisem na stronach producentów. Wybierz model, który według ciebie jest najlepszy i znalazłby zastosowanie w szkole lub pracy zawodowego trenera lub nauczyciela.
3. Wczytaj do edycji jedną z własnych prezentacji PowerPoint. Przeprowadź próbny pokaz, w którym ustalisz czas wyświetlania poszczególnych slajdów. Przygotuj pokaz slajdów z ustalonym chronometrażem i zapisz go w formacie PPSX. Sprawdź działanie prezentacji. Jeśli zajdzie konieczność, np. widz nie zdąży przeczytać treści slajdu, dokonaj korekt czasu wyświetlania.
4. Przeanalizuj treści tabeli 11.1. Podaj przykłady dla każdego z trzech przypadków. Opisz je w postaci prezentacji.
- 5*. Zainstaluj w telefonie aplikację PowerPoint. Sprawdź jej działanie na przykładzie prezentacji pobranej z komputera. Opisz w prezentacji PowerPoint zalety i wady tej aplikacji z uwzględnieniem edycji slajdów.
- 6*. Sprawdź, czy twój komputer wyposażony w kartę wi-fi ma możliwość wysyłania obrazu do ekranu bezprzewodowego. Wykorzystaj tę opcję i połącz go z telewizorem lub rzutnikiem zgodnym ze standardem Miracast. Opisz proces połączenia w krótkiej prezentacji PowerPoint.

PODSUMOWANIE LEKCJI

s. 87

- Prezentację PowerPoint możesz zapisać w formacie **PPSX**, co pozwoli na szybkie jej uruchamianie bez konieczności wczytywania edytora.

s. 89

- Opcje **Przygotuj pokaz** i **Próba tempa** pozwolą na przygotowanie pokazu działającego automatycznie zgodnie z **chronometrażem**, bez udziału prelegenta.

s. 91

- **Bezprzewodowy prezenter slajdów** pomaga w zdalnym sterowaniu pokazem slajdów.

s. 92

- Slajdy można prezentować na ekranach zewnętrznych połączonych za pośrednictwem przewodów lub bezprzewodowo dzięki sieci wi-fi. Bezprzewodowe rozwiązanie używa jednego ze standardów, np. standardu **Miracast**.



12. Utrzymujemy kontakt z klientami, czyli korespondencja seryjna

NA TEJ LEKCJI:

- poznasz przykłady zastosowania korespondencji seryjnej;
- dowiesz się, jak wykorzystać listy danych do korespondencji seryjnej;
- zredagujesz dokumenty zawierające elementy korespondencji seryjnej.

W praktyce użytkownicy pakietów biurowych spotykają się z problemem redagowania i drukowania dużej liczby dokumentów różniących się od siebie pewnymi szczegółami, np. imionami i nazwiskami adresatów. Przykładem mogą być urzędowe zawiadomienia skierowane do mieszkańców danej dzielnicy, zaproszenia, adresowanie kopert, drukowanie etykiet itp. Przy znacznej liczbie takich dokumentów opłaca się zastosować mechanizm korespondencji seryjnej, który będzie korzystał z gotowych danych zapisanych w bazie danych, arkuszu lub innym pliku edytora.

12.1. Skąd wziąć treści, czyli źródła danych dla dokumentów korespondencji seryjnej

Korzystanie z mechanizmu korespondencji seryjnej ma sens jedynie w przypadku utworzenia lub posiadania odpowiedniej bazy z danymi, które można wykorzystać do automatycznego wypełniania pewnych obszarów dokumentu. Takie źródło musi zawierać tabelę z nagłówkami. Może to być dokument tekstowy, arkusz Excel, tabela bazy danych itp.

Założmy, że dane do korespondencji będziemy przechowywać w tabeli arkusza Excel. Jeśli nie dysponujesz takim plikiem, utwórz go (rys. 12.1.). Arkusz może zawierać wiele danych i nie wszystkie muszą być użyte w danym dokumencie korespondencji seryjnej. Dzieje się tak bardzo często, np. gdy użytkownik tworzy bazę danych wykorzystywaną także do innych celów. Tabele mogą znajdować się na różnych arkuszach skoroszytu. W trakcie scalania będziesz mieć możliwość wyboru.

arkusz jako
źródło danych dla
korespondencji
seryjnej



Podczas scalania dokumentu z danymi pierwszy wiersz będzie traktowany jak nagłówki tabeli (listy).

	A	B	C	D	E	F
1						
2		Imię	Nazwisko	Miejscowość	Ulica i nr domu	Płeć
3		Zofia	Kowalska	Bydgoszcz	Kubusia Puchatka 234/12	k
4		Jan	Nowak	Nakło n. Notecią	Piracka 212/1	m
5		Anna	Kwiatkowska	Szubin	Bydgoska 198/4	k
6		Wojciech	Kowal	Paterek	Szubińska 112	m

Rys. 12.1. Przykładowy arkusz z danymi do korespondencji seryjnej

Innym źródłem danych może być tabela (lista) umieszczona w dokumencie tekstowym (rys. 12.2.).

Imię	Nazwisko	Miejscowość	Ulica i nr domu	Płeć
Zofia	Kowalska	Bydgoszcz	Kubusia Puchatka 234/12	k
Jan	Nowak	Nakło n. Notecią	Piracka 212/1	m
Anna	Kwiatkowska	Szubin	Bydgoska 198/4	k
Wojciech	Kowal	Paterek	Szubińska 112	m

Treść dokumentu poza tabelą nie będzie importowana w korespondencji seryjnej.

Do utworzenia tabeli możesz użyć różnych edytorów tekstu.

Imię	Nazwisko	Miejscowość	Ulica i nr domu	Płeć
Zofia	Kowalska	Bydgoszcz	Kubusia Puchatka 234/12	k
Jan	Nowak	Nakło n. Notecią	Piracka 212/1	m
Anna	Kwiatkowska	Szubin	Bydgoska 198/4	k
Wojciech	Kowal	Paterek	Szubińska 112	m

Treść dokumentu poza tabelą nie będzie importowana w korespondencji seryjnej.

Rys. 12.2. Przykładowe tabele danych dla korespondencji seryjnej utworzone w różnych edytorach tekstu

różne formaty
źródeł danych

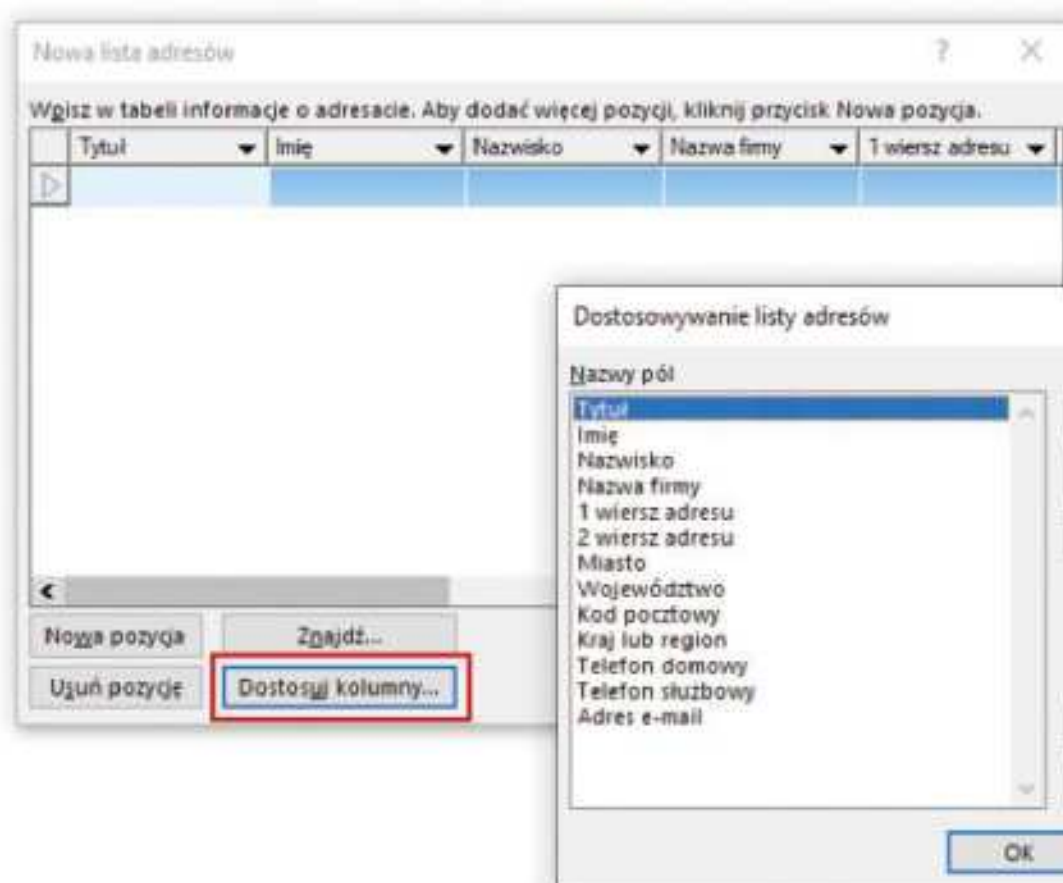
W korespondencji seryjnej możesz użyć list z tabel zapisanych w różnych formatach, w tym także otwartych, np. odt lub ods.

wpisywanie
nowej listy

Jeśli nie chcesz wykorzystać ani tworzyć osobnych dokumentów, możesz wpisywać listy w procesie edycji korespondencji seryjnej (rys. 12.3.).



Wybierz wstążkę
Korespondencja.



Edytor domyślnie
tworzy nagłówki tabeli.
Możesz je edytować,
dodawać lub usuwać.

Rys. 12.3. Przykład tworzenia nowej listy danych w trakcie edycji korespondencji seryjnej

12.2. Zawiadamiamy, czyli korespondencja seryjna w biurze

Założmy, że uczniowie z przygotowanej listy brali udział w olimpiadzie przedmiotowej. Organizator chce powiadomić ich o miejscu, jakie zajęli w eliminacjach, i o tym, czy awansowali do następnego etapu. Wyniki wpisał do przygotowanej wcześniej listy w Excelu. Wymagało to niewielkich modyfikacji (rys. 12.4.). Oczywiście w rzeczywistych warunkach uczestników byłoby znacznie więcej i użycie listy do drukowania zawiadomień dla olimpijczyków znacznie ułatwiłoby i przyspieszyłoby pracę.



	A	B	C	D	E	F	G	H
1								
2		Imię	Nazwisko	Miejscowość	Ulica i nr domu	Płeć	Awans	Wynik
3		Zofia	Kowalska	Bydgoszcz	Kubusia Puchatka 234/12	k	t	78
4		Jan	Nowak	Nakło n. Notecią	Piracka 212/1	m	n	49
5		Anna	Kwiatkowska	Szubin	Bydgoska 198/4	k	n	38
6		Wojciech	Kowal	Paterek	Szubińska 112	m	t	87

Rys. 12.4. Przykład listy z wynikami olimpiady

Treść dokumentu dla każdego z uczestników będzie identyczna poza najważniejszymi informacjami – imieniem i nazwiskiem, wynikiem i informacją o awansie do kolejnego etapu. Zredagujmy takie zawiadomienie (rys. 12.5.).

KOMITET OKRĘGOWEJ OLIMPIADY INFORMATYCZNO-ELEKTRONICZNEJ

Gdańsk, 20 grudnia 2022 roku

imię nazwisko
adres
miejscowość

ZAWIADOMIENIE

Komitet OOIE zawiadamia, że w eliminacjach okręgowych osiągnęła/osiągnął pani/pan wynik ... co kwalifikuje panią/pana do udziału w etapie wojewódzkim.

Termin eliminacji wojewódzkich zostanie opublikowany na stronie internetowej olimpiady.

Treści, które należy pobrać z listy w trakcie redagowania dokumentu z użyciem korespondencji seryjnej.

Rys. 12.5. Przykład treści zawiadomienia jako podstawy do użycia korespondencji seryjnej

Skoro podstawowa treść dokumentu jest już gotowa, możesz scalić ją z listą danych (rys. 12.4.) i użyć mechanizmu korespondencji seryjnej. Zauważ, że w przypadku informacji o awansie i formuły grzecznościowej nie będzie to przepisanie danych wprost z listy. Do dzieła.

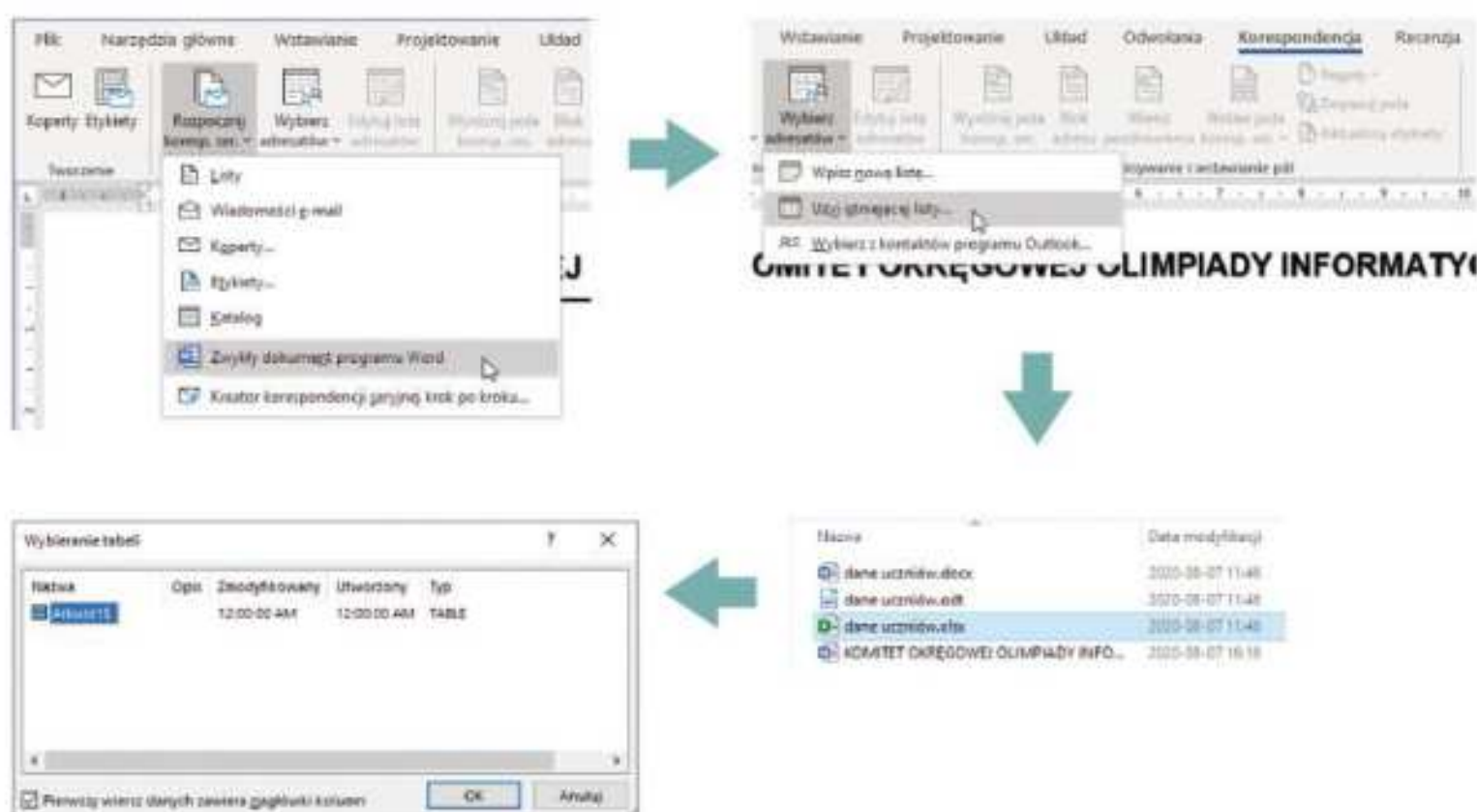


dołączanie listy
z danymi
korespondencji
seryjnej

Kolejność czynności w trakcie dołączania listy z danymi do korespondencji seryjnej:

1. Otwórz dokument z treścią zawiadomienia.
2. Wybierz wstążkę **Korespondencja** i narzędzie **Rozpocznij korespondencję seryjną**.
3. Z tej samej wstążki użyj narzędzia **Wybierz adresatów** i opcji **Użyj istniejącej listy...**
4. Wybierz **arkusz** z wybranego pliku. Zauważ, że odblokowane zostały pozostałe narzędzia wstążki **Korespondencja**.

Prześledź ten proces na rysunku (rys. 12.6.).



Rys. 12.6. Kolejność czynności w trakcie dołączania listy z danymi do korespondencji seryjnej

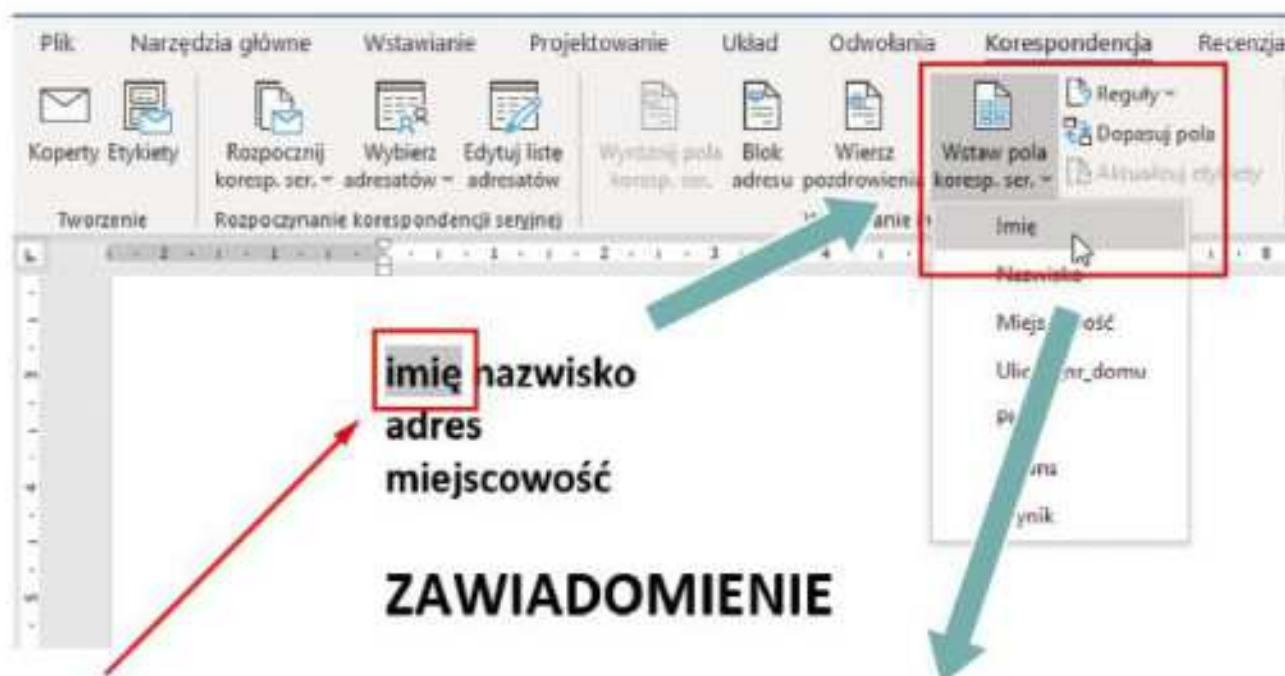
Dane z listy z arkusza dane uczniow.xls mogą teraz być użyte do redagowania dokumentu z korespondencją seryjną.

wstawianie pól
korespondencji
seryjnej

Edycję rozpoczniemy od miejsc zawierających imię, nazwisko i adres uczestnika (rys. 12.7.).

Uwaga!

W przykładzie narzędzie **Rozpocznij korespondencję seryjną** zostało użyte na istniejącym dokumencie z tekstem. Oczywiście można go użyć na dokumencie pustym i redagować tekst od podstaw jako korespondencję seryjną.



Zaznacz miejsce, w którym ma zostać wstawione pole korespondencji seryjnej. Wybierz pole. W spisie znajdziesz nagłówki tabeli listy danych.

«Imię» «Nazwisko»
«Ulica_i_nr_domu»
«Miejscowość»

Czynności powtórz dla kolejnych danych. Pola objęte korespondencją seryjną znajdują się pomiędzy znakami <<>>.

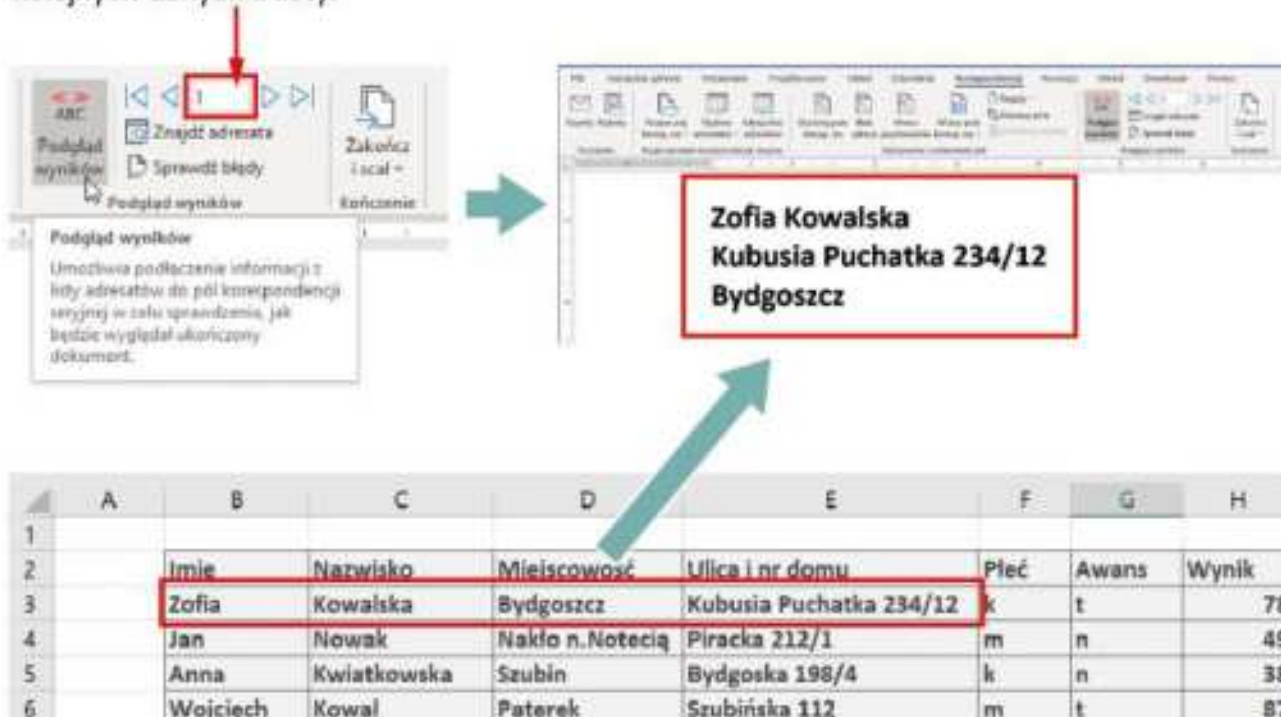
ZAWIADOMIENIE

Rys. 12.7. Wstawianie pola korespondencji seryjnej

Uwaga!

Opisany sposób można zastąpić także użyciem przycisku **Blok adresu**. Dane z listy zostały powiązane z polami korespondencji seryjnej w dokumencie. Czas sprawdzić, czy wszystko działa prawidłowo. W tym celu użyj narzędzia **Podgląd wyników** z wstążki **Korespondencja** (rys. 12.8.).

Używając klawiszy sterujących podglądem, możesz sprawdzić prawidłowość wstawiania kolejnych danych z listy.

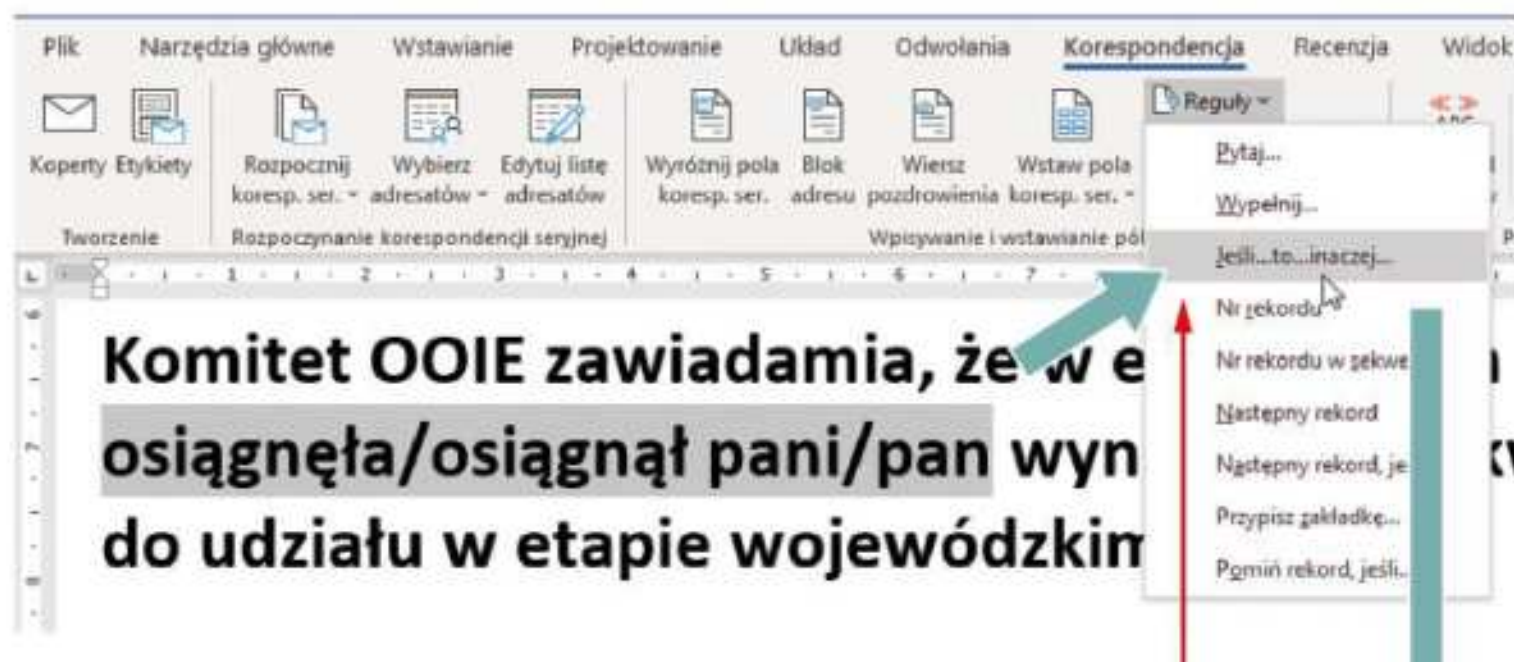


Rys. 12.8. Sprawdzanie poprawności wstawiania pól korespondencji seryjnej



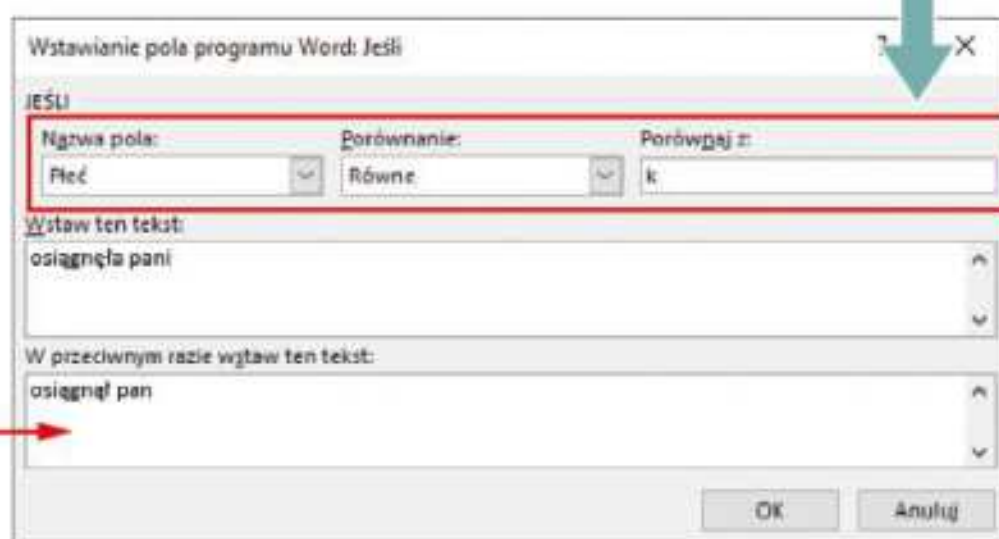
W treści dokumentu mamy pola uzależnione od płci i wyniku egzaminu. Dane są zapisane w liście w kolumnach **Płeć** i **Awans** zawierających jedynie znaczniki. Konieczne będzie użycie reguł w korespondencji seryjnej (rys. 12.9.). Pozwalają one zastąpić dane z listy innym tekstem, uzależniając go od wartości pola, np. literę **k** zastąpić słowem **pani**.

użycie reguł
w korespondencji
seryjnej



Zamiast narzędzia **Wstaw pola korespondencji seryjnej** użyj **Reguły Jeśli...to...inaczej...**

Jeśli pole **płeć** będzie miało wartość **k** (kobieta), wstaw tekst *osiągnęła pani*, jeśli **m** – *osiągnął pan*.



Rys. 12.9. Użycie **Reguły** w korespondencji seryjnej

Uwaga!

Czcionka tekstu wstawianego do korespondencji seryjnej za pomocą **Reguły** może się różnić od kroju czcionki stosowanej w dokumencie. Łatwo to zmienisz, używając standardowych opcji formatowania tekstu.

Po sprawdzeniu poprawności działania dokumentu korespondencji seryjnej można scalić dokument z listą (rys. 12.10.). Po tej operacji powstanie nowy dokument zawierający dane wstawione z listy. Jeśli zechcesz wprowadzić zmiany w korespondencji seryjnej, musisz skorzystać z pliku bazowego i ponownie użyć opcji **Zakończ i scal**.

zakończenie edycji
korespondencji
seryjnej

**Uwaga!**

Warto zachować plik bazowy korespondencji seryjnej. Dzięki niemu będzie można wprowadzać poprawki lub użyć go dla innych list danych. Jeśli listy te będą miały te same nagłówki tabel, nie będzie potrzeby modyfikacji dokumentu.



Możesz zapisać dokument wykorzystujący wszystkie rekordy (w naszym przykładzie będą to 4 zawiadomienia na oddzielnych stronach) lub tylko wybrane.

Rys. 12.10. Kończenie edycji i scalanie dokumentu z danymi w korespondencji seryjnej

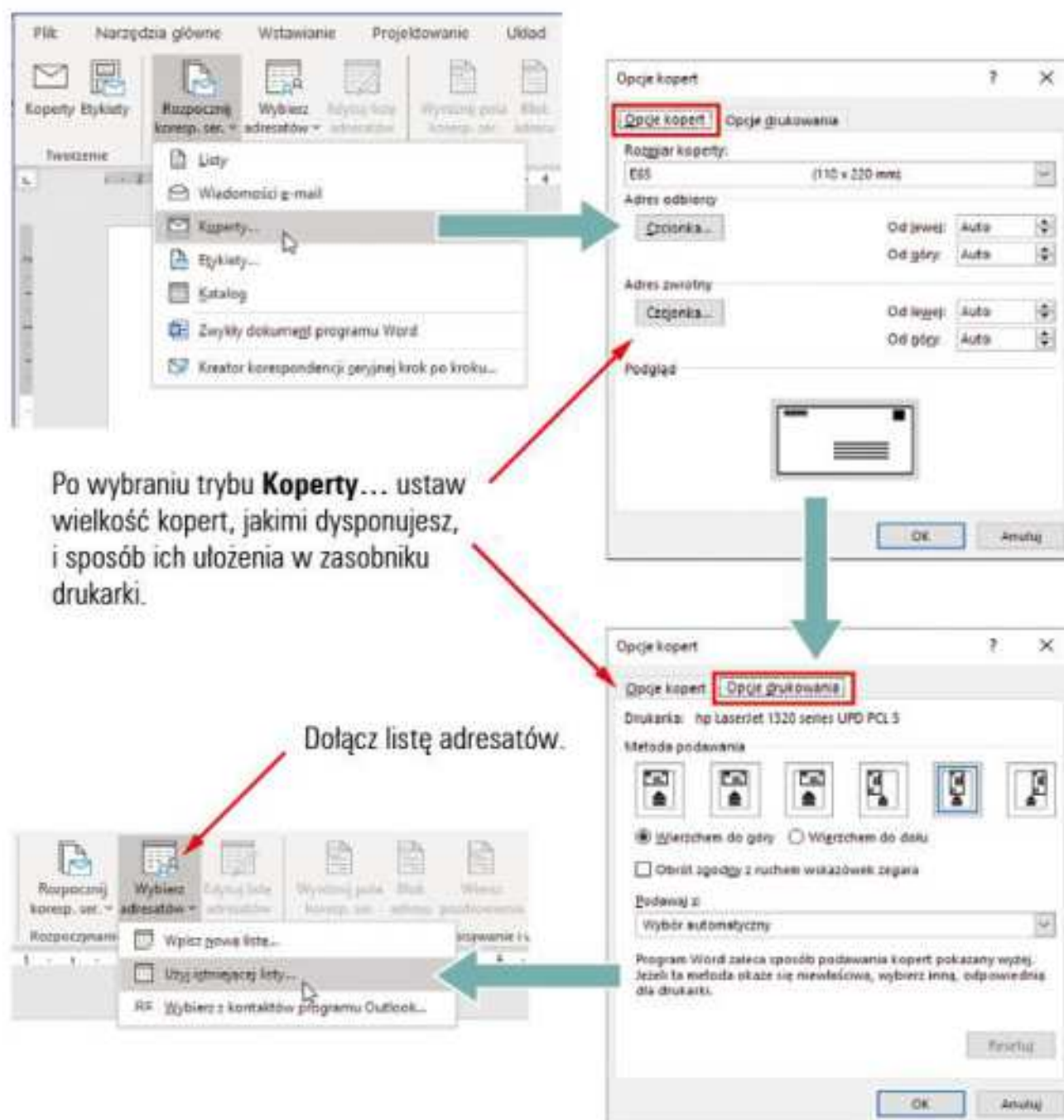
Dokument, który powstanie po zakończeniu i scaleniu naszego zawiadomienia, będzie zawierał 4 strony, z których każda będzie przeznaczona dla innego olimpijczyka. Taka sama zasada obowiązuje przy dokumentach wielostronicowych.

12.3. Nie tylko dokumenty, czyli do czego jeszcze można wykorzystać korespondencję seryjną

Korespondencja seryjna to nie tylko narzędzie do tworzenia dokumentów tekstowych. Można ją wykorzystać także do adresowania kopert lub druku etykiet na arkuszach samoprzylepnych.

adresowanie
kopert

Ponieważ mamy już gotowe zawiadomienia o wynikach olimpiady, zaadresujemy koperty do ich wysłania. Tym razem rozpoczniemy od utworzenia nowego, pustego dokumentu. Użyjemy tego samego źródła danych (rys. 12.11.).



Rys. 12.11. Inicjowanie adresowania kopert w korespondencji seryjnej

Kolejne czynności są identyczne jak w przypadku dokumentu tekstowego z poprzedniego podrozdziału. Ostatecznie dokument podstawowy dla adresowania kopert może mieć postać jak na rysunku 12.12.



Rys. 12.12. Przykładowy ostateczny wygląd dokumentu podstawowego dla adresowania kopert

**drukowanie
etykiet**

Jeśli wybierzesz opcję **Etykiety**, możesz przygotować dokument do wydrukowania na arkuszach z etykietami samoprzylepnymi. Tu także konieczny będzie wybór formatu. Wskaż format papieru z etykietami, jakim dysponujesz. Dzięki korespondencji seryjnej dane z listy będą wstawiane dokładnie w pola pokrywające się z etykietami na danym arkuszu.

ZADANIA DO WYKONANIA

1. Wykonaj własną listę, wykorzystując opcję **Wpisz nową listę...** z wstążki **Korespondencja**. Zmodyfikuj listę, by zawierała takie same nagłówki i dane jak na rysunku 12.4.
2. Na podstawie listy z zadania 1 utwórz zaproszenie na galę zakończenia olimpiady z wykorzystaniem korespondencji seryjnej. Możesz skorzystać z szablonów dostępnych w edytorze.
3. Znajdź w sklepie internetowym wymiary koperty z okienkiem. Utwórz dokument tekstowy, w którym pole adresowe znajdzie się dokładnie w miejscu okienka koperty. Użyj korespondencji seryjnej.
4. Utwórz prezentację na temat drukowania etykiet z zastosowaniem korespondencji seryjnej. Uwzględnij format etykiet i rodzaj drukarki, na której będą drukowane. Dane znajdziesz w internecie.
- 5*. Utwórz listę uczestników studniówki i wykonaj projekt zaproszenia z zastosowaniem korespondencji seryjnej. Możesz skorzystać z szablonu dostępnego w edytorze.

PODSUMOWANIE LEKCJI

s. 96

- **Źródłami listy danych** dla dokumentów korespondencji seryjnej może być tabela z dokumentu tekstowego, arkusz kalkulacyjny, tabela bazy danych, np. Access lub Outlook. MS Word akceptuje różne formaty plików z listami.

s. 99

- Pierwszy wiersz tabeli tworzy **nagłówek listy**.

s. 99

- Dołączanie listy z danymi korespondencji seryjnej zapewnia opcja **Wybierz adresatów**.

s. 99

- W opcji **Wstaw pola korespondencji seryjnej** pojawiają się do wyboru nagłówki tabeli listy danych.

s. 100

- Zanim zapiszesz plik wynikowy korespondencji seryjnej, użyj narzędzia **Podgląd wyników**.

s. 101

- **Zakończenie edycji i scalenie** spowoduje powstanie nowego pliku zawierającego dokument, w którym pola korespondencji seryjnej zostaną zastąpione danymi z listy.

s. 102

- **Adresowanie kopert** jest skuteczne pod warunkiem prawidłowego wybrania rozmiaru koperty zbliżonego do tej, na której będą drukowane adresy.

s. 104

- **Drukowanie etykiet** także wymaga dostosowania ustawień korespondencji seryjnej do rozmiarów arkuszy, na których będą drukowane.



PODSUMOWANIE PRZED SPRAWDZIANEM

1. Nowoczesna firma, czyli jak programy komputerowe ułatwiają pracę

- **Klient poczty** to program do zarządzania korespondencją elektroniczną. Pozwala także na zapamiętywanie niektórych danych adresowych. s. 13
- **Kalendarze komputerowe**, pracujące jako oddzielne aplikacje lub będące częścią pakietów, pozwalają na skuteczne zarządzanie czasem pracowników lub własnym, np. podczas przygotowań do matury. s. 14

2. Kalkulujemy, czyli jak wykorzystać arkusz kalkulacyjny w zarządzaniu finansami

- Arkusz należy tak **projektować**, aby zmiany można było w nim wprowadzić w prosty i szybki sposób. s. 15

3. Z sieci do tabeli, czyli jak interpretować dane za pomocą arkusza kalkulacyjnego

- Pobieranie (import) danych do tabeli w arkuszu kalkulacyjnym można wykonać na kilka sposobów. s. 26
- Przeniesienie danych do arkusza, dzięki możliwości użycia jego narzędzi, formuł itp., umożliwi **wykonanie na nich dowolnych operacji** dostępnych w arkuszu, co znacznie ułatwi ich wizualizację i analizę. s. 27
- **Sortowanie w tabeli arkusza** pozwala na szybsze i dokładniejsze przeanalizowanie danych. s. 26

4. Spośród wielu, czyli filtrowanie w arkuszu kalkulacyjnym

- Narzędzia do filtrowania są uniwersalne i mogą służyć do wyszukiwania danych zapisanych w obszernych tabelach według różnych parametrów. **Przykładem mogą być** liczby lub ich zakresy, tekst, kolor lub dowolnie wpisany filtr (liczba lub tekst). s. 29
- **Po filtrowaniu wiersze pozostają niezmienione**, a odfiltrowane dane pozostają jedynie niewidoczne – nie znikają z tabel. Po usunięciu filtru zostanie wyświetlona pełna, zapisana w arkuszu tabela. s. 30
- Menu kontekstowe udostępnia opcje **filtrowania zgodnie z zawartością listy danych** (kolumny tabeli). s. 30
- Filtrowanie nie musi dotyczyć liczb lub tekstu. Można filtrować **według kolorów**. s. 34
- **Fragmentatory** udostępniają przyciski do szybkiego filtrowania w tabelach. Wyświetlają także aktualny stan filtrowania. s. 35
- W filtrowaniu z użyciem kilku filtrów należy zwrócić baczną uwagę na wybór pomiędzy **koniunkcją a alternatywą**. s. 31



5. Z eksperymentu do arkusza, czyli analiza danych z doświadczenia

- s. 37 • Arkusz może służyć jako narzędzie do **symulacji różnych zjawisk**, np. losowych, których wyniki zostaną wykorzystane w dalszych badaniach.
- s. 41 • **Wyniki doświadczeń** można wpisać do tabeli arkusza i analizować.
- s. 42 • **Tworzenie wykresów** znacznie ułatwia analizę danych uzyskanych w czasie doświadczenia. Wzbogaca także **dokumentację** doświadczeń.

6. Edytor grafiki w pracy zawodowej, czyli tworzymy reklamę

- s. 43 • **Warstwy** są podstawowym mechanizmem tworzenia grafiki, a także korekty i montażu zdjęć. Na warstwie można umieścić każdy element graficzny.
- s. 45 • Zastosowanie elementów graficznych z **kanalem alfa** (przezroczystością) zapewni zachowanie kształtu wklejanego detalu. Kanał alfa jest znamienny dla 32-bitowej głębi kolorów, w której 8 bitów opisuje poziom przezroczystości każdego punktu.
- s. 46 • **Skalowanie i przemieszczanie** warstw to podstawowe czynności edycyjne grafiki, skuteczne po umieszczeniu ich w **oddzielnych warstwach**.
- s. 48 • Przygotowując elementy graficzne, należy pamiętać o prawach autorskich. W razie potrzeby można **zakupić zdjęcie lub rysunek w serwisie internetowym**.

7. Reklama jest ważna, czyli jak wykonać atrakcyjną prezentację

- s. 53 • **Prezentację** zawsze należy dostosowywać do grupy docelowej odbiorców, a także używać atrakcyjnej formy graficznej.

8. Prezentacja wideo, czyli jak przygotować prezentację filmową

- s. 65 • Przed przystąpieniem do realizacji nagrania prezentacji wideo należy przemyśleć i ułożyć **scenariusz**, a na jego podstawie **storyboard**.
- s. 65 • Podczas nagrywania i montażu powinno się stosować ogólnie przyjęte zasady tworzenia filmu, w tym dbałość o **nieprzekraczanie osi filmowej**.
- s. 69 • Zanim zainstalujesz zewnętrzny program, sprawdź czy przeznaczona do rejestracji aplikacja dostarczona z systemem spełnia twoje wymagania. Jeśli tak, instalacja nie będzie konieczna.
Do rejestracji **obrazu z telefonu komórkowego** możesz użyć aplikacji wbudowanej lub zewnętrznej.

9. Multimedia w prezentacji, czyli dźwięk i film na slajdach

- s. 73 • Do rejestracji dźwięku można użyć narzędzi dostarczonych wraz z systemem operacyjnym (np. **Rejestrator głosu**) lub narzędzi zewnętrznych (np. **Audacity**).
- s. 71 • Program **Rejestrator głosu** rejestruje dźwięk z mikrofonu i zapisuje go w formacie m4a.
- s. 73 • Program **Audacity** może służyć do rejestracji dźwięku z dowolnych źródeł i zaawansowanej, jak na darmowe narzędzie, edycji dźwięku.
- s. 75 • Na wstążce Wstawianie w PowerPoint umieszczono opcje **umieszczania na slajdach dźwięków i filmów**.
- s. 75 • Możliwe jest **umieszczenie filmu z pliku** (powiększa plik prezentacji), z **odnośnika**, np. YouTube, lub **akcji zarejestrowanej bezpośrednio z ekranu**.



10. Skuteczne wsparcie, czyli jak przygotować pokaz prezentacji

- PowerPoint wzbogacono o wiele narzędzi służących do **formatowania wideo** i ustawiania opcji jego **odtworzenia**. Można je znaleźć we **wstążkach** pojawiających się po zaznaczeniu wideo na slajdzie. s. 79, 80
- W przygotowaniu do prelekcji z użyciem prezentacji są przydatne opcje służące do drukowania **materiałów informacyjnych**. s. 81
- W **przygotowaniu prelekcji** warto zapisać w punktach czynności, od których będzie zależała jakość przekazu. s. 82
- Jeśli prezentacja ma być odtwarzana na innym komputerze niż ten, na którym powstała, należy **osadzić czcionki** w jej pliku. Trzeba tak zrobić dlatego, że w systemie komputera mogą nie być zainstalowane użyte czcionki. s. 83
- Jeśli slajdy prezentacji mają być użyte do opracowania materiałów dla słuchaczy (odbiorców przekazywanych treści), można je **wyeksportować** do formatu tekstowego, np. DOCX lub PDF. s. 83

11. Atrakcyjnie i wygodnie, czyli jak upowszechniać prezentację

- Prezentację PowerPoint możesz zapisać w formacie **PPSX**, co pozwoli na szybkie jej uruchamianie bez konieczności wczytywania edytora. s. 87
- Opcje **Przygotuj pokaz** i **Próba tempa** pozwolą na przygotowanie pokazu działającego automatycznie zgodnie z **chronometrażem**, bez udziału prelegenta. s. 89
- **Bezprzewodowy prezynter slajdów** pomaga w zdalnym sterowaniu pokazem slajdów. s. 91
- Slajdy można prezentować na ekranach zewnętrznych połączonych za pośrednictwem przewodów lub bezprzewodowo dzięki sieci wi-fi. Bezprzewodowe rozwiązanie używa jednego ze standardów, np. standardu **Miracast**. s. 92

12. Utrzymujemy kontakt z klientami, czyli korespondencja seryjna

- **Źródłami listy danych** dla dokumentów korespondencji seryjnej może być tabela z dokumentu tekstowego, arkusz kalkulacyjny, tabela bazy danych, np. Access lub Outlook. MS Word akceptuje różne formaty plików z listami. s. 96
- Pierwszy wiersz tabeli tworzy **nagłówek listy**. s. 99
- Dołączanie listy z danymi korespondencji seryjnej zapewnia opcja **Wybierz adresatów**. s. 99
- W opcji **Wstaw pola korespondencji seryjnej** pojawiają się do wyboru nagłówek tabeli listy danych. s. 99
- Zanim zapiszesz plik wynikowy korespondencji seryjnej, użyj narzędzia **Podgląd wyników**. s. 100
- **Zakończenie edycji i scalenie** spowoduje powstanie nowego pliku zawierającego dokument, w którym pola korespondencji seryjnej zostaną zastąpione danymi z listy. s. 101
- **Adresowanie kopert** jest skuteczne pod warunkiem prawidłowego wybrania rozmiaru koperty zbliżonego do tej, na której będą drukowane adresy. s. 102
- **Drukowanie etykiet** także wymaga dostosowania ustawień korespondencji seryjnej do rozmiarów arkuszy, na których będą drukowane. s. 104

A large green Android robot statue is the central focus of the image. It has a rounded head with two antennae, a single eye, and a wide, smiling mouth. The statue is made of a smooth, green material and stands on a sandy surface. In the background, there are trees and a clear blue sky. The right side of the image is covered by a solid magenta-colored overlay.

II. Linux i inne systemy operacyjne

- ▶ Różne sposoby instalacji, czyli przygotowujemy miejsce dla systemu Linux
- ▶ Wiele wersji, czyli wybieramy dystrybucję Linuxa
- ▶ Bez kosztów, czyli programy w Linux
- ▶ Tryb tekstowy jest ważny, czyli poznajemy konsolę Linux
- ▶ Okienka nie są potrzebne, czyli używamy konsoli Linux
- ▶ Komputer w kieszeni, czyli jak wykorzystać system Android w nauce i pracy
- ▶ Był pierwszym z okienkami, czyli macOS i jego właściwości



13. Różne sposoby instalacji, czyli przygotowujemy miejsce dla systemu Linux

NA TEJ LEKCJI:

- poznasz przeznaczenie systemu Linux;
- dowiesz się, czym jest wirtualna maszyna i jak jej użyć do pracy w różnych systemach operacyjnych;
- przygotujesz nośnik zewnętrzny przeznaczony do instalacji systemu operacyjnego.

Używasz komputerów na tyle długo, że zdajesz sobie sprawę, iż Windows nie jest jedynym systemem operacyjnym dla komputerów PC. Wiesz także, że alternatywą dla systemu Microsoftu może być Linux. By poznać jego właściwości i podstawy zarządzania nim, należy oczywiście zainstalować go na komputerze. Czy musisz umieścić go na dysku systemowym? Czy konieczne jest pozbycie się dotychczas używanego systemu? Jak zainstalować i skonfigurować wirtualny komputer? Odpowiedzi znajdziesz w tym rozdziale.

13.1. Czym jest Linux, czyli alternatywa nie tylko dla Windows

Według wikipedia.pl jako datę powstania jądra Linux przyjmuje się 25 sierpnia 1991 roku (rys. 13.1.).

Historia [edytuj | edytuj kod]

Jako datę powstania jądra przyjmuje się **25 sierpnia 1991** (22:57 UTC+2), czyli moment, w którym Linus Torvalds wysłał list na [grupę dyskusyjną comp.os.minix](#) rozpoczynający się słowami:

„Właśnie piszę (wolny) system operacyjny (dla zabawy, nie będzie wielki i profesjonalny jak [gnu](#)) dla klonów AT 386(486)...”^[3]

Pierwsze efekty jego pracy można było zobaczyć już 5 października 1991, kiedy to opublikował swój kod, zaczynając wiadomość tak:

„Czy tęsknicie za pięknymi czasami miniksa-1.1, kiedy mężczyźni byli mężczyznami i pisali swoje własne sterowniki? [...]”^[1]

Rys. 13.1. Fragment encyklopedii internetowej wikipedia.pl

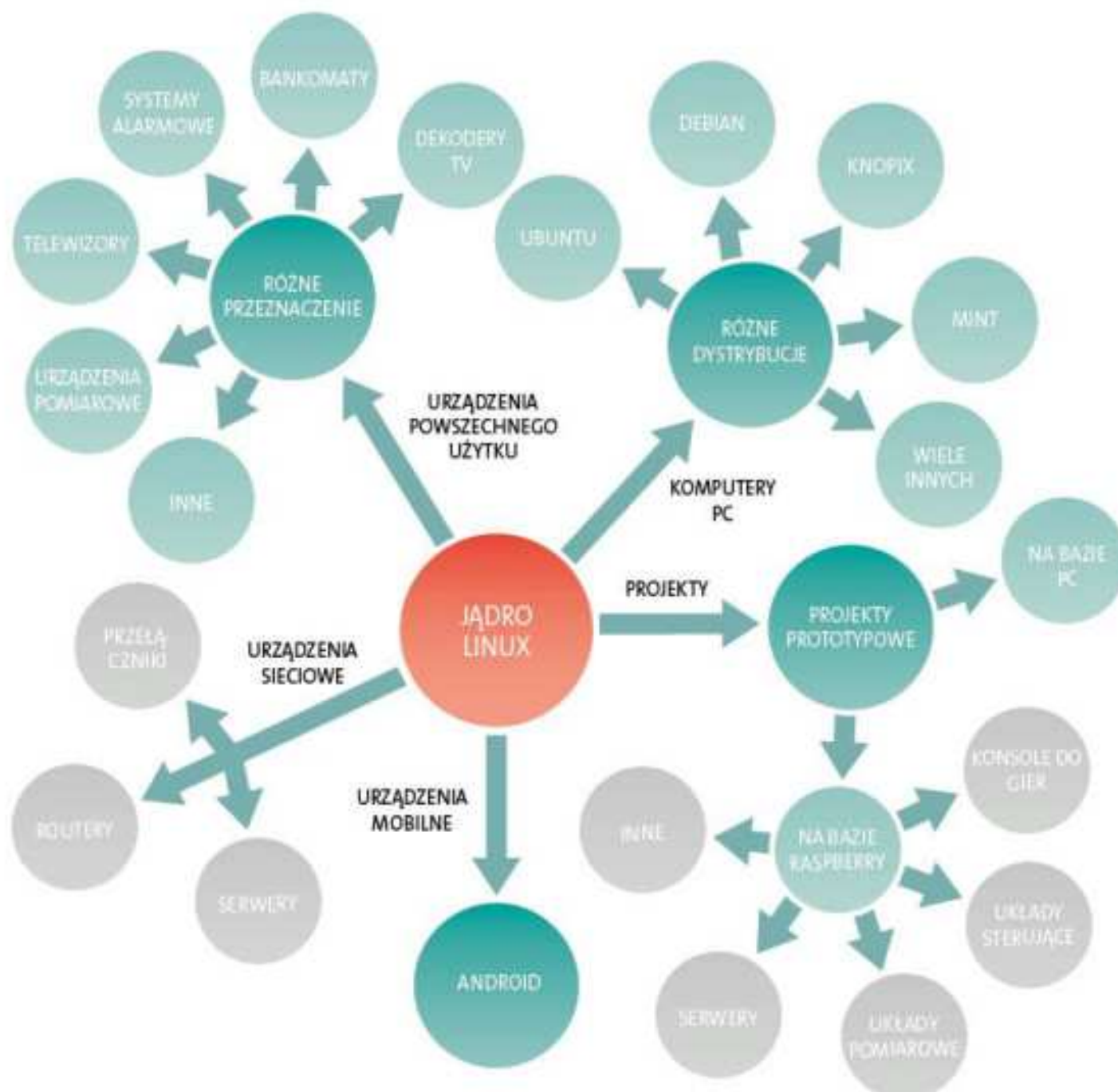
W dużej mierze Linux opierał się na komercyjnym systemie UNIX. Stąd jego nazwa nawiązująca także do imienia twórcy.



**jądro Linux
i dystrybucja**

Wiesz z doświadczenia, że istnieje wiele wersji systemu Linux. Nazywamy je **dystrybucjami**. Użycie nazwy Linux dla całej rodziny systemów jest uzasadnione, gdyż wszystkie opierają się na tym samym **jądrze Linux** (ang. *Linux kernel*) obsługującym najważniejsze funkcje, takie jak wielozadaniowość, dostęp pamięci, bibliotek, ładowanie sterowników i wiele innych.

To właśnie dzięki temu, że jądro systemu jest udostępniane na zasadach licencji GNU, powstało wiele dystrybucji o różnych zastosowaniach, a to przyczyniło się do wielkiej popularności systemu. Każdy może dobrać dystrybucję najlepiej spełniającą wymagania użytkownika i sprzętu. Nie wszyscy wiedzą, że system ten może być podstawą pracy nie tylko komputerów PC, lecz także wielu urządzeń powszechnego użytku (rys. 13.2.). Jeśli masz w domu dekodery telewizji satelitarnej lub kablowej, to istnieje duże prawdopodobieństwo, że pracuje on pod kontrolą dystrybucji Linux opracowanej przez jego producenta na bazie jądra Linux. Interesujący się elektroniką cyfrową wykorzystują niewielkie systemy mikroprocesorowe Raspberry Pi, dla których opracowano wiele dystrybucji o różnym przeznaczeniu. Powszechnie używany w telefonach Android także ma swoje korzenie w systemie Linux.



Rys. 13.2. Przykłady zastosowania systemu Linux



13.2. Jeden czy dwa systemy na dysku, czyli jaki sposób instalacji Linux wybrać

system plików

System plików jest charakterystyczną cechą każdego systemu operacyjnego. Jest on metodą zapisu, przechowywania i zarządzania plikami.

W systemie Windows powszechnie stosowany jest NTFS (ang. *New Technology File System*). Dla pamięci zewnętrznych, np. kart pamięci i pendrive'ów domyślnie jest to FAT (*File Allocation Table*).

W Linux powszechnie stosowany jest system plików Ext w kolejnych jego wersjach.

Uwaga!

Specyfikacja wymienionych w rozdziale i pozostałych **systemów plików** jest obszerna. Jeśli ciekawi cię to zagadnienie, skorzystaj z opisów zamieszczonych na wiarygodnych stronach internetowych lub w książkach.

odczyt plików z nośnika z systemem Ext

Domyślasz się, że każdy z systemów organizuje system plików podczas instalacji. Ponieważ systemy się różnią, mogą wystąpić problemy z dostępem z poziomu Windows do plików Linuxa i na odwrót. Ma to szczególne znaczenie, gdy oba systemy zainstalowano na tym samym komputerze. Na szczęście po zastosowaniu odpowiednich sterowników można pominąć tę niedogodność. Linux nie ma natomiast problemów z odczytaniem plików zapisanych na nośnikach z FAT, np. pendrive'ów.

Zajmijmy się instalacją systemu Linux na komputerach PC. Dla uproszczenia przyjmijmy, że można to zrobić na cztery sposoby.

sposoby instalacji Linux

1. **Instalacja na dysku jako jedynego systemu operacyjnego komputera.**
W jej trakcie dysk zostanie przygotowany do pracy z systemem plików Ext w wersji znamiennej dla danej dystrybucji.
2. Instalacja równolegle do systemu już zainstalowanego, np. do Windows 10.
Przy uruchomieniu komputera będzie można wybrać, który z nich ma zostać wczytany. Instalator powinien wykryć istniejący na dysku system i oczekiwać decyzji o sposobie instalacji Linux. Po zakończeniu działań i ponownym włączeniu komputera użytkownik będzie mógł wybrać jeden z systemów. Zazwyczaj domyślnie jest to Linux, a to oznacza, że po upływie pewnego czasu to on zostanie uruchomiony.
3. Instalacja wirtualnej maszynie, czyli programie emulującym pracę komputera. W tym rozwiązaniu będzie potrzebne oprogramowanie emulacyjne, np. VirtualBox. Tworzy ono środowiska interpretowane przez instalatora jako komputer gotowy do instalacji systemu operacyjnego.



4. Instalacja w pamięci zewnętrznej, np. na pendrivie lub karcie pamięci. W tym przypadku Linux zostaje uruchomiony z pominięciem systemów zainstalowanych na dysku komputera lub na komputerze bez systemu lub nawet bez dysku. Proces wymaga przygotowania nośnika za pomocą specjalnego programu.

13.3. Wirtualnie, czyli instalujemy środowisko wirtualnych komputerów

Wiesz już, że system operacyjny może zostać zainstalowany w emulatorze komputera PC. Jednym z takich darmowych i sprawdzonych środowisk jest VirtualBox. Do instalacji potrzebujesz pliku instalacyjnego, który możesz pobrać bezpośrednio ze strony producenta (www.virtualbox.org). Oczywiście można wybrać także inne programy emulacyjne, należy jednak sprawdzić opinie o stabilności ich funkcjonalności i działania. Niezależnie od wyboru programu pobierz wersję przeznaczoną dla systemu, w którym chcesz go zainstalować.

Zanim przystąpisz do instalacji, sprawdź, czy twój komputer ma włączony dostęp do technologii wirtualizacji. Zazwyczaj w systemie UEFI lub setupie BIOS opcja ta jest wyłączona. Uaktywnij ją (rys. 13.3.).

technologia
wirtualizacji

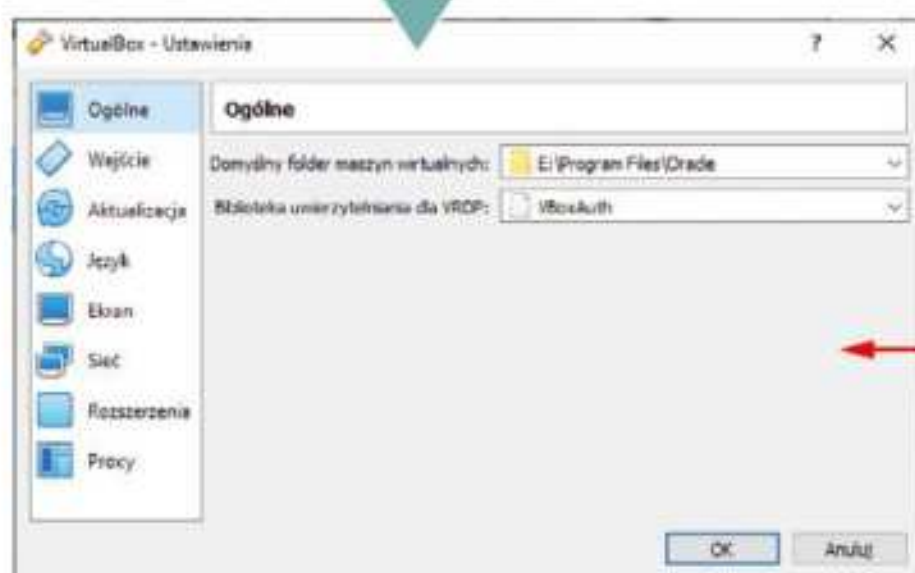


Rys. 13.3. Przykład aktywowania wirtualizacji w UEFI

Program emulacyjny jest jedynie środowiskiem umożliwiającym tworzenie wirtualnych komputerów. Każdy z nich można indywidualnie skonfigurować, przydzielając mu przestrzeń dyskową, pamięć itp. Sama instalacja przebiega automatycznie. Po uruchomieniu programu (rys. 13.4.) można przystąpić do tworzenia wirtualnego komputera i instalacji w nim wybranego systemu. Spis obsługiwanych systemów przez daną wersję VirtualBoxa znajdziesz na stronie internetowej.



Możesz utworzyć nową maszynę wirtualną lub dodać ją na podstawie pliku, np. zapisanego na innym komputerze.



Ustawienia domyślne pozwalają na szybkie uruchomienie wirtualnego komputera. Możesz jednak je modyfikować.

Rys. 13.4. Menedżer programu VirtualBox

wymagania systemu

Podstawą do prawidłowego utworzenia wirtualnego komputera jest analiza wymagań systemu, dla którego go przygotowujemy. Załóżmy, że będzie to Linux o wymaganiach jak w tabeli 13.1.

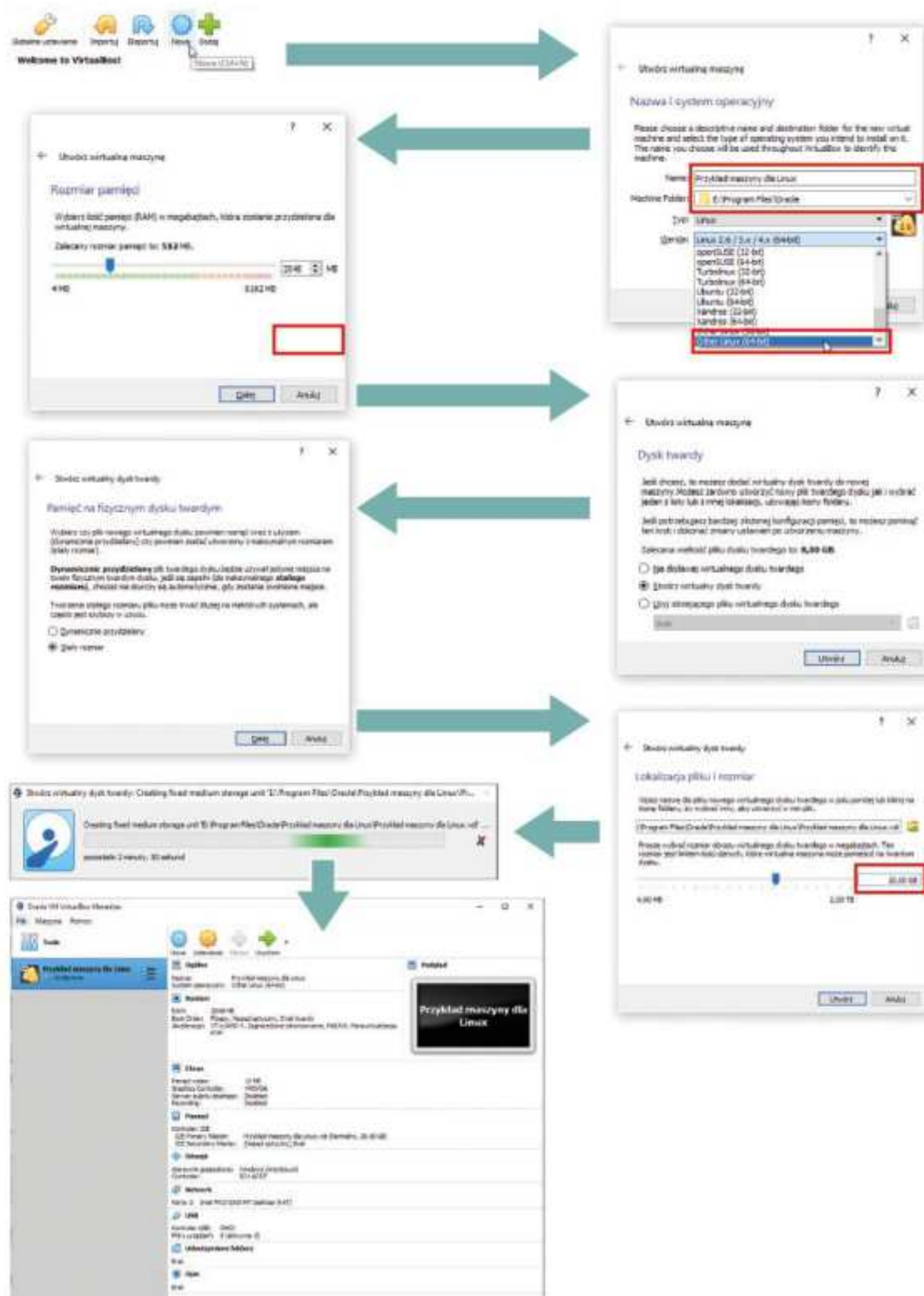
Procesor	64-bitowy lub 32-bitowy	Wersja 64-bitowa systemu nie działa na komputerach 32-bitowych
Pamięć RAM	2 GB	To wystarczająca ilość pamięci dla większości dystrybucji. Jeśli pojemność RAM komputera pozwoli na powiększenie jej dla wirtualnej maszyny, należy to zrobić.
Pojemność dysku	20 GB	Jeśli chcesz instalować dodatkowe programy lub gromadzić dane, należy zadbać o większą pojemność dysku
Karta grafiki	Dowolna o rozdzielczości nie mniejszej niż 800x600	Dla wygody pracy powinna być większa
Pamięć zewnętrzna	Obsługa przez USB lub dostęp do napędu optycznego	Napęd optyczny może być także wirtualny

Tab. 13.1. Przykładowe wymagania sprzętowe dla dystrybucji systemu Linux



instalacja VirtualBox

VirtualBox umożliwia instalowanie różnych systemów operacyjnych, dlatego w trakcie przygotowywania wirtualnego komputera należy to uwzględnić. Proces przebiega pod kontrolą kreatora. Użytkownikowi pozostaje jedynie ustawienie żądanych parametrów. W naszym przykładzie (rys. 13.5.) użyjemy danych z tabeli 13.1.



Rys. 13.5. Przykład tworzenia wirtualnej maszyny w programie VirtualBox



W trakcie przygotowywania maszyny czytaj uważnie komentarze i objaśnienia w oknach dialogowych. Pomogą w wyborze właściwej opcji.

13.4. Nie ingerujemy, czyli miejsce systemu na nośniku zewnętrznym

System Linux pobiera się zazwyczaj w postaci obrazu płyty CD lub DVD. Po przeniesieniu obrazu otrzymujemy nośnik, z którego można ładować system. Dzięki temu można go wypróbować przed podjęciem decyzji o instalacji. Tę samą funkcję może spełniać pamięć zewnętrzna (pendrive, karta pamięci itp.), np. podłączana do USB. Jednym z zastosowań takiego rozwiązania jest uruchamianie komputera w przypadku poważnej awarii zainstalowanego systemu.

Universal
USB Installer

Najprostszym sposobem przygotowania bootowalnego nośnika z systemem jest użycie odpowiedniego programu, np. **Universal USB Installer**. Wspomniane narzędzie jest darmowe i nie wymaga instalacji w systemie. Wymaga jedynie wskazania pliku z obrazem .iso i nośnika, na którym system ma być zainstalowany (rys. 13.6.). Oczywiście za jego pomocą można przygotować nośnik także z innymi niż Linux systemami.



Wybierz dystrybucję z listy. Zauważ, że na liście umieszczono bardzo wiele różnych systemów.

Wskaż miejsce, w którym został zapisany plik z obrazem .iso systemu, i wybierz nośnik docelowy.

Rys. 13.6. Przykład tworzenia zewnętrznego nośnika z systemem Linux

Uwaga!

Zwróć szczególną uwagę na wybór właściwego nośnika. Pomyłka może spowodować utratę danych ze wskazanej błędnie pamięci.

Gotowy nośnik należy umieścić w odpowiednim gnieździe (USB lub czytniku kart), uruchomić komputer i zmienić źródło wczytywania systemu.



Uwaga!

Sposób zmiany źródła systemu podczas uruchamiania komputera zależy od producenta oprogramowania płyty (UEFI lub BIOS). Zazwyczaj należy użyć jednego z klawiszy. Jeśli nie wiesz którego, zajrzyj do instrukcji użytkownika płyty głównej.

ZADANIA DO WYKONANIA

1. Opracuj krótki dokument tekstowy na temat odczytywania w Windows 10 plików zapisanych na dyskach lub partycjach z systemem Ext. Udostępnij swoje opracowanie w internecie w dowolnej formie.
2. Narysuj mapę myśli pomagającą zapamiętać metody instalacji systemu Linux na podstawie podrozdziału 13.2.
3. Zainstaluj w swoim komputerze program VirtualBox i przygotuj w nim wirtualną maszynę do instalacji Linux według danych z tabeli 13.1. Pamiętaj, by pobrać plik instalacyjny odpowiedni dla twojego komputera i systemu. Zwróć uwagę na wiarygodność źródła pobrania pliku. Porównaj efekt z ostatnim obrazem rysunku 13.5.
4. Znajdź wiarygodne źródło do pobrania programu Universal USB Installer. Pobierz go i uruchom. Sprawdź, jakie systemy możesz za jego pomocą zainstalować na zewnętrznym nośniku. Czy znasz niektóre z nich? Wymień je.
- 5*. Znajdź w internecie opis systemu Linux Ubuntu i Linux Mint w najnowszych wersjach. Porównaj w tabeli ich minimalne wymagania sprzętowe. Która z tych dystrybucji może być zainstalowana na twoim komputerze?

PODSUMOWANIE LEKCJI

- Wersje, czyli **dystrybucje** systemu Linux są oparte na wspólnym jądrze. s. 111
- **Dystrybucje Linuxa** dla komputerów osobistych są rozpowszechniane **bezpłatnie** na licencji GNU. s. 111
- System Linux jest wykorzystywany także w **urządzeniach sieciowych, powszechnego użytku, mobilnych** itp. s. 111
- **System plików** to zbiór metod zapisu i przechowywania plików oraz zarządzania nimi na nośniku, np. dysku. s. 112
- **Odczytanie w Windows** plików z dysku z systemem Ext wymaga odpowiedniego sterownika. s. 112
- **Linux można zainstalować:** s. 112
 - jako samodzielny i jedyny system operacyjny,
 - równolegle do innego systemu z możliwością wyboru systemu po włączeniu komputera,
 - w wirtualnej maszynie,
 - na nośniku zewnętrznym.
- Programy do tworzenia wirtualnych maszyn wykorzystują **technologię wirtualizacji**, którą należy odblokować w UEFI lub setupie BIOS. Domyślnie może być zablokowana. s. 113
- Podczas tworzenia wirtualnej maszyny należy wziąć pod uwagę **wymagania systemu**, dla którego ją przygotowujemy. s. 114
- Jednym z popularnych programów do tworzenia wirtualnych maszyn jest **VirtualBox**. s. 115
- Nośnik zewnętrzny z systemem można utworzyć za pomocą programu **Universal USB Installer**. s. 116



14. Wiele wersji, czyli wybieramy dystrybucję Linuxa

NA TEJ LEKCJI:

- poznasz zasady doboru dystrybucji;
- zainstalujesz Linuxa na wirtualnej maszynie.

Wybór dystrybucji Linuxa nie jest sprawą łatwą i zależy od wielu czynników. Gdy system zostanie już wybrany, pozostanie wypróbować go i zainstalować na komputerze. Wiesz już, jak przygotować dla niego miejsce. Przyszedł czas na instalację.

14.1. Trudna decyzja, czyli jak wybrać dystrybucję

Wybór dystrybucji musi poprzedzić analiza potrzeb użytkownika i możliwości komputera. Dla uproszczenia pominę przypadki poszukiwania systemu dla serwera, routera i innych zastosowań niezwiązanych bezpośrednio z komputerem osobistym.

Założmy, że użytkownikiem będzie student potrzebujący dostępu do przeglądarki internetowej, edytora tekstu i grafiki oraz arkusza kalkulacyjnego.

GUI

System powinien mieć także wygodny graficzny interfejs użytkownika, czyli **GUI** (ang. *graphical user interface*). Dzięki temu obsługa będzie prosta, oparta na wykorzystaniu pulpitu, okien i ikon. Ponieważ nasz student nie chce zajmować się instalacją dodatkowego oprogramowania, system po zainstalowaniu powinien zawierać wspomniane programy. Oczywiście dystrybucja powinna także mieć opinię stabilnej, bezpiecznej i niezawodnej oraz zawierać sterowniki do podstawowych urządzeń i elementów komputera, np. karty sieciowej, dźwiękowej itp. Te cechy można sprawdzić, czytając fora użytkowników i artykuły w prasie komputerowej. Oczywiście przed wyborem należy przeprowadzić próbę i dokładnie przetestować system. W tym celu po pobraniu obrazu .iso można zapisać płytę lub umieścić instalację na pendrivie. Alternatywnym sposobem, niewymagającym dodatkowego sprzętu, jest uruchomienie systemu z obrazu .iso w wirtualnej maszynie.

64 czy 32 bity

Jak dopasować dystrybucję do komputera? Wymagania sprzętowe poszczególnych dystrybucji są szczegółowo opisane. Warto jednak zwrócić uwagę na to, czy dana wersja dystrybucji jest przeznaczona dla **procesorów 32-bitowych czy 64-bitowych**. Oprogramowanie 64-bitowe najczęściej nie działa na procesorach z architekturą 32-bitową.



Rys. 14.1. Przykładowe wymagania użytkownika w stosunku do dystrybucji Linux

W naszych przykładach użyjemy popularnej dystrybucji spełniającej wymagania studenta: **Linux Mint**. Obraz .iso można pobrać z oficjalnej strony tej dystrybucji. Uwaga! Wymagania studenta spełniają także inne dystrybucje. Wraz z postępem technologicznym i rozwojem systemów operacyjnych będą pojawiać się nowe, atrakcyjniejsze i sprawniejsze dystrybucje. Linux Mint był stabilny i zgodny z wymaganiami w czasie pisania podręcznika.

14.2. Ostateczna decyzja, czyli instalujemy i testujemy dystrybucje

Instalacja systemu na pendrivie lub karcie pamięci została omówiona w poprzednim rozdziale. Wiesz już, jak utworzyć wirtualny komputer w programie VirtualBox (rys. 14.2.). Czas na uruchomienie i instalację systemu w maszynie wirtualnej.

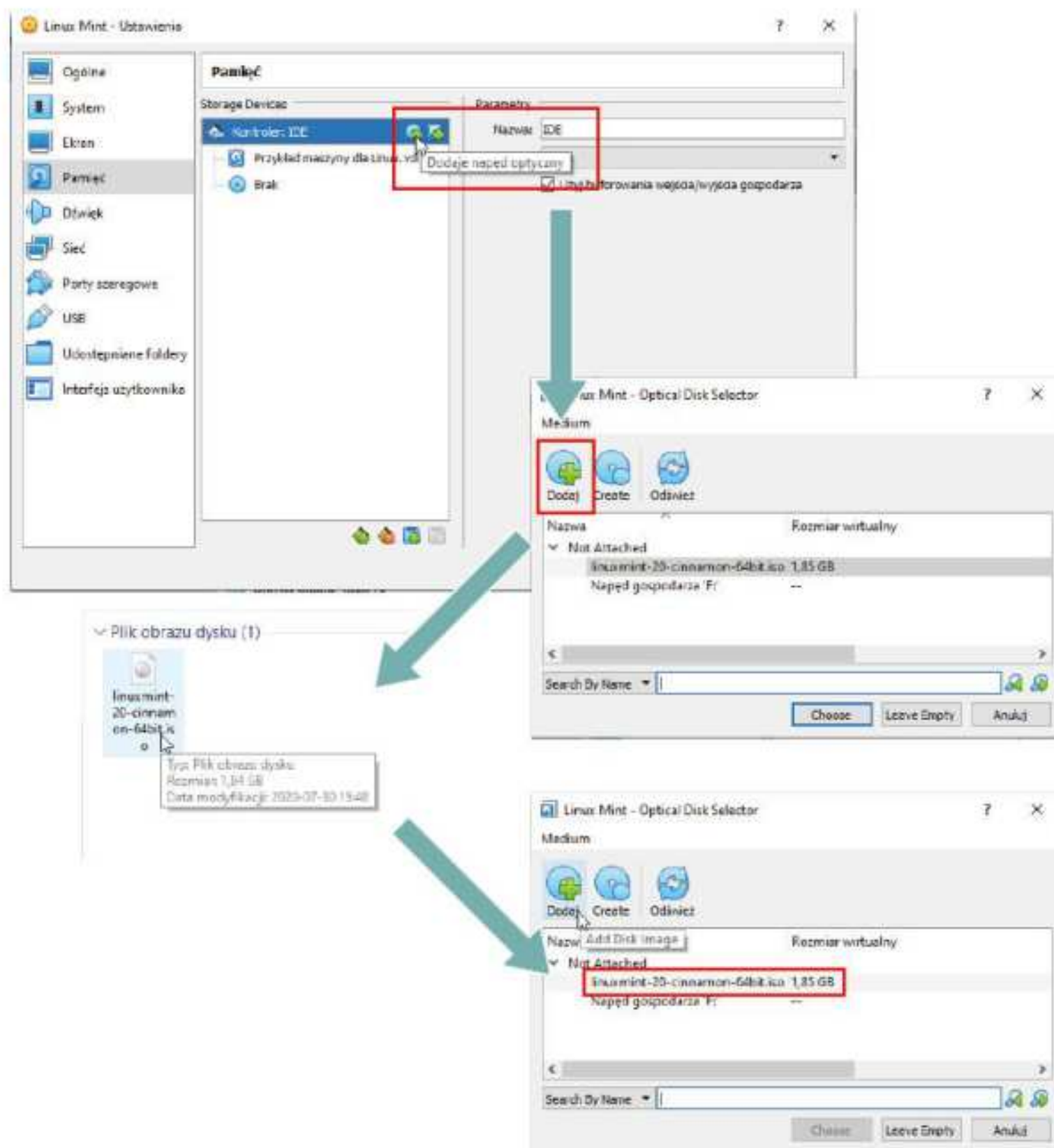


Rys. 14.2. Przykładowy wirtualny komputer utworzony zgodnie z wymaganiami dystrybucji



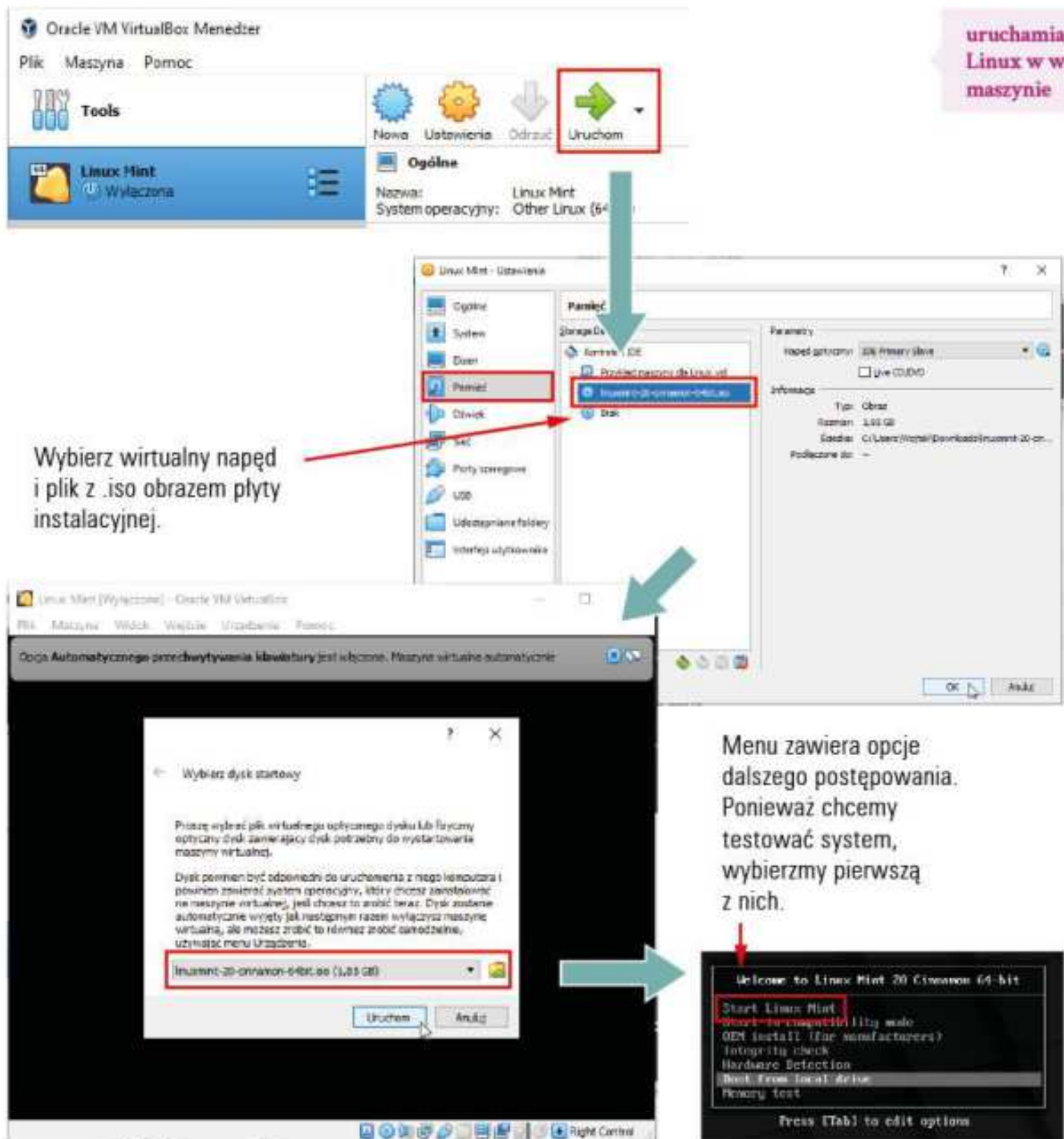
dodawanie wirtualnego napędu

Ponieważ nie mamy płyty z wypalonym obrazem, utworzymy wirtualny napęd CD. Dzięki temu nasz plik .iso będzie interpretowany przez wirtualną maszynę jako napęd optyczny z płytą (rys. 14.3.).



Rys. 14.3. Przykład dodawania wirtualnego napędu optycznego w programie VirtualBox

Po dodaniu wirtualnego napędu optycznego można przystąpić do uruchomienia instalatora (rys. 14.4.).



Rys. 14.4. Uruchamianie Linux Mint w wirtualnej maszynie z użyciem obrazu płyty instalacyjnej

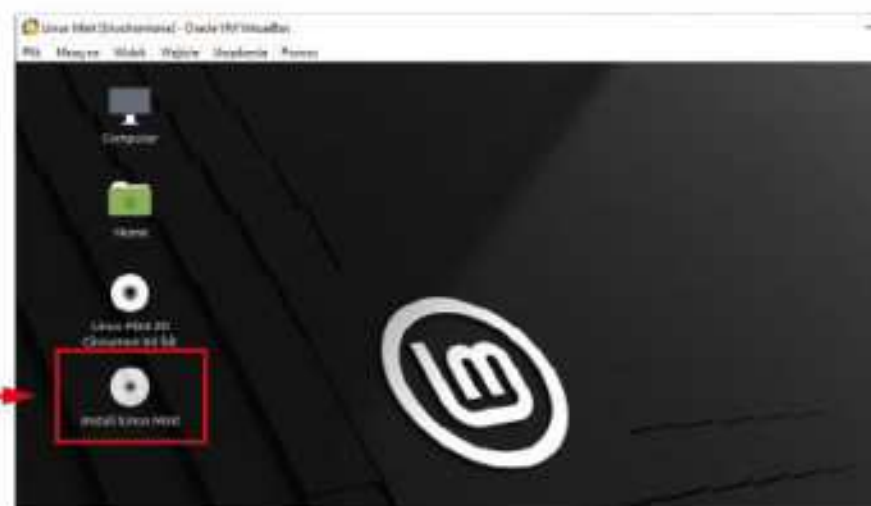
Po opisanych na ilustracji czynnościach system będzie działał na maszynie wirtualnej. Nie jest on jednak zainstalowany na jej dysku. Można go za to testować.

Jeśli podejmiesz decyzję o instalacji, wybierz ikonę instalatora systemu (rys. 14.5.). W czasie instalacji możesz wybrać niektóre opcje, np. wersję językową. Na pewnym etapie należy także podać dane użytkownika i wyświetlaną nazwę komputera, a także hasło dostępu. Proces instalacji może potrwać kilka lub kilkanaście minut.

instalacja Linuxa



Uruchomienie
procesu instalacji
systemu na dysku.



Rys. 14.5. Pulpit Linux Mint z ikoną uruchamiającą proces instalacji

Uwaga!

Proces uruchamiania instalacji Ubuntu i wielu innych dystrybucji przebiega w podobny sposób.

ZADANIA DO WYKONANIA

1. Pobierz plik instalacyjny w postaci obrazu płyty .iso dystrybucji Mint. Uruchom ten system wirtualnej maszynie. Jeśli uznasz go za atrakcyjny, zainstaluj na wirtualnym komputerze.
2. Pobierz plik instalacyjny w postaci obrazu płyty .iso dystrybucji Ubuntu. Uruchom ten system wirtualnej maszynie. Jeśli uznasz go za atrakcyjny, zainstaluj na wirtualnym komputerze.
3. Wykonaj zadania 1 i 2 oraz porównaj oba systemy. Swoje spostrzeżenia zapisz w dokumencie tekstowym. Zwróć uwagę na wygodę pracy z systemem i jakość aplikacji.
4. Bazując na pobranym pliku z systemem Ubuntu lub Mint, skonfiguruj pendrive w taki sposób, by z niego uruchamiał się system. Sprawdź działanie takiego rozwiązania na kilku różnych komputerach. Jeśli wystąpiły błędy lub problemy, to je opisz.
- 5*. Odwiedź strony użytkowników systemu Linux oraz magazynów komputerowych i przeczytaj opinie o innych popularnych dystrybucjach Linuxa. Opisz krótko cechy systemu, który wzbudził twoje zainteresowanie. Uzasadnij swój wybór.

PODSUMOWANIE LEKCJI

- s. 118 • Linux stał się przyjazny dla niezaawansowanego użytkownika dzięki **powłoce graficznej GUI**.
- s. 118 • Wybór wersji danej dystrybucji zależy także od architektury procesora. Wersje **64-bitowe** nie będą poprawnie działały w komputerach z procesorem **32-bitowym**.
- s. 120 • Na wybór dystrybucji mają wpływ także **oczekiwania użytkownika** systemu.
- s. 120 • W wirtualnym komputerze można montować **wirtualne napędy**, np. optyczne.
- s. 121 • Do uruchomienia lub instalacji systemu na wirtualnej maszynie nie potrzebujesz nośnika z obrazem instalatora. Zastąpi go **wirtualny napęd**.
- s. 121 • Po zapoznaniu się z działaniem dystrybucji można wyłączyć maszynę lub **zainstalować system**.



15. Bez kosztów, czyli programy w Linux

NA TEJ LEKCJI:

- przeanalizujesz zasoby programowe niektórych dystrybucji Linux;
- dowiesz się, jak znaleźć i instalować nowe programy w Linux.

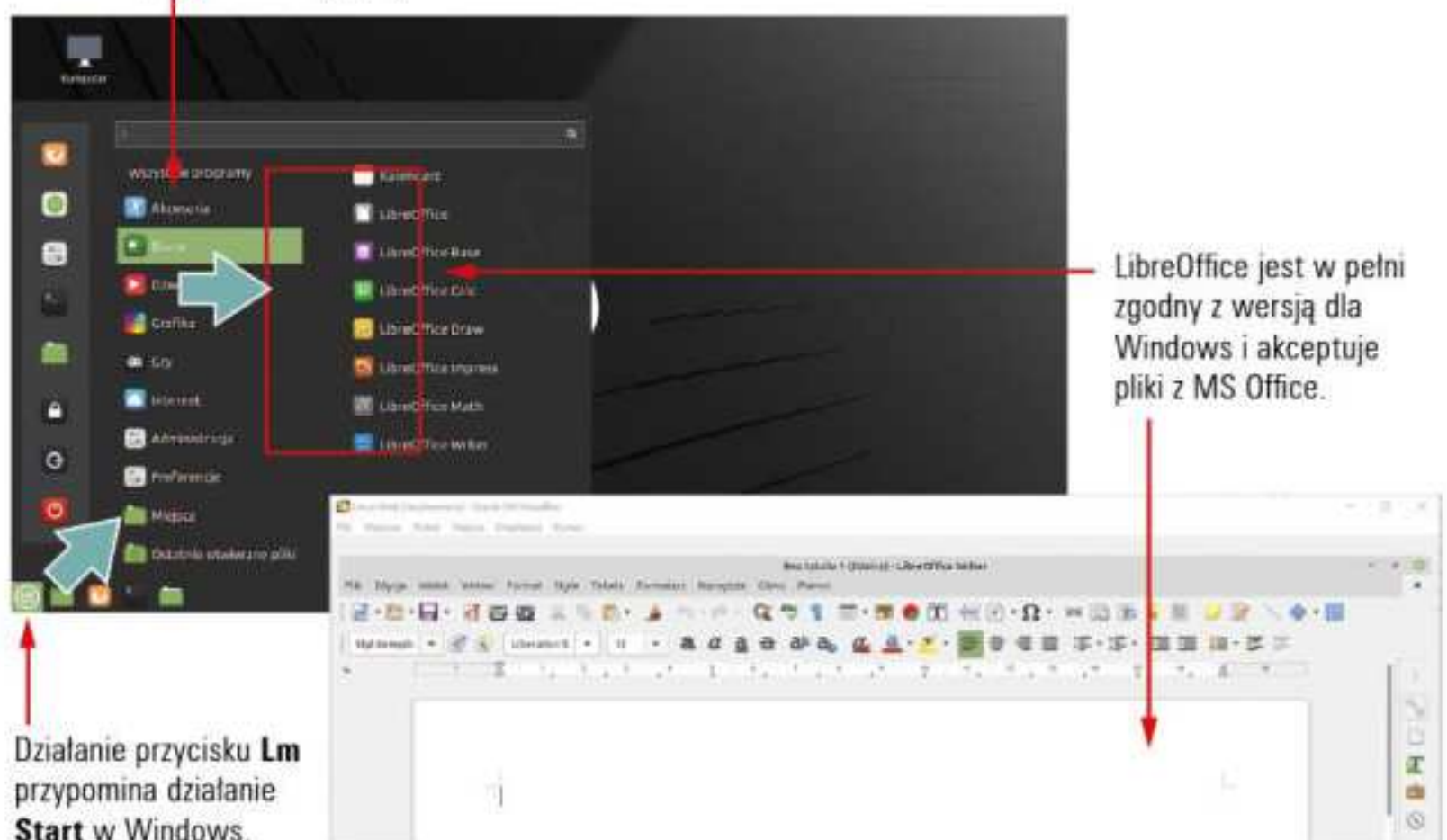
Wiesz, że programy komputerowe są rozpowszechniane w różny sposób. Można kupić licencję wraz z nośnikiem, kupić klucz aktywowania pobranego z sieci programu lub uruchamiać go on-line. Podobnie jest w przypadku programów dla Linux. Większość z nich jest udostępniana bezpłatnie. Część oprogramowania stanowi zawartość dystrybucji Linuxa. Sprawdźmy, jak to wygląda w praktyce.

15.1. Co mamy, czyli programy instalowane wraz z dystrybucją

Twój Linux już działa. Czas na zapoznanie się z programami, które są dostępne po instalacji dystrybucji. W zależności od dystrybucji i rodzaju **GUI**, znajdziesz je w różnych miejscach. Masz jednak duże doświadczenie w pracy z Windows, a większość powłok **GUI** opiera się na podobnych zasadach. Przyjrzyjmy się **programom w dystrybucji Mint** (rys. 15.1.) i **Ubuntu** (rys. 15.2.).

programy
w Linux z GUI

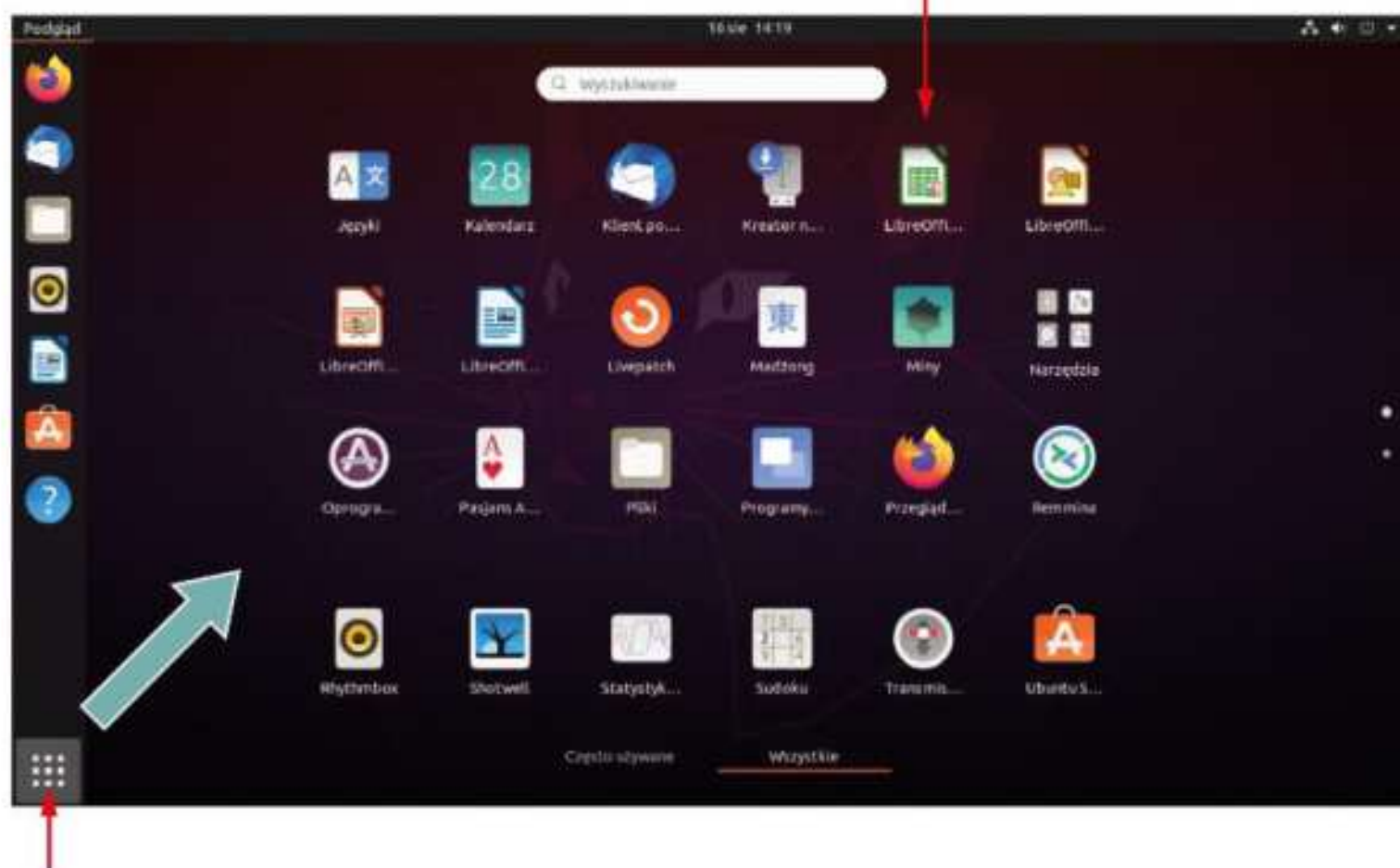
Programy w Mint pogrupowano.



Rys. 15.1. Programy dystrybucji Linux Mint



W Ubuntu także jest dostępny LibreOffice.



W Ubuntu także możesz użyć przycisku spełniającego funkcję **Start**.

Rys. 15.2. Programy dystrybucji Linux Ubuntu

Większość dystrybucji wyposażonych w GUI zawiera podstawowe oprogramowanie biurowe, np. LibreOffice, graficzne, narzędziowe i związane z usługami internetowymi, np. przeglądarki lub klientów poczty.

Analizując ilustracje 15.1. i 15.2., dojdziemy do wniosku, że do podstawowych kryteriów wyboru danej dystrybucji należy dopisać liczbę i jakość już zainstalowanych programów. Nie chodzi przy tym, by było ich jak najwięcej, ale by spełniały oczekiwania użytkownika.

Wśród programów wielu dystrybucji znajdziesz przeglądarki, np. Firefox, klientów poczty, np. Thunderbird. Niewątpliwie najwięcej miejsca na dysku zajmie LibreOffice, ale jest to obecnie najpopularniejszy darmowy pakiet biurowy używany także w systemie Windows.

Linux i chmury

Oczywiście za pośrednictwem przeglądarki można korzystać z chmur, np. Google, OneDrive, Office 365. Daje to możliwość edytowania dokumentów programów Office 365. Podobnie rzecz ma się z innymi programami dostępnymi w chmurach – np. komercyjnymi i darmowymi edytorami grafiki lub grami. Taki sposób udostępniania i licencjonowania programów jest coraz powszechniejszy, a więc dostępny także dla użytkowników Linux. Do niedawna to właśnie brak profesjonalnych programów był główną przyczyną niechęci do instalacji Linux w firmach i na komputerach prywatnych. Jak widać, przestaje to już być problemem.

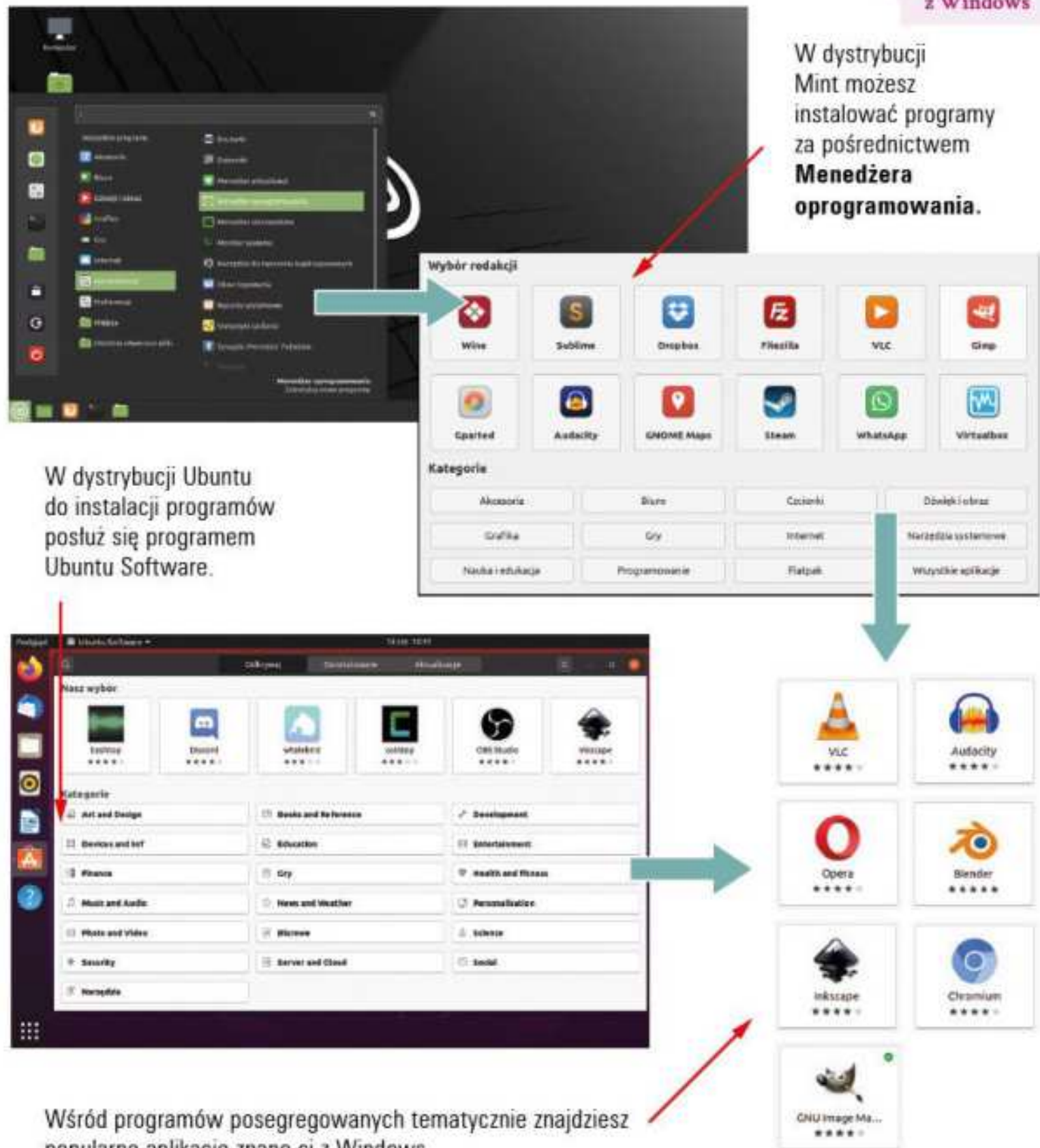


15.2. Poszerzamy możliwości, czyli jak instalować programy w Linux

W systemie Linux, podobnie jak w Windows, można także zainstalować programy, których nie ma w podstawowym zestawie. W poszczególnych dystrybucjach proces ten może się różnić. Systemy wykorzystujące GUI udostępniają w tym celu specjalną aplikację, która pomaga w wyszukaniu i instalacji. Część programów dostępnych dla danej dystrybucji znasz z systemu Windows (rys. 15.3.).

programy znane z Windows

W dystrybucji Mint możesz instalować programy za pośrednictwem **Menedżera oprogramowania**.



Rys. 15.3. Przykłady programów do instalacji oprogramowania w dystrybucji Ubuntu i Mint

podobieństwa

Zauważalne są pewne podobieństwa użycia programów z ilustracji 15.3. do działania **Sklepu Play** w urządzeniach z systemem Android lub **App Store** w iPhonech, iPadach lub iPodach. To bardzo wygodne rozwiązanie zapewniające także większe bezpieczeństwo dzięki wstępnej weryfikacji.

Wśród programów, które z racji swoich zalet i darmowej licencji zyskały duże uznanie, jest edytor grafiki rastrowej GIMP. Prześledźmy jego instalację za pomocą **Menedżera oprogramowania** (rys. 15.4.).



Rys. 15.4. Instalacja programu w dystrybucji Mint na przykładzie edytora GIMP

Przykład przedstawiony na rysunku dotyczy instalacji w systemie Linux Mint. Dla wielu innych dystrybucji instalacja przebiega bardzo podobnie i nie będzie dla ciebie problemem.



ZADANIA DO WYKONANIA

1. Korzystając z edytora tekstu, utwórz tabelę składającą się z trzech kolumn. W pierwszej z nich umieść nazwy programów, z których korzystasz na co dzień w szkole lub w domu. Mogą to być edytory tekstu, grafiki, odtwarzacze muzyczne i video itp. W wybranej dystrybucji Linux przejdź do programu przeznaczonego do instalacji oprogramowania, np. w Mint – **Menedżera Oprogramowania**. Sprawdź, które z wymienionych programów znajdziesz na liście i możesz zainstalować. Zaznacz to w drugiej kolumnie przy danej nazwie. W trzeciej kolumnie wpisz programy z listy Menedżera, które mogą zastąpić programy z pierwszej kolumny.
2. Wśród programów dostępnych dla twojej dystrybucji znajdź edytor grafiki wektorowej Inkscape. Zainstaluj go w swojej dystrybucji, a także w systemie Windows. Porównaj obie wersje i opisz ewentualne różnice.
3. W edytorze Word lub w chmurze Office 365 utwórz dokument tekstowy zawierający tabelę, ilustrację i sformatowany na różne sposoby tekst (różne czcionki, wielkość itp.). Plik zapisz w formacie .DOCX. Uruchom edytor LibreOffice w jednej z dystrybucji Linux. Sprawdź, czy twój dokument zostanie poprawnie otwarty do edycji w LibreOffice Write. W osobnym dokumencie zapisz wnioski. Przedstaw w nich wszelkie uwagi o dokumencie wyświetlonym w LibreOffice Write w porównaniu z oryginałem. Dokument zilustruj zrzutami ekranowymi.
4. Wykorzystaj arkusz z rozdziału 5. Otwórz go w LibreOffice Calc w wybranej dystrybucji Linux. Obserwuj różnice pomiędzy dokumentem wyświetlonym w Excel i Calc. Porównaj działanie formuł i wyniki obliczeń. W osobnym dokumencie zapisz wnioski. Zawrzyj w nich wszelkie uwagi o dokumencie wyświetlonym w LibreOffice Calc w porównaniu z oryginałem. Dokument zilustruj zrzutami ekranowymi.
- 5*. Porównaj najnowsze wersje przeglądarek zainstalowanych w Windows i Linux. Oceń ich zgodność. Wskaż ewentualne różnice. Wnioski zapisz w dokumencie tekstowym ilustrowanym zrzutami ekranowymi.

PODSUMOWANIE LEKCJI

- Dystrybucje z GUI zazwyczaj zawierają także **zestaw podstawowych** programów, takich jak przeglądarka internetowa, edytor tekstu, arkusz kalkulacyjny itp. s. 123
- Za pośrednictwem przeglądarek w systemie Linux **można korzystać z chmur**, takich jak Google, Office 365 itp. s. 124
- W dostępnym oprogramowaniu dystrybucji Linux **znajdują się znane i popularne programy** dla Windows, np. GIMP, VLC, Blender. s. 125
- Menedżery oprogramowania z Linux działaniem **przypominają Sklep Play lub AppStore**. s. 126



16. Tryb tekstowy jest ważny, czyli poznajemy konsolę Linux

NA TEJ LEKCJI:

- dowiesz się, jaką funkcję pełni terminal i konsola w środowisku Linux;
- utworzysz nowe konto użytkownika;
- nauczysz się zmieniać hasła użytkowników.

Dystrybucje Linux z powłoką graficzną oferują wiele aplikacji, które pozwalają na zarządzanie katalogami, plikami, użytkownikami itp. Jest to bardzo wygodne, szczególnie dla osób posługujących się także Windows. Funkcjonuje jednak wiele dystrybucji pozbawionych GUI. Wtedy konieczna jest znajomość poleceń wydawanych przez konsolę. Ale czy tylko w tym przypadku?

16.1. Konsola czy terminal, czyli jak to się zaczęło

konsola i terminal

Jedynym sposobem komunikowania się z systemem i programami jest **Konsola**. W dystrybucjach z GUI emulatorem konsoli jest **Terminal**. To okno z możliwością wpisywania poleceń i odczytywania komunikatów systemu.

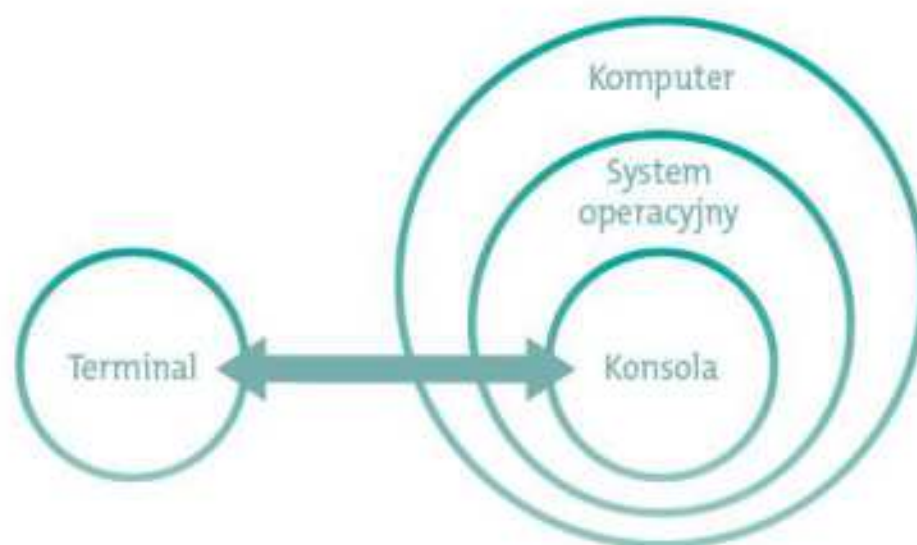
powłoka systemowa

Definicja powłoki systemowej według wikipedia.pl:

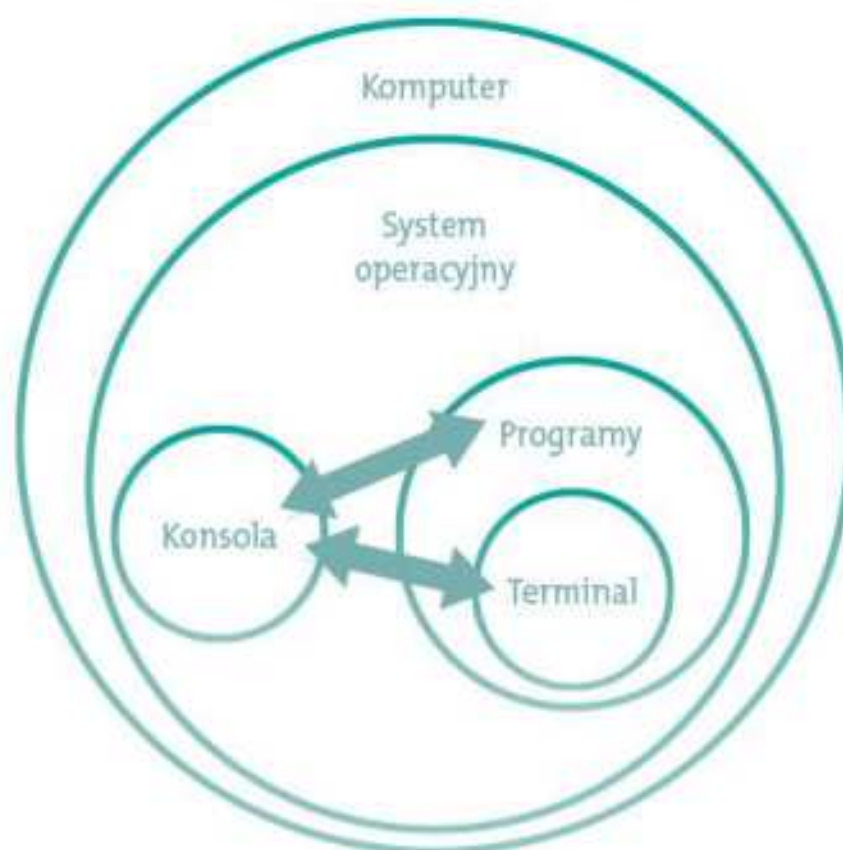
Powłoka systemowa (ang. shell) – program komputerowy pełniący funkcję pośrednika pomiędzy systemem operacyjnym lub aplikacjami a użytkownikiem, przyjmując jego polecenia i „wyprowadzając” wyniki działania programów. To pośrednictwo nie jest obowiązkowe (programy mogą być bardziej „samodzielne”).

skąd wzięła się konsola

Konsola jest powłoką systemową, a pomysł takiego rozwiązania ma swoje źródło w budowie pierwszych komputerów. Wówczas funkcję konsoli pełniło urządzenie składające się w pierwszych wersjach z klawiatury i elektrycznej maszyny do pisania, a w późniejszym czasie – z klawiatury z monitorem tekstowym. Takie urządzenie było połączone z komputerem za pośrednictwem portu szeregowego i było jedynym kanałem komunikowania się człowieka z komputerem. Terminale podobnego typu są stosowane także współcześnie nie tylko do komunikowania się operatora bezpośrednio z systemem komputera, systemem mikroprocesorowym, centralą telefoniczną itp., lecz także w kioskach informacyjnych, automatach sprzedających bilety, stanowiskach do korzystania z internetu w miejscach publicznych, np. na uczelni itp. Terminalem jest także czytnik kart płatniczych w sklepie. Każde z tych urządzeń pełni funkcję pośrednika pomiędzy użytkownikiem a komputerem, na którym dane operacje są wykonywane. Takie terminale spotykasz na co dzień (rys. 16.1.). Jeśli odniesiesz ich działanie i zastosowanie do powłoki systemu Linux, zrozumiesz funkcje i przeznaczenie jego konsoli i terminala. Należy jednak założyć, że z powłoki systemowej korzystają także inne programy i aplikacje (rys. 16.2.).



Rys. 16.1. Przykładowy schemat współpracy zewnętrznego terminala z komputerem



Rys. 16.2. Przykładowy schemat relacji pomiędzy terminalem jako programem a konsolą systemu operacyjnego

16.2. Pierwsze kroki z konsolą, czyli jak uruchomić Terminal w Linux

Skoro rozumiesz funkcję, jaką pełni terminal, sprawdź jego działanie w praktyce. Po zainstalowaniu Linux możesz utworzyć konta użytkowników i nadać im odpowiednie uprawnienia. To zagadnienie znasz już z Windows.

Uruchomienie terminala:

Sposób 1

1. Po uruchomieniu systemu Linux, np. Mint lub Ubuntu, użyj kombinacji klawiszy **Alt+F2**. Otworzy się okienko wprowadzania poleceń. W poszczególnych dystrybucjach może się ono różnić wyglądem.



2. Wpisz jedno z poleceń `rxvt`, `xterm`, `gnome-terminal` lub `konsole` i potwierdź klawiszem **Enter**. **Terminal** konsoli zostanie otwarty – jest ona gotowa do pracy (rys. 16.3.).

Po wciśnięciu **Alt+F2** wyświetli się okno wprowadzania systemowi poleceń.



Uruchomienie **Terminala** w Linux Mint



Uruchomienie **Terminala** w Linux Ubuntu

Rys. 16.3. Uruchomienie **Terminala** w Mint i Ubuntu za pomocą okna poleceń

Sposób 2.

Użyj programu **Terminal** z menu programów (rys. 16.4.). Z tego sposobu można skorzystać w systemach z graficznym interfejsem użytkownika.



Ikony aplikacji **Terminal** w Ubuntu znajdziesz w grupie programów **Narzędzia**.



Ikony aplikacji **Terminal** w Mint znajdziesz w grupie programów **Administracja**.

Rys. 16.4. Uruchomienie **Terminala** w Mint i Ubuntu przez wybranie ikony aplikacji



W oknie terminala można wydawać polecenia systemowi i odczytywać jego komunikaty, np. potwierdzenia wykonania polecenia (rys. 16.5.).



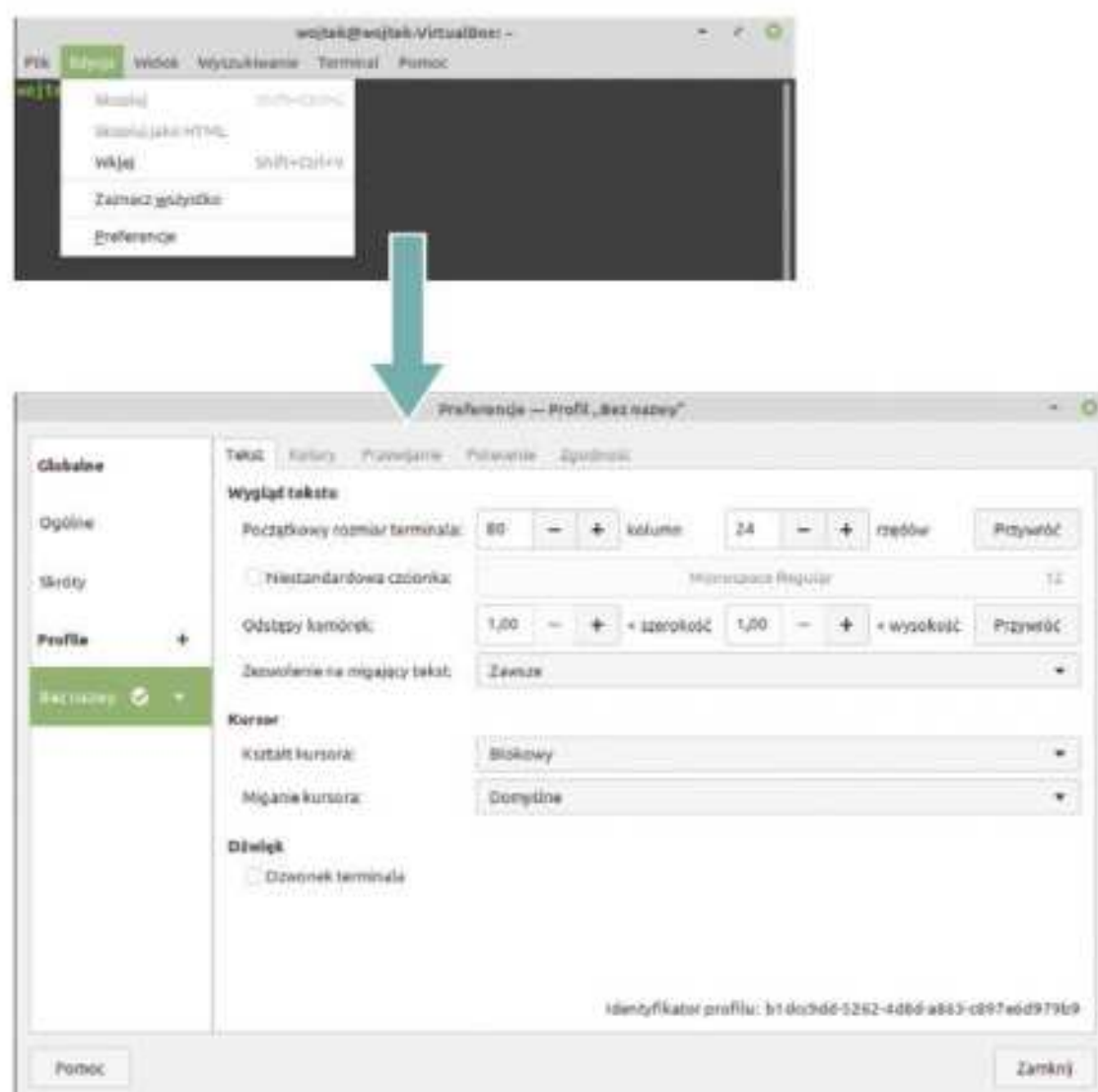
wojtek – nazwa zalogowanego użytkownika

wojtek-VirtualBox – nazwa komputera

\$ – oznacza, że użytkownik nie ma uprawnień administratora

Rys. 16.5. **Terminal** gotowy do pracy

Gotowy do pracy **Terminal** można skonfigurować (rys. 16.6.) i przystosować do własnych preferencji.



Rys. 16.6. Okno ustawienia preferencji okna **Terminala**



16.3. Nowe konto użytkownika, czyli używamy terminala

Uwaga!

W systemie Linux duże znaczenie ma zapis nazw kont, katalogów plików itp. Należy zwrócić uwagę na duże i małe litery, np. konto muzyk nie jest tym samym kontem co Muzyk.

Jednym z elementów bezpiecznej pracy w systemie Linux jest używanie konta użytkownika bez uprawnień administratora. Takie właśnie zostało założone podczas instalacji systemu (wbudowanym i domyślnym kontem administratora jest root).

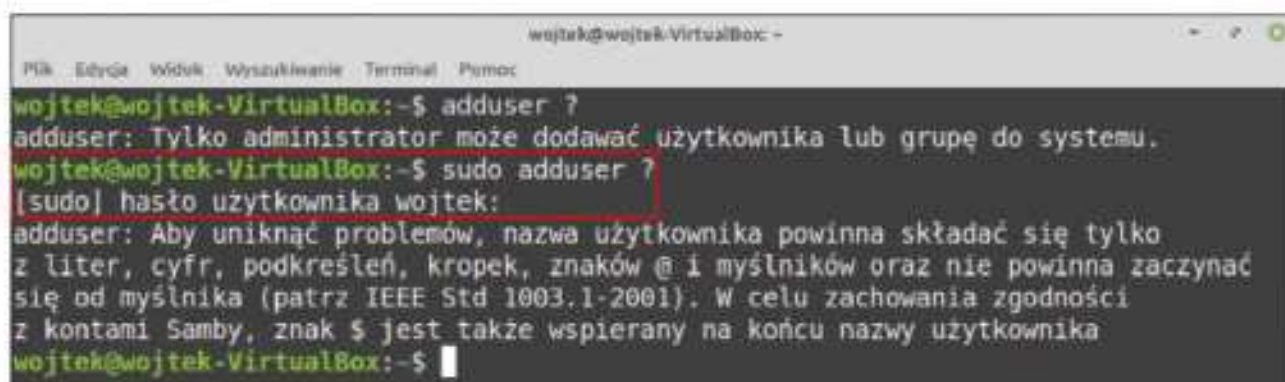
Nie musisz jednak zmieniać aktywnego konta, by dokonać zmian w systemie, np. założyć nowe konto. Konieczne będzie jednak uruchomienie programu **sudo**. Ponieważ wszystko odbywa się w terminalu, wystarczy, że wpiszesz jego nazwę.

W celu założenia nowego konta użyjemy polecenia **adduser**. Ponieważ to polecenie wymaga uprawnień administratora, należy uruchomić program **sudo**.

Definicja **sudo** według wikipedia.pl:

sudo (z ang. *super user do*) – program stosowany w systemach operacyjnych GNU/Linux, Unix i podobnych w celu umożliwienia użytkownikom uruchomienia aplikacji jako inny użytkownik systemu. Najczęściej służy ono do uruchamiania aplikacji zarezerwowanych dla administratora zwanego rootem.

Jeśli w poleceniu **adduser** użyjesz niewłaściwej nazwy użytkownika, np. zawierającej niedozwolone znaki, Terminal pomoże ci w poprawieniu błędu (rys. 16.7).



```
wojtek@wojtek-VirtualBox: ~$ adduser ?
adduser: Tylko administrator może dodawać użytkownika lub grupę do systemu.
wojtek@wojtek-VirtualBox: ~$ sudo adduser ?
[sudo] hasło użytkownika wojtek:
adduser: Aby uniknąć problemów, nazwa użytkownika powinna składać się tylko
z liter, cyfr, podkreśleń, kropek, znaków @ i myślników oraz nie powinna zaczynać
się od myślnika (patrz IEEE Std 1003.1-2001). W celu zachowania zgodności
z kontami Samby, znak $ jest także wspierany na końcu nazwy użytkownika
wojtek@wojtek-VirtualBox: ~$
```

Rys. 16.7. Przykład użycia systemu pomocy w Terminalu Linux

Uwaga!

Jeśli jakiegokolwiek polecenie będzie sprawiało ci kłopoty, możesz skorzystać z opcji pomocy. Wystarczy, że do polecenia dodasz **-h** lub **--help**.

Skoro wiesz już, jakie warunki powinna spełniać prawidłowa nazwa użytkownika, załóż konto o nazwie **konto_do_nauki** (rys. 16.8).



```

wojtek@wojtek-VirtualBox:~$ sudo adduser konto_do_nauki
Dodawanie użytkownika "konto_do_nauki"...
Dodawanie nowej grupy "konto_do_nauki" (1001)...
Dodawanie nowego użytkownika "konto_do_nauki" (1001) w grupie "konto_do_nauki"...
Tworzenie katalogu domowego "/home/konto_do_nauki"...
Kopiowanie plików z "/etc/skel" ...
Nowe hasło :
Proszę ponownie wpisać nowe hasło :
passwd: hasło zostało zmienione
Zmieniam informacje o użytkowniku konto_do_nauki
Wpisz nową wartość lub wciśnij ENTER by przyjąć wartość domyślną
Imię i nazwisko []: Wojtek Uczeń
Numer pokoju []: 115
Telefon do pracy []: 123456789
Telefon domowy []: 87654321
Inne []: konto do ćwiczeń z Terminalem
chfn: name with non-ASCII characters: 'Wojtek Uczeń'
chfn: 'konto do ćwiczeń z Terminalem' contains non-ASCII characters
Czy informacja jest poprawna? [T/n] T
wojtek@wojtek-VirtualBox:~$

```

Rys. 16.8. Proces tworzenia nowego konta użytkownika w Linux

W trakcie zakładania konta czytaj dokładnie komunikaty systemu. Będzie konieczne wybranie hasła. Wpisywanie informacji o użytkowniku można pominąć. Możesz także wpisać tylko niektóre z nich, np. imię i nazwisko.

Sprawdźmy, czy konto zostało założone. Zmieńmy aktualne konto używane w **Terminalu** z **wojtek** na **konto_do_nauki** za pomocą polecenia **su** (rys. 16.9).

Jeśli chcesz **usunąć z ekranu Terminala** dotychczasowe polecenia i komunikaty, użyj polecenia **clear**.

zmiana konta
obsługiwanego
w Terminalu

czyszczenie okna
Terminala

```

konto_do_nauki@wojtek-VirtualBox: /home/wojtek
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
wojtek@wojtek-VirtualBox:~$ su konto_do_nauki
Hasło:
konto_do_nauki@wojtek-VirtualBox:/home/wojtek$

```

Rys. 16.9. Zmiana aktywnego użytkownika w Linux za pomocą polecenia **su**

Zmieńmy teraz hasło dla użytkownika **konto_do_nauki**. Użyjemy polecenia **passwd** (rys. 16.10).

zmiana hasła
użytkownika

```

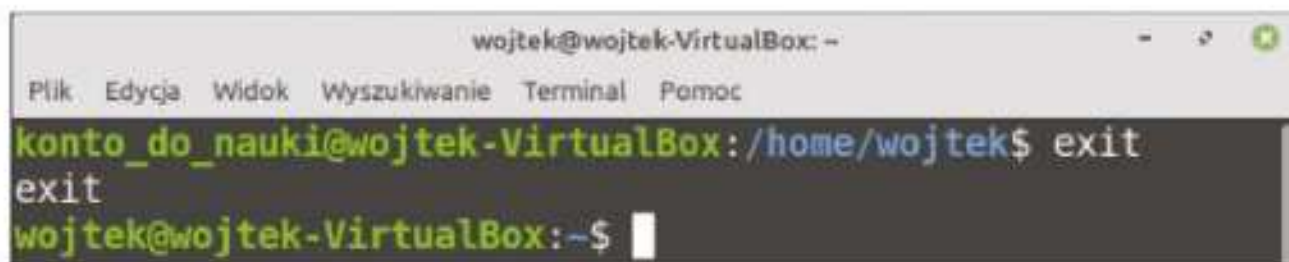
konto_do_nauki@wojtek-VirtualBox: /home/wojtek
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
konto_do_nauki@wojtek-VirtualBox:/home/wojtek$ passwd konto_do_nauki
Zmienianie hasła dla konto_do_nauki.
Current password:
Nowe hasło :
Proszę ponownie wpisać nowe hasło :
passwd: hasło zostało zmienione
konto_do_nauki@wojtek-VirtualBox:/home/wojtek$

```

Rys. 16.10. Zmiana hasła dla konta użytkownika poleceniem **passwd**



Wróćmy teraz do konta **wojtek**. W tym celu należy zakończyć użytkowanie Terminala przez użytkownika **konto_do_nauki**. Użyjmy polecenia `exit`, by kolejne polecenia dotyczyły użytkownika **wojtek** (rys. 16.11.).



```
wojtek@wojtek-VirtualBox: ~  
Plik  Edycja  Widok  Wyszukiwanie  Terminal  Pomoc  
konto_do_nauki@wojtek-VirtualBox: /home/wojtek$ exit  
exit  
wojtek@wojtek-VirtualBox: ~$
```

Rys. 16.11. Użycie polecenia `exit`

Przełączając konta użytkowników, zyskujemy dostęp do przypisanych im zasobów z ich uprawnieniami. Daje to możliwość wykonywania operacji niejako w imieniu danego użytkownika.

ZADANIA DO WYKONANIA

1. Przygotuj prezentację opisującą zasady konfiguracji **Terminala**. Wykonaj zrzuty ekranowe dla różnych ustawień. Wybierz wariant, który będzie ci najbardziej odpowiadał. Uzasadnij swój wybór w prezentacji.
2. Załóż, że jesteś administratorem komputera, przy którym będzie pracować także nowy pracownik. Wasz komputer stoi w pokoju nr 122. Telefon służbowy to 12340987, a prywatny to 987654321. Twój nowy kolega będzie odbywał staż zawodowy. Utwórz dla niego konto i nadaj mu hasło. Wykorzystaj informacje o pracowniku w celu opisanego konta. Opisz wszystkie wykonane czynności w krótkiej prezentacji. Dodaj zrzuty ekranowe.
3. Przełącz użytkowanie **Terminala** na konto użytkownika z zadania 2. Zmień hasło do tego konta. Opisz czynności w krótkim dokumencie tekstowym. Wykorzystaj zrzuty ekranowe.
- 4*. Wykonaj dokument tekstowy, w którym rozwiniesz odpowiedź **Terminala** na polecenie `su -h`. Wyjaśnij szerzej opis poszczególnych opcji.

PODSUMOWANIE LEKCJI

- s. 128 • **Powłoka systemowa** to program pełniący funkcję pośrednika pomiędzy systemem operacyjnym lub aplikacjami a użytkownikiem.
- s. 128 • **Konsola** jest powłoką systemową.
- s. 129 • **Terminal** to program emulujący Konsolę. Dzięki temu możliwa jest tekstowa komunikacja z powłoką systemu.
- s. 129 • **Terminal** w Linux można uruchomić w oknie poleceń (**Alt+F2**), wpisując jedno z poleceń **rxvt**, **xterm**, **gnome-terminal**, **konsole** lub klikając ikonę **Terminala**.
- s. 129 • Użytkownik może definiować **preferencje Terminala**, np. zmienić wielkość liter.
- s. 132 • Domyślnym kontem administratora jest **root**.
- s. 132 • Niektóre polecenia można wydawać jedynie w imieniu administratora. Służy do tego program **sudo**.
- s. 132 • Do tworzenia kont użytkowników służy polecenie **adduser**.
- s. 133 • Do zmiany konta obsługiwanego w terminalu służy polecenie **su**.
- s. 133 • Hasło do konta zmieniamy poleceniem **passwd**.



17. Okienka nie są potrzebne, czyli używamy konsoli Linux

NA TEJ LEKCJI:

- dowiesz się, jak tworzyć i usuwać katalogi, kopiować pliki i nadawać uprawnienia za pomocą konsoli Linux.

Umiesz już zarządzać użytkownikami w systemie Linux za pomocą konsoli. Do-myślasz się, że to nie wszystkie jej możliwości. Oczywiście istnieje jeszcze wiele poleceń o różnym przeznaczeniu – między innymi takie, dzięki którym można operować na katalogach i plikach oraz nadawać do nich uprawnienia.

17.1. Pliki i katalogi nie dla wszystkich, czyli uprawnienia dla różnych użytkowników

W Linux można ustalić uprawnienia do plików i katalogów indywidualnie dla każdego z użytkowników. Zaczniemy od sprawdzenia, jakie uprawnienia ma nasz użytkownik. Posłużymy się poleceniem `ls` (list), które użyte bez argumentu wyświetli jedynie listę. Dopiero po użyciu `-l` uzyskamy pełną informację o plikach lub katalogach (rys. 17.1.).

Uwaga!

Polecenie `ls` wyświetla zawartość aktualnego katalogu. W przykładzie jest to `/home/wojtek`.

```
wojtek@wojtek-VirtualBox:~$ ls -l
razem 796
drwxr-xr-x 2 wojtek wojtek 4096 wrz 26 12:12 Dokumenty
drwxrwxr-x 3 wojtek wojtek 4096 wrz 26 10:47 home
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Muzyka
drwxr-xr-x 2 wojtek wojtek 4096 sie 23 10:36 Obrazy
drwxr-xr-x 2 wojtek wojtek 4096 wrz 15 22:24 Pobrane
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Publiczny
drwxr-xr-x 2 wojtek wojtek 4096 sie 17 08:53 Pulpit
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Szablony
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Wideo
-rw-rw-r-- 1 wojtek wojtek 359223 sie 14 12:05 'Zrzut ekranu z 2020-08-14 12-05-27.png'
-rw-rw-r-- 1 wojtek wojtek 344146 sie 14 12:06 'Zrzut ekranu z 2020-08-14 12-06-43.png'
-rw-rw-r-- 1 wojtek wojtek 67515 sie 14 12:08 'Zrzut ekranu z 2020-08-14 12-08-09.png'
wojtek@wojtek-VirtualBox:~$
```

polecenie `ls`

Rys. 17.1. Przykład użycia polecenia `ls`

Jeśli chcesz ograniczyć listę wyświetlanych plików do tych o nazwie zaczynają-cej się np. od pewnej sekwencji, możesz użyć znaku `*`. Pełni on podobną funk-cję jak w przypadku poleceń w Windows. Symbol `*` zastępuje wszystkie znaki przed lub za określonym fragmentem nazwy pliku (rys. 17.2.).



```
wojtek@wojtek-VirtualBox:~$ ls -l
razem 792
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Dokumenty
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Muzyka
drwxr-xr-x 2 wojtek wojtek 4096 sie 23 10:36 Obrazy
drwxr-xr-x 2 wojtek wojtek 4096 wrz 15 22:24 Pobrane
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Publiczny
drwxr-xr-x 2 wojtek wojtek 4096 sie 17 08:53 Pulpit
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Szablony
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Wideo
-rw-rw-r-- 1 wojtek wojtek 359223 sie 14 12:05 'Zrzut ekranu z 2020-08-14 12-05-27.png'
-rw-rw-r-- 1 wojtek wojtek 344146 sie 14 12:06 'Zrzut ekranu z 2020-08-14 12-06-45.png'
-rw-rw-r-- 1 wojtek wojtek 67515 sie 14 12:08 'Zrzut ekranu z 2020-08-14 12-08-09.png'
wojtek@wojtek-VirtualBox:~$ ls Zrz* -l
-rw-rw-r-- 1 wojtek wojtek 359223 sie 14 12:05 'Zrzut ekranu z 2020-08-14 12-05-27.png'
-rw-rw-r-- 1 wojtek wojtek 344146 sie 14 12:06 'Zrzut ekranu z 2020-08-14 12-06-45.png'
-rw-rw-r-- 1 wojtek wojtek 67515 sie 14 12:08 'Zrzut ekranu z 2020-08-14 12-08-09.png'
wojtek@wojtek-VirtualBox:~$
```

Rys. 17.2. Użycie znaku * w poleceniu `ls`

użycie znaków
* oraz ?

Możliwe jest także użycie znaku `?`. Spowoduje on, że na danej pozycji w nazwie wyświetlanych plików będzie mógł wystąpić dowolny znak. Przykładem jest wyświetlanie wszystkich plików o pewnej nazwie różniących się np. dwucyfrową liczbą na końcu (rys. 17.2.). W naszym przykładzie chcemy wyświetlić właściwości plików o nazwach **dokument_01**, **dokument_02**, i **dokument_13**. Jak widać, ich nazwy różnią się dwucyfrową liczbą wpisaną na końcu. Takich plików nie ma w katalogu opisanym ścieżką dostępu `/home/wojtek` (rys. 17.1.). Załóżmy, że znajdują się w katalogu `/home/wojtek/Dokumenty`. Jak zmienić domyślny katalog dla polecenia `ls`? Służy temu polecenie `cd` (*change directory*) (rys. 17.3.).

Uwaga!

Aby dobrze zrozumieć mechanizm jego działania, przypomnij sobie, czym jest ścieżka dostępu.

zmiana katalogu

```
wojtek@wojtek-VirtualBox:~$ ls -l
razem 792
drwxr-xr-x 2 wojtek wojtek 4096 wrz 26 10:18 Dokumenty
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Muzyka
drwxr-xr-x 2 wojtek wojtek 4096 sie 23 10:36 Obrazy
drwxr-xr-x 2 wojtek wojtek 4096 wrz 15 22:24 Pobrane
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Publiczny
drwxr-xr-x 2 wojtek wojtek 4096 sie 17 08:53 Pulpit
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Szablony
drwxr-xr-x 2 wojtek wojtek 4096 sie 10 20:23 Wideo
-rw-rw-r-- 1 wojtek wojtek 359223 sie 14 12:05 'Zrzut ekranu z 2020-08-14 12-05-27.png'
-rw-rw-r-- 1 wojtek wojtek 344146 sie 14 12:06 'Zrzut ekranu z 2020-08-14 12-06-45.png'
-rw-rw-r-- 1 wojtek wojtek 67515 sie 14 12:08 'Zrzut ekranu z 2020-08-14 12-08-09.png'
wojtek@wojtek-VirtualBox:~$ cd Dokumenty
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokument ?? -l
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_01
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_02
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:04 dokument_13
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Rys. 17.3. Zmiana aktualnego katalogu i użycie znaku `?` w poleceniu `ls`



Cofnięcie się do katalogu wyższego w ścieżce dostępu można zrealizować za pomocą `cd ..` (`cd`, spacja i dwie kropki).

Skoro umiesz już wyświetlać zawartość poszczególnych katalogów, czas na ich tworzenie. Służy temu polecenie `mkdir` (*make directory*) (rys. 17.4.).

tworzenie
i usuwanie
katalogu

Zmiana aktualnego katalogu
za pomocą polecenia `cd`.

Tworzenie nowego katalogu
poleceniem `mkdir`.

```
wojtek@wojtek-VirtualBox:~$ cd Dokumenty
wojtek@wojtek-VirtualBox:~/Dokumenty$ mkdir nowy_katalog
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls -l
razem 4
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_01
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_02
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:04 dokument_13
drwxrwxr-x 2 wojtek wojtek 4096 wrz 26 10:52 nowy_katalog
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 tekst1
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:04 tekst12
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Rys. 17.4. Tworzenie katalogu za pomocą polecenia `mkdir`

Skoro można utworzyć katalog, można go także usunąć. Zrobisz to, postępując się poleceniem `rmdir` (*remove directory*). W naszym przykładzie usuniesz katalog **nowy_katalog**, wpisując polecenie `rmdir nowy_katalog`.

Za pomocą poleceń konsoli możliwe jest także kopiowanie i usuwanie plików. Utwórzmy kopię pliku w tym samym katalogu, zmieniając jednocześnie jego nazwę (rys. 17.5.):

kopiowanie
i usuwanie plików

Tworzenie kopii pliku w tym samym katalogu.

```
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokument_01 -l
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_01
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_02
wojtek@wojtek-VirtualBox:~/Dokumenty$ cp dokument_01 dokument_01_KOPIA
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokument_01* -l
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 10:03 dokument_01
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 13:14 dokument_01_KOPIA
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Rys. 17.5. Wykonanie kopii pliku za pomocą polecenia kopiowania `cp`



Do usuwania pliku służy polecenie `rm` (rys. 17.6.):

```
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls *KOPIA -l
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 13:14 dokument 01 KOPIA
wojtek@wojtek-VirtualBox:~/Dokumenty$ rm dokument_01_KOPIA
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls *KOPIA -l
ls: nie ma dostępu do '*KOPIA': Nie ma takiego pliku ani katalogu
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Rys. 17.6. Usuwanie pliku za pomocą polecenia kopiowania `rm`

Uwaga!

Rozdział nie wyczerpał wszystkich możliwości opisanych poleceń. Jeśli interesują cię inne formy zapisywania wzorców wyświetlania zawartości katalogów, ich tworzenia i usuwania, poszukaj informacji na wiarygodnych stronach internetowych lub w podręcznikach obsługi Linux.

17.2. Uprawnienia do plików, czyli nie każdy może mieć dostęp

Skoro wiesz już, jak wyświetlać zawartość katalogu, czas na wyjaśnienie znaczenia znaków (uprawnień) wyświetlanych na początku linii po użyciu polecenia `ls -l`.

oznaczenia
literowe
uprawnień

Do oznaczenia uprawnień do pliku lub katalogu używa się trzech liter:

`r` (*read*) oznacza prawa do odczytu,

`w` (*write*) oznacza prawa do zapisu,

`x` (*execute*) oznacza wykonanie (uruchomienie pliku w systemie).

Tak więc pełne prawa dla użytkownika lub grupy zostaną zapisane jako `rwX`. W przypadku wystąpienia znaku - (minus) dane prawo nie zostało przydzielone.

Kolejność występowania symboli uprawnień nie jest przypadkowa. Rozważmy przykłady:

uprawnienia

1. Wszyscy otrzymali komplet uprawnień `drwxrwxrwx`.

Dla ułatwienia rozdzielmy zapis uprawnień (rys. 17.7.).

d rwx rwx rwx

↑	↑	↑	↑
Oznacza, że uprawnienia dotyczą katalogu.	Pełne uprawnienia dla właściciela katalogu.	Pełne uprawnienia dla grupy, którą przypisano plikowi.	Pełne uprawnienia dla pozostałych użytkowników i grup.

Rys. 17.7. Uprawnienia do pliku lub katalogu



2. Plik **dokument** z rysunku 17.4. ma następujące oznaczenie uprawnień:

`-rw-rw-r--`

Rozszyfrujemy ten zapis. Minus na pierwszej pozycji z lewej informuje o tym, że nie jest to katalog (to plik). Następne trzy znaki `rw-` oznaczają, że właściciel ma uprawnienia do czytania i zapisu pliku, ale nie może go uruchamiać. To oczywiste, ponieważ nasz dokument nie jest programem. Takie same uprawnienia ma grupa, którą przypisano do pliku. Świadczą o tym kolejne trzy znaki `rw-`. Pozostali użytkownicy systemu mogą ten plik jedynie odczytywać (`r--`).

Uprawnienia z przykładu 2 zostały nadane automatycznie podczas tworzenia dokumentu. Oczywiście można je zmieniać. Służy do tego polecenie `chmod`.

argumenty
polecenia `chmod`

Argumentami polecenia `chmod` są litery odpowiadające użytkownikom:

- `u` – (*user*) właściciel,
- `g` – (*group*) grupa,
- `o` – (*others*) pozostali,
- `a` – (*all*) wszyscy.

Jeśli chcemy dodać uprawnienia, przed literą używamy znaku `+` (plus). Jeśli odbieramy uprawnienia, stawiamy `-` (minus) (rys. 17.8.):

Nadanie kompletu uprawnień do pliku **dokument** wszystkim użytkownikom.

```
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokumen? -l
-rw-rw-r-- 1 wojtek wojtek 0 wrz 26 12:12 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$ chmod a+rxw dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokumen? -l
-rwxrwxrwx 1 wojtek wojtek 0 wrz 26 12:12 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$ chmod g-rwx,o-rwx dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls dokumen? -l
-rwx----- 1 wojtek wojtek 0 wrz 26 12:12 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Pozostawienie wszystkich uprawnień do pliku **dokument** wyłącznie właścicielowi.

Rys. 17.8. Zmiana uprawnień do pliku z użyciem opcji wyrażonych literami

Możliwe jest także stosowanie zapisu uprawnień w formie liczb:

- 4 (*read*) oznacza prawa do odczytu,
- 2 (*write*) oznacza prawa do zapisu,
- 1 (*execute*) oznacza wykonanie (uruchomienie pliku w systemie).

Uprawnienia można więc zapisać według schematu:

- pełne prawa to liczba 7,
- zapis i odczyt to liczba 6,

oznaczenia
liczbowe
uprawnień



tylko odczyt to liczba 4,
brak uprawnień to liczba 0.

Dla przykładu z rysunku 17.8. pozostawienie wszystkich uprawnień wyłącznie właścicielowi będzie miało postać (rys. 17.9.):

```
chmod 700 dokument
```

Nadanie kompletu uprawnień do pliku **dokument** wszystkim użytkownikom.

```
wojtek@wojtek-VirtualBox:~/Dokumenty$ chmod 777 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls -l dokumen?
-rwxrwxrwx 1 wojtek wojtek 0 wrz 26 12:12 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$ chmod 700 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$ ls -l dokumen?
-rwx----- 1 wojtek wojtek 0 wrz 26 12:12 dokument
wojtek@wojtek-VirtualBox:~/Dokumenty$
wojtek@wojtek-VirtualBox:~/Dokumenty$
```

Pozostawienie wszystkich uprawnień do pliku **dokument** wyłącznie właścicielowi.

Rys. 17.9. Zmiana uprawnień do pliku z użyciem opcji wyrażonych liczbami

ZADANIA DO WYKONANIA

1. Użyj polecenia `ls` do wyświetlenia uprawnień dla grupy plików zawierających w nazwie literę **a**. Opisz krótko zastosowaną metodę.
2. W systemie Linux uruchom edytor tekstu. Zapisz pusty plik pod nazwą **Zadanie2**. Za pomocą konsoli zmień uprawnienia do niego w taki sposób, by tylko właściciel mógł go edytować, a pozostali użytkownicy jedynie czytać. Użyj parametrów zapisanych za pomocą liter **r**, **w**, **x**.
3. Dla pliku z zadania 2 zmień uprawnienia na takie, by wskazana grupa uzyskała możliwość jego edycji. Użyj parametrów zapisanych za pomocą liter i liczb.
- 4*. Za pomocą polecenia konsoli utwórz nowy katalog o nazwie **katalog_cwiczen**. Skopiuj do niego plik z poprzedniego zadania, zmieniając jego nazwę. Użyj odpowiedniego polecenia konsoli.

PODSUMOWANIE LEKCJI

- s. 135 • Używając polecenia `ls`, możesz w nazwie pliku zastąpić ciąg znaków **gwiazdką** (Shift 8) lub jeden znak **pytajnikiem**.
- s. 136 • Zmiana aktualnego katalogu jest wynikiem działania polecenia `cd`.
- s. 137 • Katalog można utworzyć za pomocą polecenia `mkdir`, a usunąć za pomocą polecenia `rmdir`.
- s. 138 • Uprawnienia można definiować za pomocą liter **r**, **w**, **x** lub liczb **4**, **2** i **1** i zmieniać poleceniem `chmod`.
- s. 138 • Uprawnienia można nadawać plikom i katalogom osobno dla właściciela, wskazanej grupy lub pozostałych użytkowników.



18. Komputer w kieszeni, czyli jak wykorzystać system Android w nauce i pracy

NA TEJ LEKCJI:

- poznasz krótką historię systemu Android;
- zapoznasz się z niektórymi pożytecznymi aplikacjami edukacyjnymi;
- dowiesz się, jak połączyć swój telefon z komputerem i zmusić do współpracy Androida i Windows.

Android kojarzy się nam głównie ze smartfonami i nowoczesnymi telewizorami. Aplikacje dla tego systemu operacyjnego pozwalają na prowadzenie i odbieranie transmisji strumieniowych, uruchamianie zaawansowanych gier i korzystanie z wielu usług internetowych. Coraz częściej także rozmowy nie są już prowadzone poprzez usługi telefonii operatora, a dzięki komunikatorom internetowym. Współczesny telefon stał się pełnoprawnym, sprawnym i szybkim komputerem mieszczącym się w kieszeni.

18.1. Trzeci gracz, czyli system operacyjny dla urządzeń mobilnych

Telefonia komórkowa jest z nami od dziesięcioleci. Od początku producenci prześcigali się w opracowywaniu coraz mniejszych i dłużej pracujących telefonów. Większość z nich była wyposażona w niewielki ekran wyświetlający kilka linii tekstu i klasyczną klawiaturę z 12 klawiszami. Niektóre modele oferowały także proste gry, radio UKF, notatnik, dyktafon itp. Prawdziwą rewolucję wywołało wprowadzenie kolorowych ekranów dotykowych i wydajnych procesorów. Wtedy stało się jasne, że konieczne będzie utworzenie sprawnie działającego systemu operacyjnego na wzór systemów komputerów osobistych (macOS, Windows, Linux). Telefony miały się bowiem stać osobistym komputerem mieszczącym się w kieszeni. Wielu producentów próbowało wprowadzić na rynek system operacyjny dla telefonów (Microsoft, Blackberry, Nokia). Jednak pierwszym był iOS (2007 rok), który w pełni nadawał się do zarządzania telefonem z ekranem dotykowym bez klawiatury. Inni producenci także pracowali nad systemem, czego efektem były **Android** i Windows dla urządzeń mobilnych.

Projektanci oparli jego budowę na **jądrze pochodzącym z systemu Linux**. Dzięki temu nie musieli tworzyć systemu od podstaw. Nie oznacza to jednak kompatybilności z dystrybucjami na komputery PC. Programów z PC nie można bezpośrednio uruchomić w Androidzie. Pierwsza wersja systemu została udostępniona w 2008 roku. Dzięki coraz większym możliwościom telefonów i kolejnym wersjom Android jest sprawnym, szybkim i niezawodnym systemem operacyjnym instalowanym w większości urządzeń mobilnych. Jego wersje są

Android oparty na jądrze Linux



jednak ściśle związane z konkretnymi producentami smartfonów, tabletów, smartTV, odtwarzaczy strumieniowych i innych urządzeń. Każdy bowiem dopasowuje ten system do możliwości swoich wyrobów, zmieniając powłokę graficzną, aplikacje itp. Wspólną cechą jest możliwość pobierania aplikacji z jednego sklepu, co daje praktycznie nieograniczone możliwości wykorzystania takiego sprzętu. Współczesny telefon służy już nie tylko do rozmów i pisanie SMS-ów, lecz także do większości zastosowań do niedawna zarezerwowanych dla komputerów PC. Dzięki aplikacjom i sprawnym przeglądarkom internetowym można słuchać radia internetowego, oglądać telewizję, korzystać z płatnych i darmowych transmisji strumieniowych itp. Producenci systemów operacyjnych zaczęli także współpracować na polu częściowej integracji i ułatwienia współpracy pomiędzy urządzeniami z Androidem a komputerem osobistym.

18.2. Aplikacje edukacyjne, czyli osobiste centrum wiedzy i pomocy

Wśród aplikacji systemu Android znajdziesz wiele przydatnych narzędzi. Korzystasz z nich często. Wykorzystujesz mapy, nawigację, komunikatory, aplikacje portali społecznościowych i wiele innych. Warto przyjrzeć się niektórym z nich.

aplikacja
Translator

Założmy, że znajdujesz się w lokalu, w którym wywieszono plansze z napisami w języku chińskim. Chcesz wiedzieć, co oznaczają? Dowiesz się dzięki aplikacji Translator. Wystarczy wybrać odpowiednią ikonę (aparat fotograficzny) (rys. 18.1.).



Rys. 18.1. Działanie aplikacji **Translator** z użyciem kamery telefonu

Można to traktować jak ciekawostkę, ale jeśli użyjemy **Translatora** na zagranicznej wycieczce w egzotycznym kraju, zabawa zmienia się w bardzo przydatną usługę. Jeśli wybierzesz symbol mikrofonu i zaczniesz mówić do mikrofonu telefonu, Translator to przetłumaczy, a rozmówca usłyszy tekst w swoim języku. Oczywiście to, co nam odpowie, także będzie przetłumaczone.

Zdarza się również, że przygotowując się do zajęć, trafiacie na informacje w języku, którego nie znacie. Translator pomoże wam w tłumaczeniu.

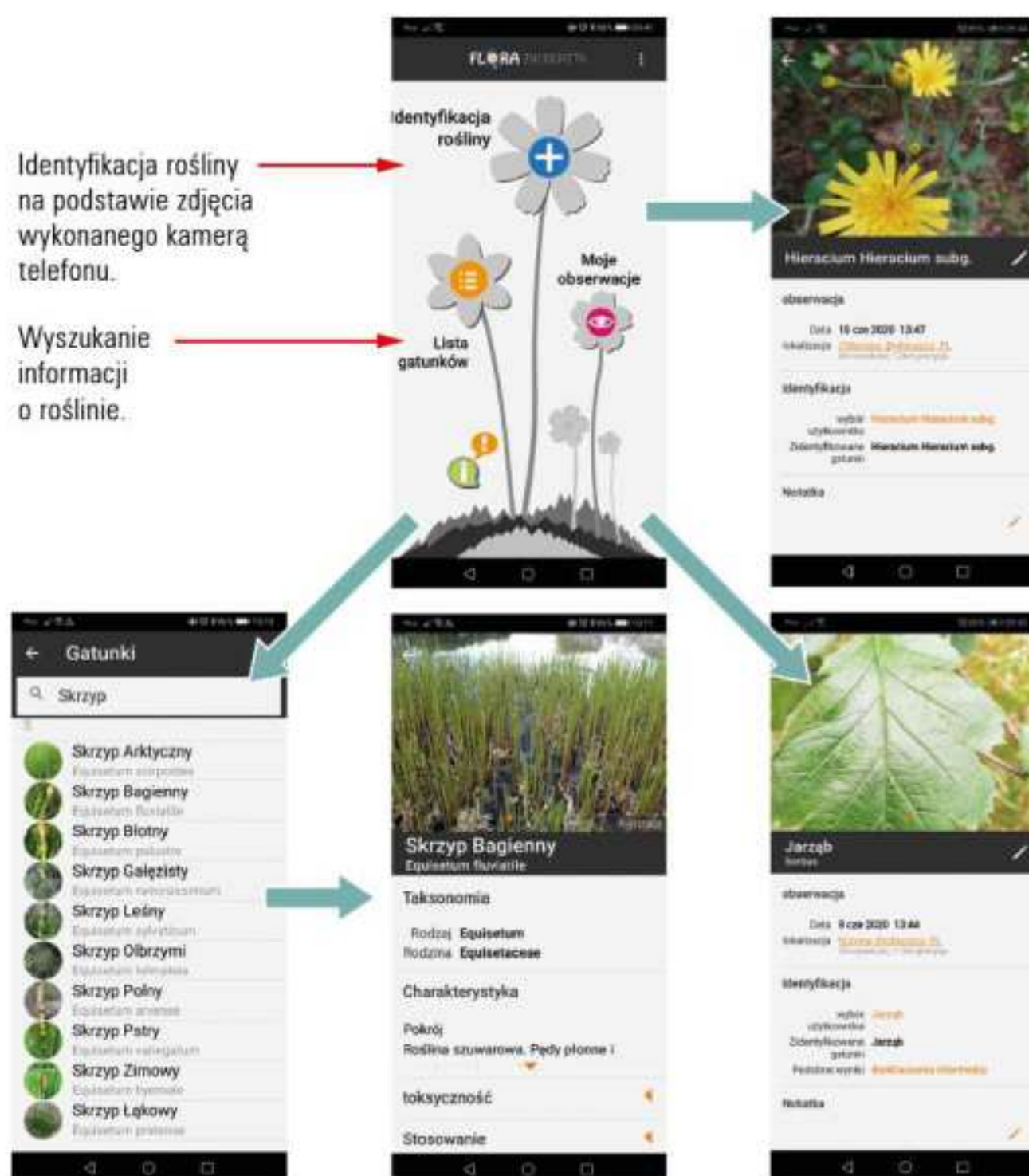


W większości telefonów zainstalowano także aplikację **Zdrowie**. Potrafi ona współpracować z opaskami kontrolującymi parametry życiowe, wagami łazienkowymi i innymi urządzeniami monitorującymi aktywność użytkownika. Dzięki temu można rejestrować, przechowywać i wizualizować wyniki takich pomiarów pochodzących z dowolnego okresu działania programu. Możliwe jest także monitorowanie snu czy ostrzeganie o przekroczeniu określonych wartości tętna. Większość dostępnych urządzeń do monitorowania organizmu, kontrolowania treningu i wysiłku itp. jest kompatybilna z aplikacją Zdrowie.

aplikacja Zdrowie

W systemie Android można także zainstalować aplikacje, które **na podstawie obrazu z kamery rozpoznają konkretne obiekty, np. rośliny**. Wśród nich są także programy niekomercyjne. Zawierają atlas roślin, w którym znajduje się wiele informacji na ich temat. Rozpoznawanie roślin na spacerze nie będzie już problemem (rys. 18.2.). Z laptopem trudno jest wędrować, a na spacerach i wycieczkach telefon z Androidem zawsze mamy przy sobie.

rozpoznawanie obrazów



Rys. 18.2. Przykład darmowego programu do identyfikacji roślin



18.3. Pełna współpraca, czyli integrujemy telefon z Androidem z komputerem osobistym

Integracja systemów w skali miasta, kraju, a często i całego świata nie jest już niczym nadzwyczajnym. W ten sposób współpracują ze sobą systemy informatyczne banków, szpitali, urzędów państwowych itp. Skoro sprawdza się to w dużej skali, to i w małej, np. w zasięgu jednego użytkownika, powinno sprawdzać się doskonale. W systemie Windows 10 do połączenia wielu funkcji z telefonem służy aplikacja **Twój telefon** (rys. 18.3.). Dzięki niej jest możliwa wymiana danych i usług pomiędzy telefonem z Androidem a komputerem PC. Warunkiem działania takiego połączenia jest wspólny dostęp do sieci wi-fi.

cechy aplikacji Twój telefon

Program **Twój telefon** sparowany z telefonem z systemem Android wyróżnia się takimi cechami, jak:

- odbieranie i wysyłanie SMS,
- prowadzenie rozmów telefonicznych,
- dostęp do zdjęć,
- zarządzanie powiadomieniami,
- wymiana plików.

Twój telefon może połączyć się z telefonami z systemami Android lub iOS.

Po połączeniu z telefonem możesz korzystać z jego funkcji w komputerze PC.



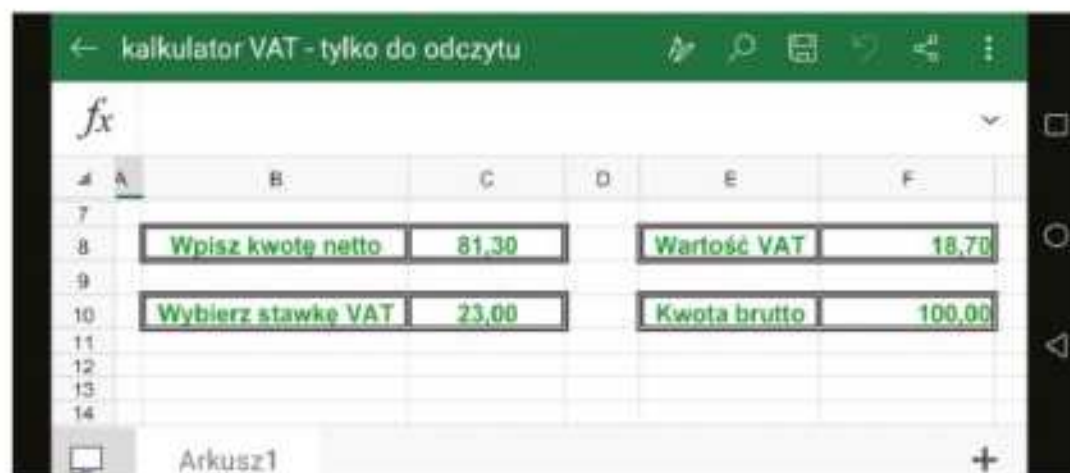
Rys. 18.3. Przykład wykorzystania aplikacji **Twój telefon** w systemie Windows 10

wymiana danych

Integracja urządzeń mobilnych z komputerami może także polegać na **wymianie i możliwości odczytu oraz edycji dokumentów**. Programy komputerowe zapisują swoje pliki w określonych, oryginalnych formatach. Stąd potrzeba tworzenia aplikacji umożliwiających korzystanie z nich także w urządzeniach z Androidem.



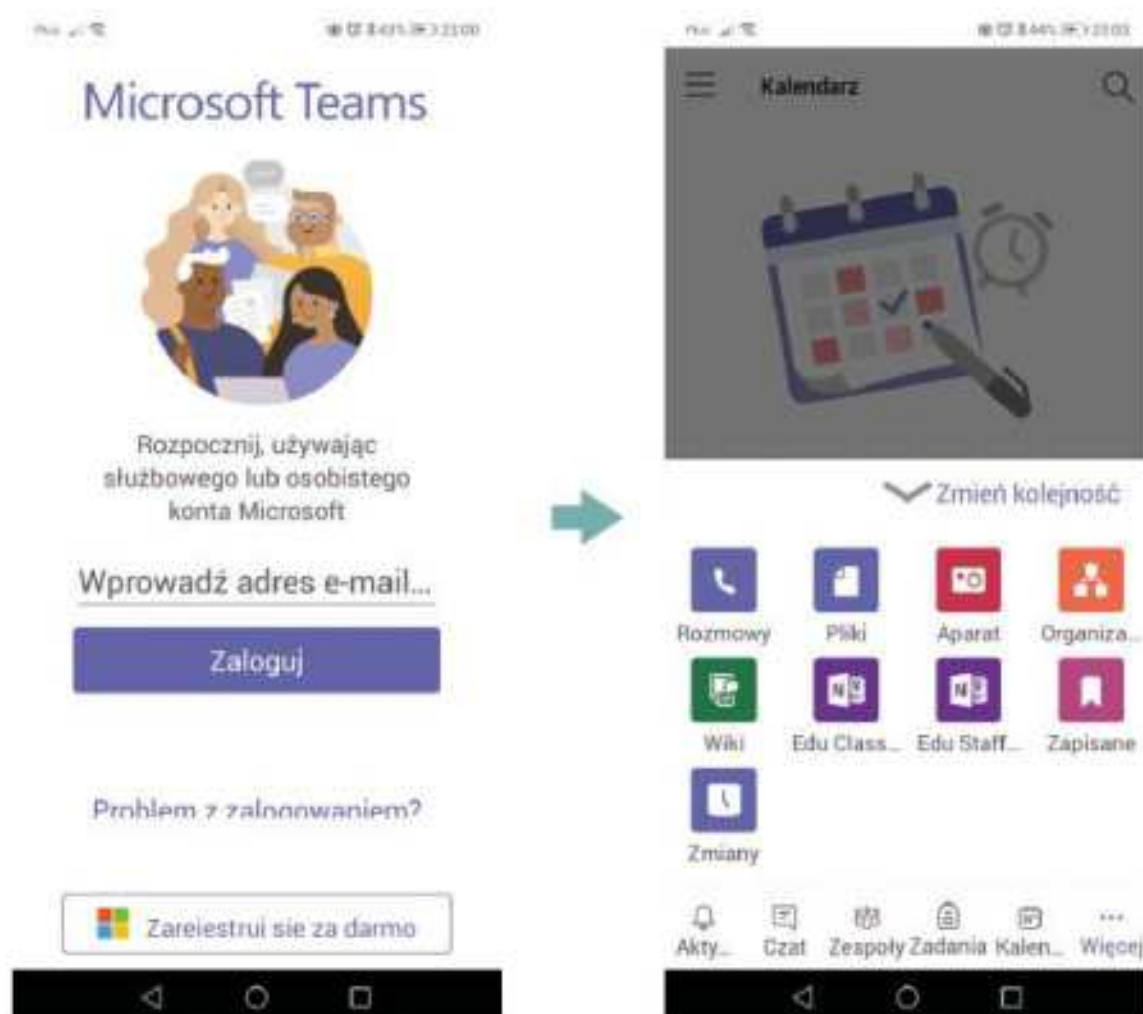
Może to być pakiet Office (rys. 18.4.), aplikacja chmury Prezi i wiele innych. Dzięki temu, że większość smartfonów potrafi wyświetlać obraz także na telewizorze lub projektorze za pomocą aplikacji zgodnej z Chromecast, można wykorzystać te aplikacje np. do prezentacji zdjęć, dokumentów itp. bezpośrednio z telefonu. Jeśli dodasz do tego integrację telefonu z chmurą, to nie będzie nawet potrzeby bezpośredniego przenoszenia plików z komputera do telefonu.



Rys. 18.4. Przykładowy arkusz Excel w telefonie z Androidem

Jedną z aplikacji do prowadzenia wirtualnych spotkań za pośrednictwem internetu jest **MS Teams**. Można jej używać on-line w chmurze Office 365 lub zainstalować ją stacjonarnie w systemie Windows. Oczywiście istnieje także aplikacja dla Androida (rys. 18.5.).

**Teams
w Androidzie**



Rys. 18.5. Aplikacja Teams w telefonie z Androidem



Ponieważ poszczególne telefony i wersje systemu Android się różnią, ich konfigurację należy przeprowadzić zgodnie z instrukcją obsługi. Podstawowe zasady są jednak takie same jak w przypadku systemu Windows. Unikanie podawania własnych danych, włączanie ochrony systemu i zasady bezpieczeństwa to podstawy bezpiecznego użytkowania systemu Android. Programy pobrane ze sklepu Play są sprawdzane przez Google, ale zdarzają się także takie, które wyłudniają dane lub wręcz szpiegują użytkownika. Jeśli instalacja programu nie jest niezbędna, lepiej z niej zrezygnować.

ZADANIA DO WYKONANIA

1. Przejrzyj aplikacje w twoim telefonie. Podziel je na grupy – rozrywka, edukacja, komunikacja, multimedia i internet. Z każdej grupy wybierz jedną, według ciebie najlepszą. Wykonaj zrzuty ekranu i prześlij je do komputera w dowolny sposób. Opisz wybrane aplikacje w dokumencie tekstowym. Uzasadnij swój wybór.
2. Znajdź opis aplikacji **FLORA Incognita** wspomnianej w rozdziale. Jeśli nadal jest aplikacją niekomercyjną, pobierz ją i wypróbuj jej działanie. Porównaj wyniki rozpoznawania rośliny z innym źródłem wiedzy, np. atlasem roślin. Sformułuj wnioski.
3. Skorzystaj ze stron Microsoftu i dowiedz się więcej na temat programu **Twój telefon**. Sprawdź, które telefony mogą z nim się łączyć, jakie są warunki i ograniczenia połączenia Windowsa i Androida. Swoje spostrzeżenia i najważniejsze informacje zapisz w notatce.
- 4*. Za pomocą polecenia konsoli utwórz nowy katalog o nazwie **katalog_cwiczen**. Skopiuj do niego plik z poprzedniego zadania i zmień jego nazwę. Użyj odpowiedniego polecenia konsoli.

PODSUMOWANIE LEKCJI

- s. 141 • System Android jest oparty na **jądrze systemu Linux**, a każdy z producentów opracowuje własne wersje nakładek graficznych, aplikacji i programów zarządzających telefonem.
- s. 142 • Aplikacja **Translator** tłumaczy na dowolny język sfotografowane napisy i wypowiedziane do mikrofonu słowa. Dzięki temu jest możliwy dialog pomiędzy ludźmi władającymi różnymi językami.
- s. 143 • Aplikacja **Zdrowie** jest programem integrującym urządzenia do kontrolowania stanu organizmu i zarządzania wysiłkiem. Wykorzystuje także własne metody pomiaru, np. liczby kroków.
- s. 143 • Dzięki wykorzystaniu kamery telefonu aplikacja **FLORA Incognita** pozwala na **rozpoznawanie roślin**, szybkie ich identyfikowanie i zdobycie o nich informacji.
- s. 144 • Aplikacja **Twój telefon** łączy system telefonu z Windows. Dzięki temu wszystkie czynności związane z obsługą telefonu (np. odbieranie rozmów, SMS-ów itp.) można wykonać na komputerze.
- s. 145 • Wiele aplikacji jest klientami usług znanych z innych systemów, np. **Teams**, który pozwala na uczestnictwo w spotkaniach internetowych, np. zdalnych lekcjach.



19. Był pierwszym z okienkami, czyli macOS i jego właściwości

NA TEJ LEKCJI:

- poznasz krótką historię systemu macOS;
- dowiesz się, jakimi cechami wyróżnia się system macOS.

System Windows instaluje się fabrycznie na komputerach typu PC. Nic nie stoi na przeszkodzie, by także komputer złożony samodzielnie wyposażyć w system zakupiony legalnie. Czy jest to możliwe z systemem macOS? Niestety nie. W tym przypadku komputer i system operacyjny są ze sobą ściśle związane. Jakże to ma wady i zalety? Może się o nich przekonać jedynie użytkownik komputerów Mac.

19.1. Kompatybilność, czyli aplikacje w aplikacjach macOS

Mimo dużych różnic dzielących systemy macOS oraz Windows możliwa jest wymiana plików pomiędzy nimi. To naturalne w świecie, w którym obieg danych i informacji musi przebiegać bezproblemowo. Aplikacje macOS mogą więc korzystać z podstawowych formatów plików popularnych w Windows, będących **standardem w informatyce**.

Zalicza się do nich formaty dokumentów tekstowych i graficznych. Oczywiście dotyczy to także plików znamienych dla popularnych aplikacji do tworzenia tekstów, prezentacji, grafiki itp.

formaty
dokumentów

Korzystający z **chmur informatycznych** również nie będą mieli problemów z wykorzystaniem aplikacji chmurowych. Większość uruchamia się w przeglądarce internetowej. Istnieje także możliwość pobrania części z nich w wersji przeznaczonej dla macOS. Większość aplikacji **wyznaczających standardy** w swojej dziedzinie można kupić w wersjach dla obu systemów.

chmury w macOS

Jeśli zakupisz subskrypcję Office 365, w której skład wchodzi popularne edytory Word, PowerPoint i arkusz Excel, możesz je zainstalować w systemie komputera Mac. Podobnie jest z aplikacjami graficznymi, np. z pakietu Adobe. Podczas zakupu aplikacji należy zwrócić uwagę na wersję, która powinna być zgodna z używaną wersją systemu macOS.

aplikacje
wyznaczające
standardy

W Windows 10 standardową przeglądarką jest **Edge**. W macOS tę funkcję pełni **Safari**. Nie oznacza to jednak, że nie można zainstalować alternatywnych aplikacji. Przykładem mogą być popularne Firefox, Opera i Chrome. Co ciekawe, w systemie Windows można zainstalować **Safari**.

przeglądarka
internetowa



Dzięki tym cechom użytkownicy PC i komputerów Mac mogą korzystać z wielu popularnych aplikacji przeznaczonych zarówno dla amatorów, jak i profesjonalistów.

Oczywiście oprócz aplikacji znanych użytkownikom Windows w macOS zainstalowano wiele innych programów. Dzięki nim można odtwarzać i tworzyć multimedia, w tym filmy, a także przygotowywać prezentacje, przetwarzać zdjęcia itp. (rys. 19.1.).



Rys. 19.1. Przykłady aplikacji systemu macOS

19.2. Bezpieczeństwo danych, czyli co cenimy w macOS

Powszechnie znana jest opinia o wysokim stopniu odporności na wirusy i złośliwe oprogramowanie, a co za tym idzie – o bezpieczeństwie danych zgromadzonych w komputerach z systemem macOS i urządzeniach przenośnych Apple.

Ta opinia nie wzięła się znikąd. Ponieważ system i komputer Mac są stworzone w całości „dla siebie”, można było odpowiedzialność za zabezpieczenia podzielić pomiędzy **sprzęt i oprogramowanie**. Takie rozwiązanie jest dużą barierą dla działań hakerów i szkodliwych programów.

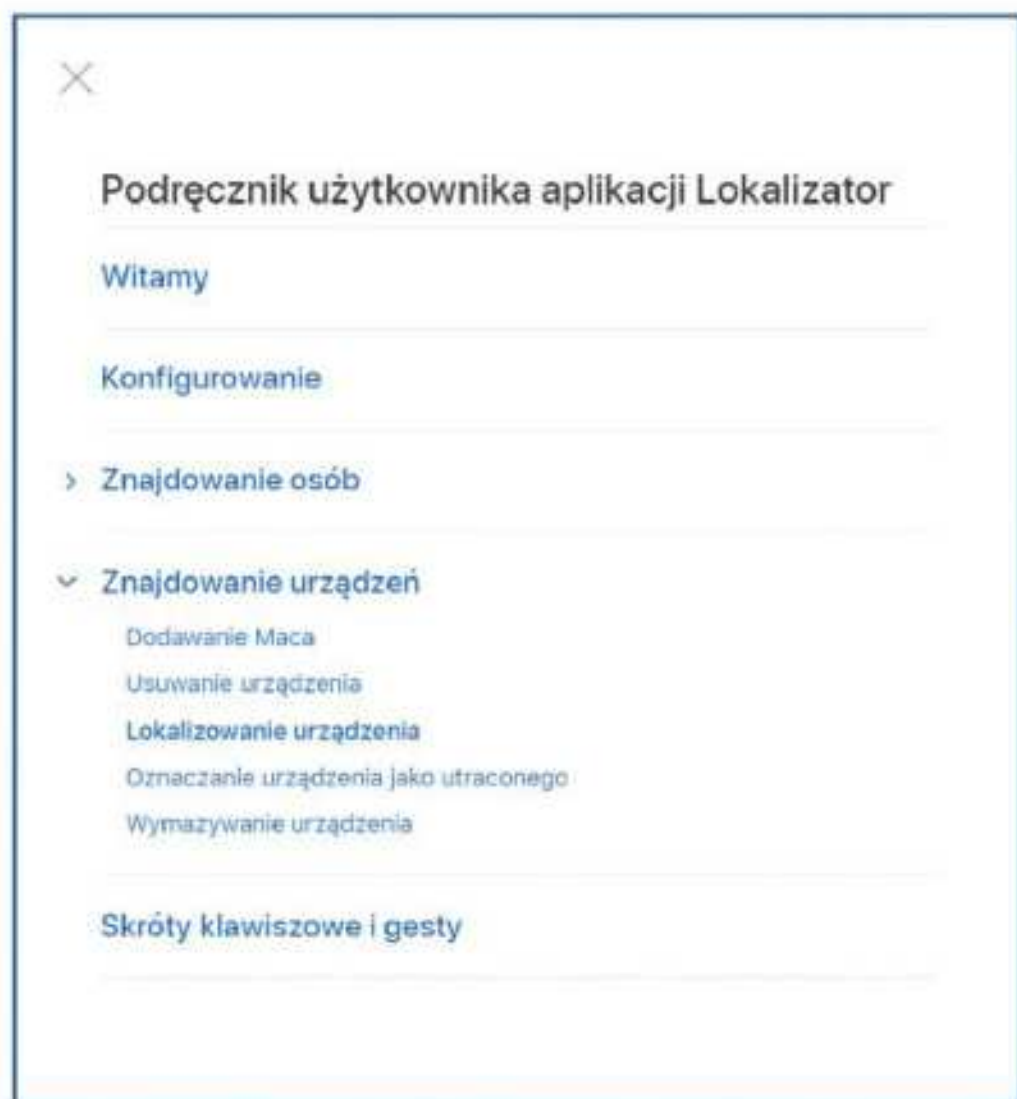
Stwarza to także możliwość szybkiej lokalizacji komputera w razie jego zagubienia lub kradzieży. Dbą o to aplikacja **Lokalizator**, która jest skuteczna nawet w przypadku uśpienia lub odłączenia komputera od sieci. Oczywiście Lokalizator, by spełnić swoje zadanie, musi być aktywny i skonfigurowany. Wówczas komputer będzie wydawał dźwięki, gdy znajdzie się w zasięgu słuchawek bezprzewodowych AirPods.

Opis konfiguracji i użytkowania **Lokalizatora**, podobnie jak innych aplikacji macOS, znajdziesz w podręczniku użytkownika (rys. 19.2.) na stronie <https://support.apple.com/>.

sprzęt i system
na straży
bezpieczeństwa

lokalizator

podręcznik
użytkownika
Lokalizatora



Rys. 19.2. Spis treści podręcznika użytkownika aplikacji **Lokalizator**

Swój udział w zapewnieniu bezpieczeństwa systemowi macOS ma także przeglądarka Safari. Jedną z jej cech jest uniemożliwianie śledzenia twoich działań w sieci, np. historii odwiedzanych przez ciebie stron.

Bardzo ważnym elementem zapewniającym bezpieczeństwo jest zastosowanie podobnego sposobu zdobywania i instalacji programów jak w znanym ci systemie Android. W macOS zainstalowano aplikację Mac App Store. Wszystkie programy oferowane w tym „sklepie” są dokładnie sprawdzone i bezpieczne. Mac App Store działa także w urządzeniach mobilnych firmy Apple.

App Store

System macOS wyposażono także w program Gatekeeper zapewniający kontrolę wszystkich programów, które użytkownik zamierza zainstalować. Kontroli tej podlegają zarówno aplikacje z Mac App Store, jak i pozostałe, np. pobrane z internetu.

Firma Apple oprócz komputerów Mac produkuje szereg urządzeń mobilnych. Wszystkie, w tym również komputery, współpracują ze sobą za pośrednictwem chmury iCloud (rys. 19.3.).

chmura iCloud

Na głównej stronie tej usługi można przeczytać krótką charakterystykę iCloud:



„Usługa iCloud jest zintegrowana z każdym urządzeniem Apple. Dzięki temu wszystkie Twoje rzeczy – między innymi zdjęcia, pliki czy notatki – są bezpieczne, aktualne i dostępne wszędzie tam, gdzie akurat jesteś. Wszystko działa automatycznie, więc o nic się nie martwisz i zajmujesz tylko tym, co lubisz. Na początek w iCloud dostajesz 5 GB miejsca na przechowywanie danych”.

Oczywiście nie jest to jedyne znane rozwiązanie. Podobne mechanizmy znajdziesz w chmurze firm Google i Microsoft.



Rys. 19.3. Urządzenia Apple współpracujące za pośrednictwem iCloud

19.3. Złącza w Mac, czyli podłączamy urządzenia zewnętrzne

Komputery Mac wyposażono w złącza do podłączenia urządzeń zewnętrznych. Podobnie jak w PC, w wielu z nich można korzystać z **USB**.

USB w Mac

Dzięki temu można podłączyć pendrive, dysk zewnętrzny, drukarkę itp. Część z tych urządzeń, podobnie jak w przypadku systemu Windows, do poprawnej pracy potrzebuje zewnętrznych sterowników dostarczonych przez producenta. Pewnym problemem może być system plików.

używanie pendrive'ów w Mac

Na szczęście komputery Mac bez problemu odczytują i zapisują pliki w pamięci z systemem **FAT32** stosowanym w pendrive'ach.

problem z NTFS

Niestety system plików stosowany w Windows, czyli **NTFS**, nie jest w pełni obsługiwany przez macOS.

Powoduje to duże problemy z używaniem dysków zewnętrznych. Oczywiście istnieje rozwiązanie tych kłopotów w postaci aplikacji. Przykładem może być **Microsoft NTFS for Mac** firmy Paragon Software. Niestety, są to programy komercyjne. Po ich zainstalowaniu możesz wykonywać wszystkie operacje na plikach i folderach na dysku NTFS. Programy tego typu są niezbędne do obsługi dysków zewnętrznych używanych wcześniej z Windows.



Dyski zewnętrzne sformatowane w systemie macOS otrzymują system plików **APFS**. Do ich odczytu w Windows też należy użyć odpowiedniego programu.

system plików
APFS

Oprócz typowych złączy USB w komputerach Mac stosuje się także gniazdo typu **USB C** (rys. 19.4.).

USB C
i **Thunderbolt**



Rys. 19.4. Wtyczka i gniazdo typu **USB C**

To niepozorne złącze jest również elementem magistrali **Thunderbolt 3** (rys. 19.5.) opracowanej przez Apple. Zawiera ona w sobie 4 protokoły transmisji USB 3, DisplayPort (cyfrowa transmisja obrazu i dźwięku), Thunderbolt 3 i PCI Express (interfejs dyskowy).



Uwaga!

Kable Thunderbolt 3 (rys. 19.5.) służą do obsługi magistrali Thunderbolt i USB 3.1. Kable USB 3.1 (USB C) nie działają prawidłowo ze złączami Thunderbolt.

Rys. 19.5. Wtyczka z symbolem magistrali Thunderbolt

Gniazda dla Thunderbolt pozwalają także na ładowanie baterii urządzeń przenośnych i laptopów Mac.

Rodzaj magistrali	Transfer danych
Thunderbolt 1	10 Gbit/s
Thunderbolt 2	20 Gbit/s
Thunderbolt 3	40 Gbit/s
USB 2.0	480 Mbit/s
USB 3.0	5 Gbit/s
USB 3.1	10 Gbit/s
USB 3.2	20 Gbit/s

Tab. 19.1. Porównanie transferu magistral USB i Thunderbolt



Uwaga!

W systemie macOS można zainstalować oprogramowanie, które pozwala na zainstalowanie Windows 10. Dokładny opis można znaleźć na stronie support.apple.com.

ZADANIA DO WYKONANIA

1. Sprawdź w internecie, czy istnieją darmowe programy pozwalające na odczyt i zapis danych na dyskach z systemem NTFS przez komputery Mac z systemem macOS.
2. Sprawdź w internecie, czy istnieją darmowe programy pozwalające na odczyt i zapis danych na dyskach z systemem APFS przez komputery PC z systemem Windows 10.
3. Przeszukaj zasoby internetowe lub czasopisma i dopasuj rodzaje gniazd do poszczególnych pozycji tabeli 19.1.
- 4*. Utwórz tabelę, w której umieścisz nazwy aplikacji zaimplementowanych fabrycznie w macOS i ich odpowiedniki funkcjonalne z Windows 10. Zaznacz te, które nie mają odpowiedników w Windows i opisz ich przeznaczenie.

PODSUMOWANIE LEKCJI

- s. 147 • System macOS i jego aplikacje obsługują **podstawowe i popularne formaty plików** z Windows.
- s. 147 • Za pośrednictwem przeglądarek w macOS można korzystać z tych samych **chmur informatycznych**, co w Windows.
- s. 147 • **Aplikacje wyznaczające standardy**, takie jak Office, Photoshop, AUTOCAD itp., są sprzedawane również w wersji dla macOS.
- s. 147 • W macOS można zainstalować **popularne przeglądarki** internetowe, np. Firefox, Opera i Chrom.
- s. 147 • Podstawową przeglądarką dla macOS jest **Safari**. Istnieje także wersja tego programu dla Windows.
- s. 148 • Zabezpieczenie przed złośliwym oprogramowaniem w komputerach Mac jest zrealizowane **sprzętowo i programowo**.
- s. 148 • Do odnajdowania zagubionego lub skradzionego sprzętu służy aplikacja **Lokalizator**.
- s. 149 • Wszystkie urządzenia Apple można włączyć do wspólnej chmury **iCloud**.
- s. 150 • Komputery Mac wyposaża się w magistrale **USB i Thunderbolt**.
- s. 151 • Dyski w macOS mają system plików **APFS**, który nie jest kompatybilny z **NTFS** (system plików Windows).
- s. 151 • Do podłączenia do komputera Mac dysku zewnętrznego z systemem plików NTFS (z Windows) niezbędny jest **odpowiedni program**.



PODSUMOWANIE PRZED SPRAWDZIANEM

13. Różne sposoby instalacji, czyli przygotowujemy miejsce dla systemu Linux

- Wersje, czyli **dystrybucje** systemu Linux są oparte na wspólnym jądrze. **Dystrybucje Linuxa** dla komputerów osobistych są rozpowszechniane **bezpłatnie** na licencji GNU. s. 111
- System Linux jest wykorzystywany także w **urządzeniach sieciowych, powszechnego użytku, mobilnych** itp. s. 111
- **System plików** to zbiór metod zapisu i przechowywania plików oraz zarządzania nimi na nośniku, np. dysku. s. 112
- **Odczytanie w Windows** plików z dysku z systemem Ext wymaga odpowiedniego sterownika. s. 112
- **Linux można zainstalować**: jako samodzielny i jedyny system operacyjny, równolegle do innego systemu z możliwością wyboru systemu po włączeniu komputera, w wirtualnej maszynie, na nośniku zewnętrznym. s. 112
- Programy do tworzenia wirtualnych maszyn wykorzystują **technologię wirtualizacji**, którą należy odblokować w UEFI lub setupie BIOS. Domyślnie może być zablokowana. s. 113
- Podczas tworzenia wirtualnej maszyny należy wziąć pod uwagę **wymagania systemu**, dla którego ją przygotowujemy. s. 114
- Jednym z popularnych programów do tworzenia wirtualnych maszyn jest **VirtualBox**. s. 115
- Nośnik zewnętrzny z systemem można utworzyć za pomocą programu **Universal USB Installer**. s. 116

14. Wiele wersji, czyli wybieramy dystrybucję Linuxa

- Linux stał się przyjazny dla niezaawansowanego użytkownika dzięki **powłoce graficznej GUI**. s. 118
- Wybór wersji danej dystrybucji zależy także od architektury procesora. Wersje **64-bitowe** nie będą poprawnie działały w komputerach z procesorem **32-bitowym**. s. 118
- Na wybór dystrybucji mają wpływ także **oczekiwania użytkownika** systemu. s. 120
- W wirtualnym komputerze można montować **wirtualne napędy**, np. optyczne. s. 120
- Do uruchomienia lub instalacji systemu w wirtualnej maszynie nie potrzebujesz nośnika z obrazem instalatora. Zastąpi go **wirtualny napęd**. s. 121
- Po zapoznaniu się z działaniem dystrybucji można wyłączyć maszynę lub **zainstalować system**. s. 121

15. Bez kosztów, czyli programy w Linux

- Dystrybucje z **GUI** zazwyczaj zawierają także **zestaw podstawowych** programów, takich jak przeglądarka internetowa, edytor tekstu, arkusz kalkulacyjny itp. s. 123
- Za pośrednictwem przeglądarek w Linux **można korzystać z chmur**, np. Google, Office 365 itp. s. 124
- W dostępnym oprogramowaniu dystrybucji Linux **znajdują się znane i popularne programy** dla Windows, np. GIMP, VLC, Blender. s. 125
- Menedżery oprogramowania z Linux działaniem **przypominają Sklep Play lub AppStore**. s. 126



16. Tryb tekstowy jest ważny, czyli poznajemy konsolę Linux

- s. 128 • **Powłoka systemowa** to program pełniący funkcję pośrednika pomiędzy systemem operacyjnym lub aplikacjami a użytkownikiem.
- s. 128 • **Konsola** jest powłoką systemową.
- s. 129 • **Terminal** to program emulujący Konsolę. Dzięki temu jest możliwa tekstowa komunikacja z powłoką systemu.
- s. 129 • Terminal w Linux można uruchomić w oknie poleceń (**Alt+F2**), wpisując jedno z poleceń **rxvt**, **xterm**, **gnome-terminal**, **konsole** lub klikając ikonę Terminala.
- s. 129 • Użytkownik może definiować **preferencje Terminala**, np. zmienić wielkość liter.
- s. 132 • Domyślnym kontem administratora jest **root**.
- s. 132 • Niektóre polecenia można wydawać jedynie w imieniu administratora. Służy do tego program **sudo**.
- s. 132 • Do tworzenia kont użytkowników służy polecenie **adduser**.
- s. 133 • Do zmiany konta obsługiwanego w terminalu służy polecenie **su**.
- s. 133 • Hasło do konta zmieniamy poleceniem **passwd**.

17. Okienka nie są potrzebne, czyli używamy konsoli Linux

- s. 135 • Używając polecenia **ls**, możesz w nazwie pliku zastąpić ciąg znaków **gwiazdką** (Shift 8) lub jeden znak **pytajnikiem**.
- s. 136 • Zmiana aktualnego katalogu jest wynikiem działania polecenia **cd**.
- s. 137 • Katalog można utworzyć za pomocą polecenia **mkdir**, a usunąć za pomocą polecenia **rmdir**.
- s. 138 • Uprawnienia można definiować za pomocą liter **r**, **w**, **x** lub liczb **4**, **2** i **1** i zmieniać poleceniem **chmod**.
- s. 138 • **Uprawnienia** można nadawać plikom i katalogom osobno dla właściciela, grupy lub pozostałych użytkowników.

18. Komputer w kieszeni, czyli jak wykorzystać system Android w nauce i pracy

- s. 141 • System Android jest oparty na **jądrze systemu Linux**, a każdy z producentów opracowuje własne wersje nakładek graficznych, aplikacji i programów zarządzających telefonem.
- s. 142 • Aplikacja **Translator** tłumaczy na dowolny język sfotografowane napisy i wypowiedziane do mikrofonu słowa. Dzięki temu możliwy jest dialog pomiędzy ludźmi władającymi różnymi językami.
- s. 143 • Aplikacja **Zdrowie** jest programem integrującym urządzenia do kontrolowania stanu organizmu i zarządzania wysiłkiem. Wykorzystuje także własne metody pomiaru, np. liczby kroków.
- s. 143 • Dzięki wykorzystaniu kamery telefonu aplikacja **FLORA Incognita** pozwala na **rozpoznawanie roślin**, szybkie ich identyfikowanie i zdobycie o nich informacji.
- s. 144 • Aplikacja **Twój telefon** łączy system telefonu z Windows. Dzięki temu wszystkie czynności związane z obsługą telefonu (np. odbieranie rozmów, SMS-ów itp.) można wykonać na komputerze.
- s. 145 • Wiele aplikacji jest klientami usług znanych z innych systemów, np. **Teams**, który pozwala na uczestnictwo w spotkaniach internetowych, np. zdalnych lekcjach.



19. Był pierwszym z okienkami, czyli macOS i jego właściwości

- System macOS i jego aplikacje obsługują **podstawowe i popularne formaty plików** z Windows. s. 147
- Za pośrednictwem przeglądarek w macOS można korzystać z tych samych **chmur informatycznych**, co w Windows. s. 147
- **Aplikacje wyznaczające standardy**, takie jak Office, Photoshop, AUTOCAD itp., są sprzedawane również w wersji dla macOS. s. 147
- W macOS można zainstalować **popularne przeglądarki** internetowe, np. Firefox, Opera i Chrom. s. 147
- Podstawową przeglądarką dla macOS jest **Safari**. Istnieje także wersja tego programu dla Windows. s. 147
- Zabezpieczenie przed złośliwym oprogramowaniem w komputerach Mac jest zrealizowane **sprzętowo i programowo**. s. 148
- Do odnajdowania zagubionego lub skradzionego sprzętu służy aplikacja **Lokalizator**. s. 148
- Wszystkie urządzenia Apple można włączyć do wspólnej chmury **iCloud**. s. 149
- Komputery Mac wyposaża się w magistrale **USB i Thunderbolt**. s. 150
- Dyski w macOS mają system plików **APFS**, który nie jest kompatybilny z **NTFS** (system plików Windows). s. 151
- Do podłączenia do komputera Mac dysku zewnętrznego z systemem plików NTFS (z Windows) niezbędny jest **odpowiedni program**. s. 151

III. Programowanie i rozwiązywanie problemów za pomocą komputera

- ▶ Przypomnij sobie, czyli wracamy do środowiska programistycznego
 - ▶ Przypomnij sobie, czyli podstawy języka programowania
 - ▶ Czy litery to liczby, czyli kod ASCII i porównanie tekstów
 - ▶ Metoda naiwna, czyli szukamy wzorca w tekście
 - ▶ Realizacja algorytmu, czyli program szuka wzorca
 - ▶ Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego
 - ▶ Zbiór Cantora, czyli najprostsze tworzenie fraktali
- ▶ Drzewo binarne, czyli hierarchiczna struktura danych
 - ▶ Trójkątny fraktal, czyli trójkąt Sierpińskiego
 - ▶ Krzywa fraktalna, czyli płatek Kocha



20. Przypomnij sobie, czyli wracamy do środowiska programistycznego

NA TEJ LEKCJI:

- przypomnisz sobie podstawowe informacje o środowisku programistycznym;
- uruchomisz środowisko programistyczne dla języka C++;
- sprawdzisz poprawność działania środowiska.

Szkolny przedmiot informatyka to nie tylko programowanie. Wiele czasu poświęca się takim zagadnieniom, jak: sieci, edytory, chmury itp. Po przerwie wracasz jednak do poznawania nowych algorytmów i ich kodowania w języku C++. Warto więc przypomnieć sobie sposoby instalacji środowiska programistycznego i posługiwania się nim.

20.1. Przygotowujemy narzędzia, czyli instalujemy IDE i kompilator języka

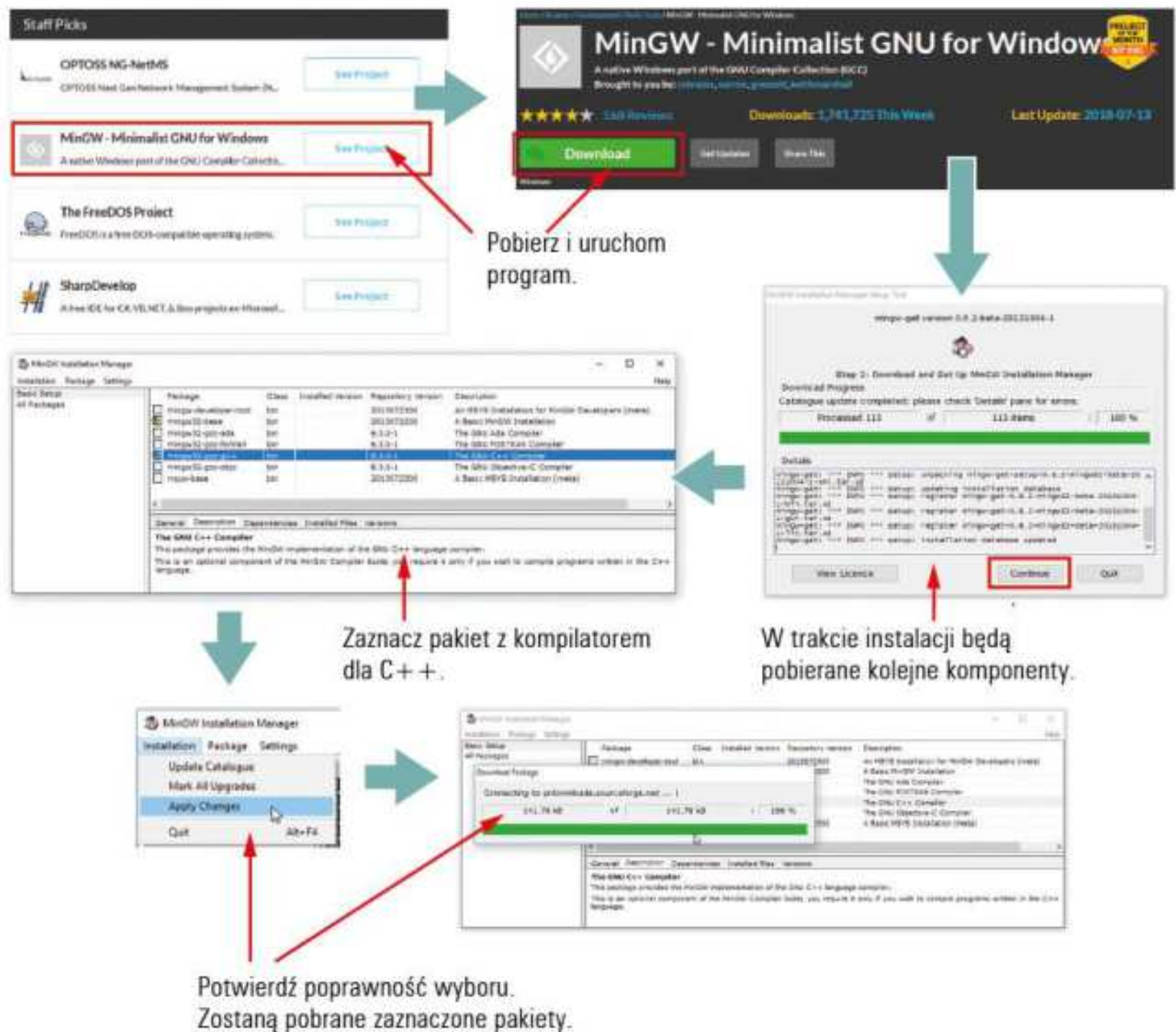
IDE (ang. *Integral Development Environment*), czyli zintegrowane środowisko programistyczne, to program lub zespół programów służących do tworzenia, testowania, modyfikowania i konserwacji programów komputerowych w danym języku. Często takie środowiska pozwalają programować w różnych językach, np. w Javie i C++. Najprostsze, graficzne, zapewne poznałeś/poznałaś, ucząc się programowania w **Scratch** lub **mBlock**. Wiele środowisk jest dostępnych bezpłatnie. Wśród nich popularne są Eclipse i Code Blocks. W poprzednich tomach do nauki programowania w języku C++ używaliście Eclipse. Dlatego skupimy się teraz na przypomnieniu podstaw jego konfiguracji i użytkowania w takiej właśnie wersji.

MinGW

Używanie Eclipse do **programowania w C++** wymaga zainstalowania dodatkowego oprogramowania, które zawiera narzędzia pozwalające na kompilację programów (kodów źródłowych) przeznaczonych dla Windows.

Powszechnie używanym darmowym środowiskiem dla C++ jest **MinGW** (*Minimalist GNU for Windows*). Szczegółowe informacje na jego temat znajdziesz w internecie.

Zanim pobierzesz i uruchomisz **Eclipse**, przeprowadź instalację MinGW (rys. 20.1.). Odszukaj w sieci instalator tego środowiska. Możesz skorzystać ze strony sourceforge.net, na której można znaleźć wiele narzędzi typu Open-Source dla programistów.



Rys. 20.1. Instalacja pakietu MinGW w systemie Windows

Po zainstalowaniu MinGW możesz już pobrać i uruchomić Eclipse dla C++.

Rozpocznij od pobrania pliku instalacyjnego ze strony <http://www.eclipse.org/downloads/packages/installer>.

Eclipse przygotowano dla trzech systemów operacyjnych: MacOS, Linux i Windows. Pobieranie zainicjujesz wyborem odpowiedniej wersji (rys. 20.2.).

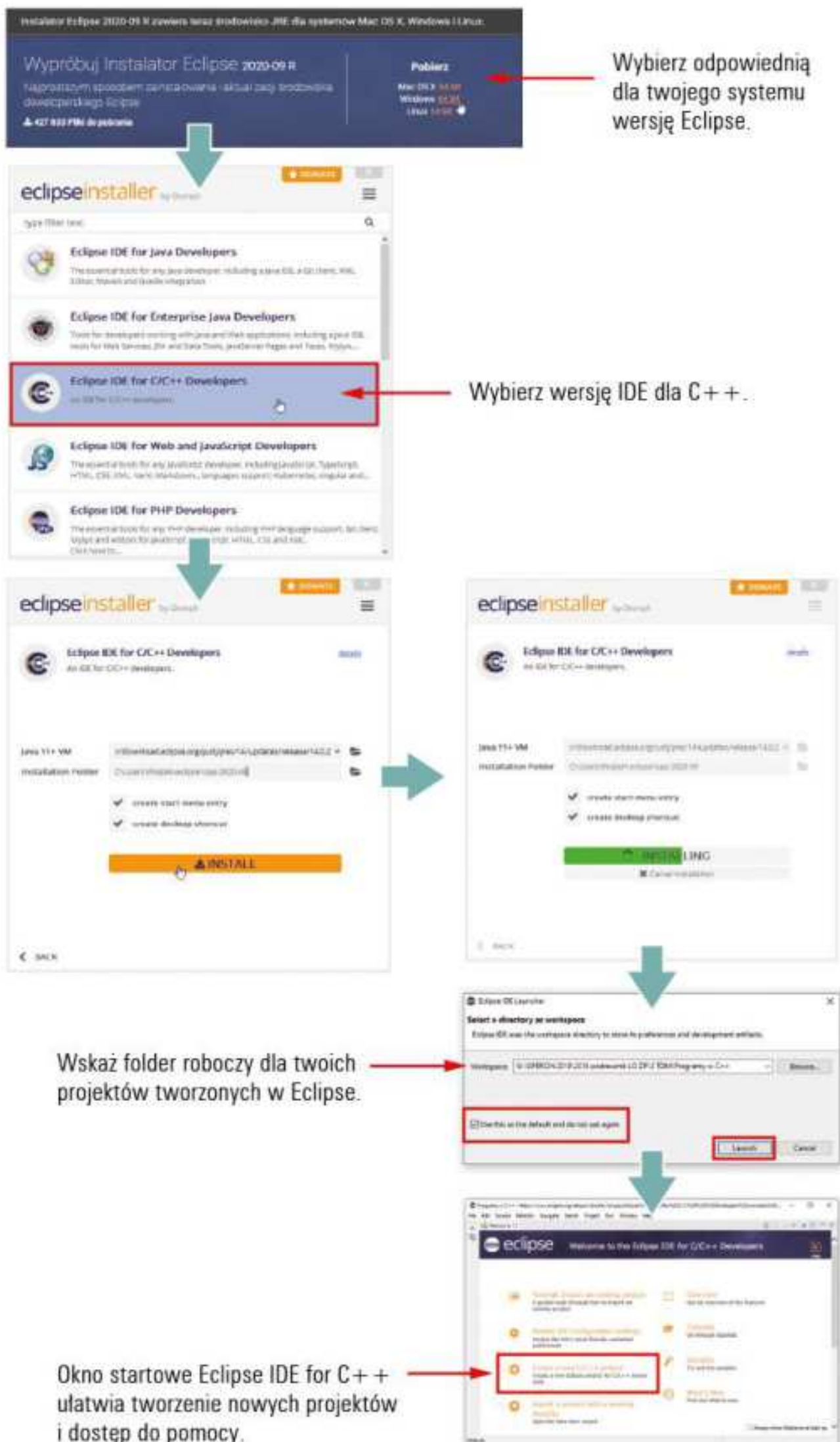
wersje Eclipse

Eclipse jest darmowy, ale pracuje nad nim zespół, który potrzebuje wsparcia. Można mu przestać dobrowolną składkę. Programiści czerpiący dochód z korzystania z IDE Eclipse powinni wspierać ten projekt.



III. PROGRAMOWANIE I ROZWIĄZYWANIE PROBLEMÓW...

instalacja Eclipse



Rys. 20.2. Proces pobierania i instalacji Eclipse dla C++



20.2. Używamy Eclipse, czyli programowanie w IDE

Eclipse ułatwia stawianie pierwszych kroków w programowaniu w C++. Sprawdź to, samodzielnie wykonując czynności z rysunku 20.3.

pierwsze kroki
w Eclipse

Z menu **Start** wybierz **New**, a następnie **C/C++ Project**.

Wybierz opcję kompilacji programu przy użyciu CDT – zestawu narzędzi pozwalających na tworzenie programów w Eclipse.

Wpisz nazwę własną projektu i wskaż na użycie kompilatora MinGW.

Przykładowy program wyświetlający pozdrowienie dla świata zostanie wczytany automatycznie.

```

1 //=====
2 // Name      : HelloWorld.cpp
3 // Author
4 // Version
5 // Copyright  : Your copyright notice
6 // Description: Hello World in C++, Ansi-style
7 //=====
8
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     cout << "!!!Hello World!!!" << endl; // prints !!!Hello World!!!
14     return 0;
15 }
16

```

Rys. 20.3. Tworzenie nowego projektu w Eclipse C++ z programem testowym

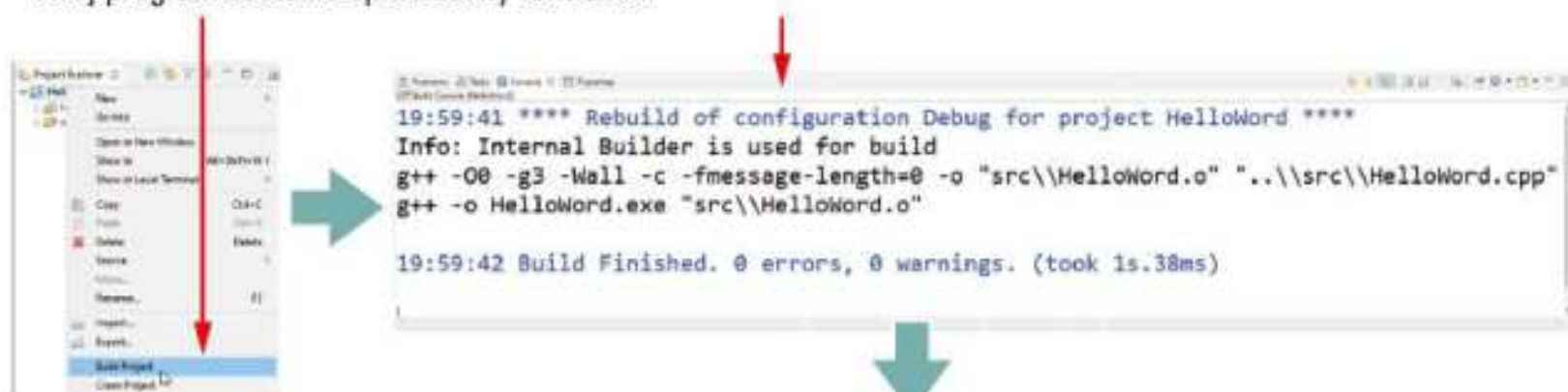


Twój projekt jest gotowy do uruchomienia. Napisz `!!!Hello World!!!` możesz zmienić na dowolny inny. Aby sprawdzić działanie programu i przetestować IDE, należy uruchomić ten program (rys. 20.4.).

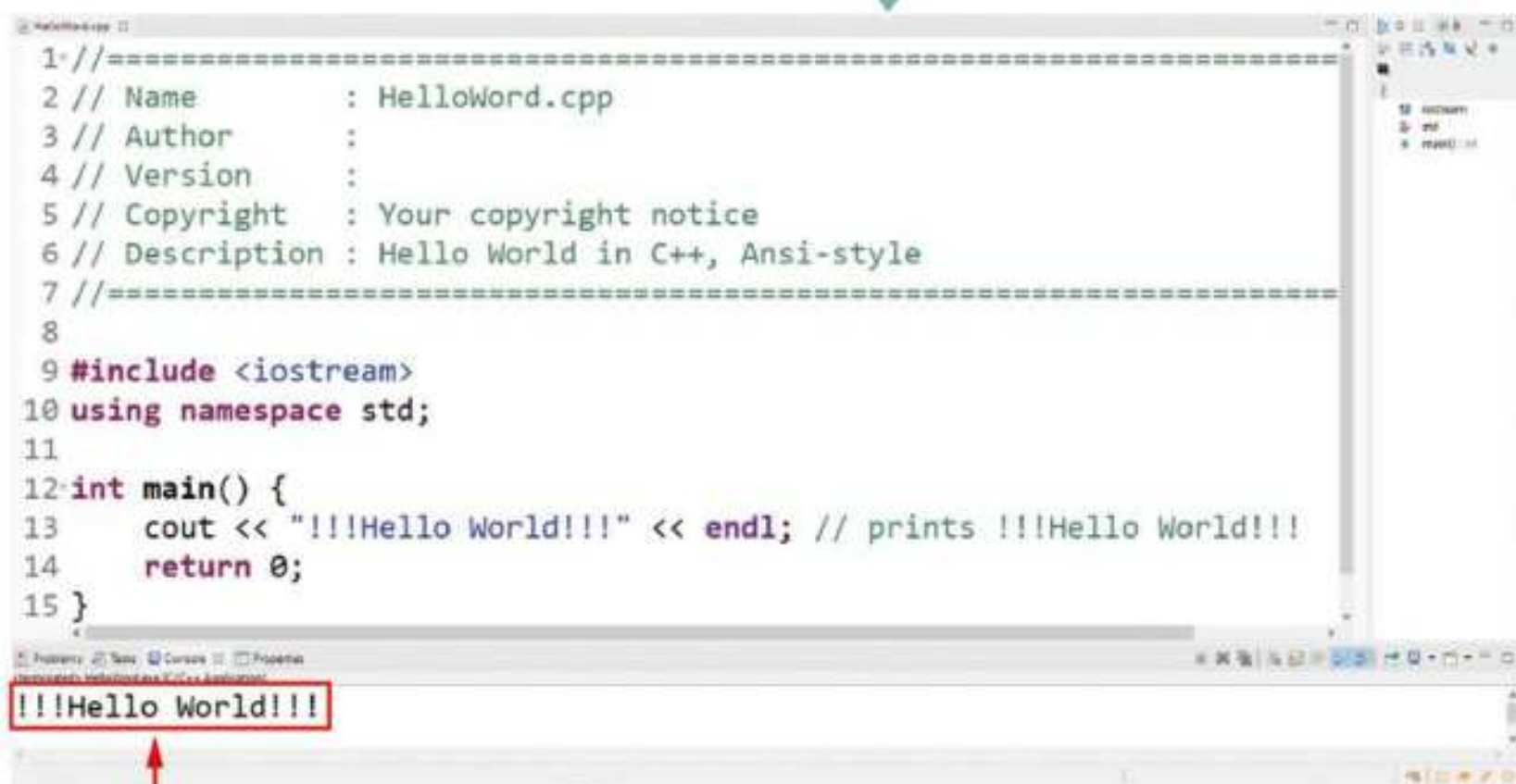
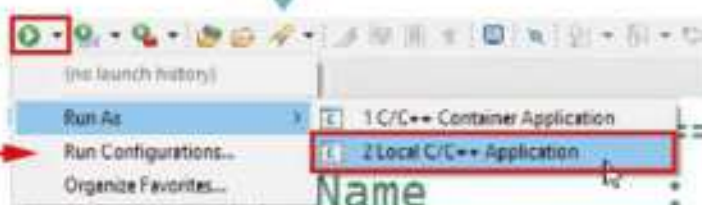
kompilacja

Twój program najpierw zostanie zweryfikowany pod kątem składni i semantyki. Następnie tekst programu będzie przetłumaczony na język maszynowy zrozumiały dla komputera. To proces kompilacji. W trakcie tego procesu zostaną dołączone moduły pozwalające na uruchomienie programu w systemie operacyjnym. Tak powstanie plik typu .EXE dla systemu Windows.

Kliknij prawym przyciskiem nazwę projektu i wybierz opcję **Build Project** (zbuduj projekt). Twój program zostanie sprawdzony formalnie.



Wybierz sposób uruchomienia programu jako aplikacji Windows.



Efekt działania programu

Rys. 20.4. Proces uruchomienia przykładowego programu w C++ w środowisku Eclipse



20.3. Podstawy podstaw, czyli struktura programu w C++

Przeanalizujmy strukturę programu Hello World (rys. 20.5.).

```
1 // =====
2 // Name      : HelloWorld.cpp
3 // Author    :
4 // Version   :
5 // Copyright : Your copyright notice
6 // Description : Hello World in C++, Ansi-style
7 // =====
8
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     cout << "!!!Hello World!!!" << endl; // prints !!!Hello World!!!
14     return 0;
15 }
```

Dobry nawyk – umieszczanie na początku programu informacji o jego przeznaczeniu, autorze, właścicielu praw itp. Tekst po `//` jest traktowany jak komentarz.

Rys. 20.5. Treść programu Hello World

`#include <iostream>`

dyrektywy

Znak `#` przed instrukcją oznacza, że jest ona dyrektywą preprocesora. Takie polecenie jest wykonywane przed kompilacją. Pozwala to na dołączenie do programu plików nagłówkowych, czyli umożliwienie mu korzystania z funkcji danej biblioteki. Po dołączeniu `iostream` będzie możliwe wykorzystanie konsoli do wyświetlania wyniku działania programu, czyli skierowanie strumienia danych do konsoli systemu (na ekran). Program uruchomi się więc w Windows, a plik uzyska rozszerzenie `.EXE`.

`using namespace std;`

przestrzeń STD

Informacja dla kompilatora, że w programie zostanie użyta standardowa przestrzeń `std`, zawierająca deklaracje dostępu do całej biblioteki standardu C++. Przy układaniu dużych programów mogą nastąpić konflikty nazw, czyli różne obiekty mogą być nazwane tak samo. Można je umieszczać w różnych przestrzeniach i dzięki temu nie będą ze sobą kolidowały. Można to wytłumaczyć na przykładzie uczniów o tym samym nazwisku i imieniu. Jeśli należałoby do różnych klas, to ich przestrzenie są oddzielne i ich nazwiska nie zostaną pomyłone. W przestrzeni szkoły będą kolidowały i mogą prowadzić do pomyłek.

`int main()`

funkcja main

Program zaczyna działanie od kodu zawartego w funkcji `main()`. Słowo kluczowe `int` oznacza, że system operacyjny będzie informowany o całkowitym (skrót `int` – integer) wyniku działania programu (funkcja zwróci wartość). Najczęściej nie jest to konieczne i w takim przypadku można użyć słowa kluczowego `void` (funkcja, która nie zwraca żadnej wartości).

Na końcu ciała funkcji z nagłówkiem `int` należy dopisać `return 0;`. Będzie to informacja dla systemu o zakończeniu wszystkich operacji (przekazanie wartości 0).

`cout << "!!!Hello World!!!" << endl;`

wyprowadź komunikat



To zasadnicza część programu, która powoduje wyświetlenie napisu. Ciąg znaków komunikatu należy umieścić w cudzysłowie.

`cout<< należy rozumieć jako „wyprowadź komunikat”.`

Uwaga!

Słowo kluczowe to słowo, które w danym języku programowania ma już przypisane znaczenie.

Uwaga!

Długie komentarze umieszcza się pomiędzy `/*` a `*/`.

Skoro uruchomiłeś/uruchomiłaś IDE, możesz rozpocząć programowanie.

ZADANIA DO WYKONANIA

1. Zmodyfikuj program opisany w rozdziale tak, by wyświetlał na ekranie inny tekst. Ustal doświadczalnie, jak wprowadzić poprawki, by tekst zajmował kilka linii. Opisz (przedyskutuj) swoje rozwiązanie.
2. Po zainstalowaniu Eclipse dla C++ wykonaj spolszczenie tego środowiska. Opisz (przedyskutuj), w jaki sposób można przywrócić wersję z językiem angielskim.
3. Ułóż algorytm opisujący czynności, które należy wykonać od momentu rozpoczęcia nowego projektu aż do uruchomienia dowolnego programu w Eclipse dla C++.
- 4*. Uzupełnij algorytm z zadania 3 opisem poprawnej struktury programu ułożonego w C++.

PODSUMOWANIE LEKCJI

- s. 158 • Środowisko programistyczne Eclipse może być wykorzystane do programowania w języku C++. Należy jednak pobrać i zainstalować **odpowiednią wersję, uwzględniając także system operacyjny**, w którym ma być zainstalowana.
- s. 158 • Do kompilacji C++ w Eclipse są potrzebne narzędzia pozwalające na kompilację programów w środowisku Windows. Jednym z powszechnie stosowanych jest **MinGW**. Należy go zainstalować przed instalacją Eclipse dla C++.
- s. 160 • Eclipse najlepiej pobrać ze strony domowej. Instalacja przebiega automatycznie. Należy jednak pamiętać o wybraniu odpowiedniej wersji (dla C++) i ścieżki do folderu, w którym będą gromadzone projekty.
- s. 161 • **Eclipse** ułatwia stawianie pierwszych kroków w programowaniu w C++ między innymi poprzez przykładowe programy pokazujące poprawną strukturę. Dzięki temu można ćwiczyć kompilację i uruchamianie programów.
- s. 163 • Dyrektywy oznaczone `#` są wykonywane przed kompilacją i powodują dołączenie odpowiednich bibliotek (zbiorów funkcji).
- s. 163 • **`using namespace std;`** to informacja dla kompilatora, że w programie zostanie użyta standardowa przestrzeń **`std`**. Przy układaniu dużych programów mogą nastąpić konflikty nazw, czyli różne obiekty mogą być nazwane tak samo. Można je umieszczać w różnych przestrzeniach i dzięki temu nie będą ze sobą kolidowały.
- s. 163 • Program zaczyna działanie od kodu zawartego w funkcji **`main()`**.
- s. 163 • **`cout<<`** należy rozumieć jako „wyprowadź komunikat”.



21. Przypomnij sobie, czyli podstawy języka programowania

NA TEJ LEKCJI:

- przypomnisz sobie podstawowe informacje o języku C++;
- utrwalisz podstawowe informacje o budowie programu w języku C++;
- przypomnisz sobie działanie pętli i instrukcji warunkowych oraz znaczenie funkcji w języku C++.

Wiesz już, że środowisko programistyczne Eclipse możesz skonfigurować dla wielu języków programowania. Umiesz to zrobić dla C++, ale nie będzie dla ciebie problemem przygotowanie go dla innych. Zanim zaczniesz układać kolejne programy, warto przypomnieć sobie podstawy C++.

21.1. Porządek musi być, czyli podstawowa struktura programu w C++

Układając program w języku C++, programista tworzy **kod źródłowy** składający się z wyrażen zrozumiałych dla człowieka.

Komputer jednak takiej postaci programu nie jest w stanie wykonać, dlatego kod jest poddawany kompilacji, czyli tłumaczeniu na język maszynowy i scalaniu z modułami umożliwiającymi uruchomienie go w danym systemie operacyjnym.

kompilacja

Język, w którym mówimy, to wyrazy połączone w zdania z zachowaniem zasad gramatyki i ortografii. Podobnie jest z językiem programowania. Składa się on z poleceń, instrukcji (wyboru i warunkowych), nazw zmiennych, operatorów oraz bibliotek i funk-

cji. Przykładowy program spełniający te warunki przedstawiono na rysunku 21.1.

Rys. 21.1. Przykładowa struktura programu w języku C++

```
1 //=====
2 // Name      : string.cpp
3 // Author    : Wojtek
4 // Version   : 1.0
5 // Copyright : Wojtek
6 // Description : struktura programu w C++
7 //=====
8
9 #include <iostream>
10 using namespace std;
11 string komunikat="jak się masz?";
12 int main() {
13
14     cout << komunikat << endl;
15     return 0;
16 }
```

Przypomnij sobie – linia programu zaczynająca się // jest traktowana jako komentarz.

#include dołącza bibliotekę o nazwie podanej pomiędzy < >, using namespace ustala przestrzeń nazw.

Funkcja main() tworzy blok główny programu. Wszystko, co znajduje się pomiędzy { }, będzie wykonywane aż do napotkania instrukcji return 0;.



`using namespace` informuje, że będzie używana dana przestrzeń nazw. Dla standardowych bibliotek C++ jest to `std`. Dzięki temu możesz napisać `cout <<"jak się masz" << endl;` zamiast `std::cout <<"jak się masz" << endl;`

21.2. Operatory, czyli działania i relacje

Warto przypomnieć sobie, czym są operatory.

Dzięki nim możliwe jest zapisanie działań matematycznych (tab. 21.1.) i relacji (tab. 21.2.).

Działanie	Operator	Przykład
Dodawanie	+	$y = x + 1$
Odejmowanie	-	$y = x - 4$
Mnożenie	*	$y = x * z$
Dzielenie	/	$y = (x + 25) / z$
Reszta z dzielenia	%	$y = z \% 2$

Tab. 21.1. Operatory działań matematycznych

Opis	Operator
Mniejszy	<
Większy	>
Równy	==
Różny	!=
Mniejszy lub równy	<=
Większy lub równy	>=

Tab. 21.2. Operatory relacyjne w języku C++

Na przykład, jeśli napiszesz `y = (x - 4) % 7`, będzie to oznaczało, że `y` będzie równe reszcie z dzielenia `(x - 4)` przez 7.

21.3. Zmienne, czyli przechowywanie danych

Definiując zmienną, nadajemy jej nazwę. Jeśli zdefiniujesz zmienną `komunikat="wprowadź pierwszą liczbę"`, oznaczać to będzie, że pod nazwą `komunikat` kryje się tekst ujęty w cudzysłów. Jeśli danego tekstu będziesz używać wielokrotnie, możesz zapisać go w zmiennej. W wielu programach i urządzeniach ten sam komunikat wyświetla się w różnych sytuacjach.



```
//deklaracja zmiennej przechowującej tekst
string komunikat;
//nadanie wartości zmiennej
komunikat = "Dziękujemy i zapraszamy ponownie";
//wyświetlenie tekstu zawartego w zmiennej
cout << komunikat;
```

deklaracja
zmiennej

Można także w jednej linii jednocześnie zadeklarować zmienną i nadać jej wartość początkową.

```
string komunikat="Dziękujemy i zapraszamy ponownie";
```

nadawanie
wartości zmiennej
lub stałej

Stałe różnią się od zmiennych tym, że przypisana im wartość nie może zostać zmieniona w żadnym miejscu programu.

Uwaga!

Deklarując zmienną lub stałą, należy ustalić jej typ.

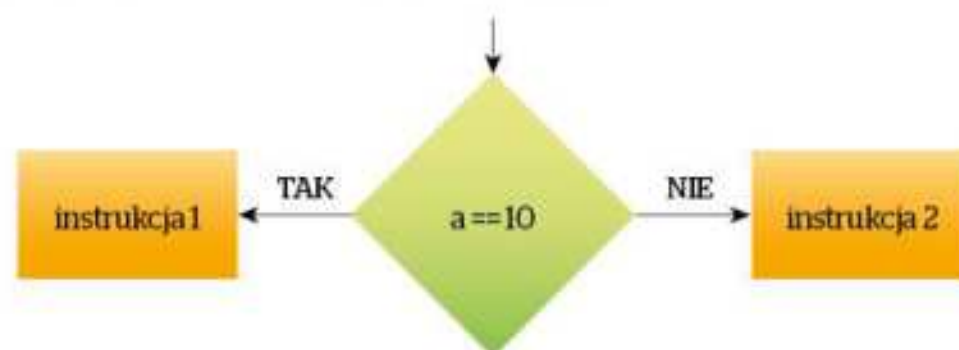
Nazwa typu	Liczba bajtów	Zakres wartości
bool	1	false lub true (prawda lub fałsz)
char	1	od -128 do 127
unsigned char	1	od 0 do 255
wchar_t	2	od 0 do 65 535
short	2	od -32 768 do 32 767
unsigned short	2	od 0 do 65 535
int	4	od -2 147 483 648 do 2 147 483 647
unsigned int	4	od 0 do 4 294 967 295
long	4	od -2 147 483 648 do 2 147 483 647
unsigned long	4	od 0 do 4 294 967 295
long long	8	od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807
unsigned long long	8	od 0 do 18 446 744 073 709 551 615
float	4	3.4E +/- 38 (7 cyfr)
double	8	1.7E +/- 308 (15 cyfr)
long double	8	1.7E +/- 308 (15 cyfr)

Tab. 21.3. Typy danych dla zmiennych lub stałych



21.4. Instrukcje sterujące, czyli realizacja bloków warunkowych algorytmu

W każdym algorytmie, który wymaga rozgałęzień, stosuje się bloki decyzyjne (rys. 21.2.). W przykładzie, jeśli warunek jest spełniony, będzie realizowana instrukcja 1, a jeśli nie – instrukcja 2.



Rys. 21.2. Przykład bloku decyzyjnego

Do zapisania warunków w języku C++ używa się operatorów logicznych i symboli porównawczych (tab. 21.4.).

Operator	Symbol porównawczy	Działanie
&&		Koniunkcja (I)
		Alternatywa (LUB)
!		Negacja (NIE)
	==	Równy
	!=	Różny
	>	Większy
	<	Mniejszy
	>=	Większy lub równy
	<=	Mniejszy lub równy

Tab. 21.4. Operatory logiczne i symbole porównawcze w języku C++

Dzięki blokowi decyzyjnemu możemy więc rozgałęziać algorytmy i w zależności od spełnienia lub niespełnienia warunku podejmować odpowiednie działania.

```
//Postać pierwsza instrukcji warunkowej
if (warunek)
{ instrukcja 1; // jeśli warunek będzie spełniony,
  instrukcja 2; // zostaną wykonane instrukcje w { },
  }           // a jeśli nie, zostaną one pominięte
//Postać druga instrukcji warunkowej
```



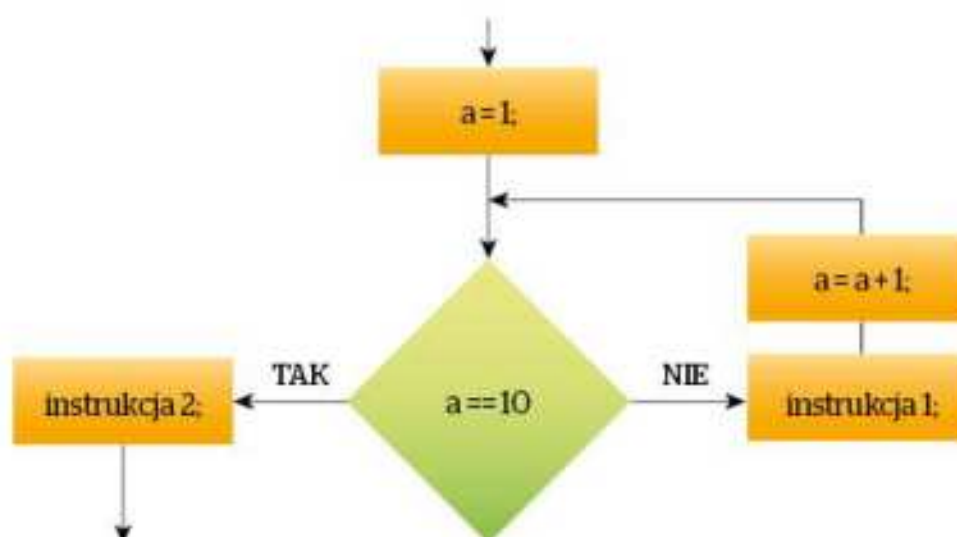
```

if (warunek)
{ instrukcja 1; //Jeśli warunek będzie spełniony,
  instrukcja 2; //zostaną wykonane instrukcje 1 i 2,
}          //a pominięte instrukcje 3 i 4
else
{instrukcja 3; //W przeciwnym przypadku zostaną
  instrukcja 4; //pominięte instrukcje 1 i 2, a wykonane
}          //3 i 4

```

Rys. 21.3. Instrukcje warunkowe w C++

Jeśli algorytm zawiera zapętlenie (rys. 21.4.), które jest realizowane, można użyć **instrukcji pętli**, dopóki spełniony jest warunek.



Rys. 21.4. Przykład pętli w algorytmie

Poznaj dwie postaci pętli stosowane w języku C++.

Pierwsza z nich (rys. 21.5.) jest wykonywana, dopóki warunek nie przestanie być prawdziwy. Realizuje ona fragment algorytmu z rysunku 21.4.

```

while (a!=10)
{
  instrukcja 1;
  a=a+1;    //Instrukcje w { } wykonywane dopóki jest spełniony warunek
}
instrukcja 2;

```

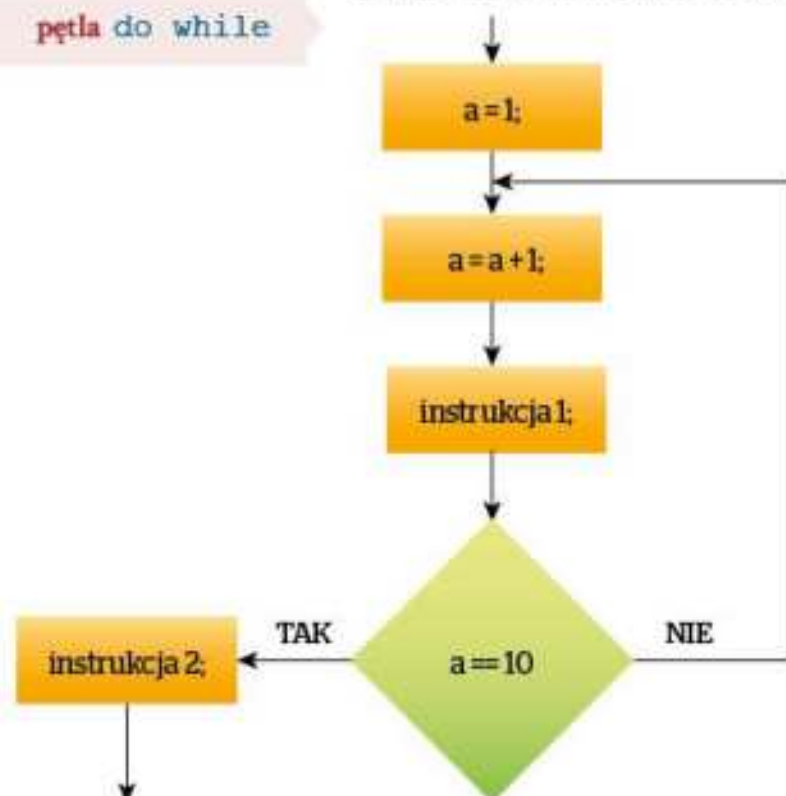
Rys. 21.5. Przykład pętli while w języku C++

W drugim rodzaju pętli (rys. 21.6.) warunek jest sprawdzany po wykonaniu wszystkich instrukcji, które ona obejmuje. Oznacza to, że wszystkie polecenia



pętla do while

i instrukcje pętli zostaną wykonane przynajmniej jeden raz i powtarzane tak długo, aż warunek przestanie być spełniany.



```

do
{a=a+1;
instrukcja 1
/*instrukcje w { } wykonane są
przynajmniej
1 raz
bez względu na warunek, a następnie
dopóki
warunek jest spełniony */
}
while (a!=10);
instrukcja 2;
  
```

Rys. 21.6. Przykład pętli do while w języku C++

Fragment algorytmu z rysunku 21.4. można zrealizować także za pomocą pętli `for` (rys. 21.7.).

```

for (a=1; a<10; a++) /*Począwszy od a=1, dopóki a<10, wykonuj
instrukcje w { } za każdym razem
zwiększając a o 1*/
{
instrukcja 1;
}
instrukcja 2;
  
```

Rys. 21.7. Instrukcja sterująca `for` w języku C++

Zauważ, że instrukcja z algorytmu `i=i+1;` jest realizowana jako parametr (`i++`) instrukcji sterującej `for`. Jaka jest więc różnica między konstrukcjami `while` i `for`? Pętla `for` jest wykonywana pewną ustaloną liczbę razy. W przykładzie dla `a` od 1 do 10, czyli 9 razy. W przypadku pętli `while` lub `do while` liczba wykonywanych pętli może być różna w zależności od momentu spełnienia warunku. Pętle mogą się zagnieżdżać, czyli występować wewnątrz innych pętli.

21.5. Programy pod nazwą, czyli funkcje w C++

funkcja

Funkcja jest to fragment programu, który jest zdefiniowany poza funkcją główną, a wywołany w niej najczęściej poprzez podanie jego nazwy oraz ewentualnych argumentów (w nawiasie za nazwą).

Jeśli funkcja nie ma zwracać żadnej wartości, czyli po prostu wykonać swoje zadanie i skończyć działanie, definiujemy ją jako funkcję typu `void` (rys. 21.8.).



Uwaga!

W nazwach zmiennych, stałych i funkcji nie wolno używać białych znaków, a duże i małe litery są rozróżniane.

```
void nazwa_funkcji()
{
    //ciało funkcji
}
```

Rys. 21.8. Deklaracja funkcji niezwracającej żadnej wartości i bez argumentów wejściowych

Jeśli funkcja ma zwracać wartość, np. liczbę całkowitą, definiujemy typ jej wartości (nie danych) i nazwę (rys. 21.9.).

```
int nazwa_funkcji()
{
    //ciało funkcji
}
```

Rys. 21.9. Deklaracja funkcji zwracającej wartość w postaci liczby całkowitej, bez argumentów wejściowych

Jeśli dana funkcja ma zostać wykonana, wystarczy w programie zapisać jej nazwę.

ZADANIA DO WYKONANIA

1. Znajdź w literaturze lub na zaufanych stronach internetowych informacje na temat zagnieżdżonych instrukcji warunkowych. Opisz na przykładzie istotę takiej konstrukcji.
2. Opracuj i zapisz w postaci graficznej przykład algorytmu, który może zostać zrealizowany za pomocą zagnieżdżonych instrukcji warunkowych.
3. Podaj przykłady problemów informatycznych, które można rozwiązać za pomocą pętli `while` i `do while`.
- 4*. Ułóż funkcję obliczającą objętość kuli.

PODSUMOWANIE LEKCJI

- **Kompilacja** polega na dołączeniu modułów umożliwiających uruchomienie programu w systemie operacyjnym i utworzeniu pliku z kodem maszynowym zrozumiałym dla procesora.
- **Operatory relacyjne** są używane w instrukcji warunkowej lub pętlach.
- W pętli **do while** instrukcje są wykonywane **przynajmniej jeden raz** bez względu na prawdziwość warunku.
- **Funkcja** jest fragmentem programu, który nazwano i który jest wykonywany poprzez podanie jego nazwy oraz ewentualnych argumentów (w nawiasie za nazwą).

s. 165

s. 167

s. 170

s. 170



22. Czy litery to liczby, czyli kod ASCII i porównanie tekstów

NA TEJ LEKCJI:

- poznasz kod ASCII;
- dowiesz się, jak wykorzystać kody ASCII w edycji tekstu;
- przeanalizujesz metody porównywania tekstów.

Odkąd uruchomiono w internecie sprawne i efektywne wyszukiwarki, zagadnienia związane z operacjami na tekstach stały się jednym z ważniejszych zadań informatyki. Czym byłoby korzystanie z internetu, gdybyśmy zostali pozbawieni możliwości wyszukiwania informacji? Algorytmy współczesnych wyszukiwarek są bardzo skomplikowane i realizują także wiele innych funkcji. Wyszukiwanie to nie tylko poszukiwanie wzorca, lecz także operowanie na zbiorach danych. W tym rozdziale znajdziesz informacje, które w przyszłości posłużą ci do budowania algorytmów i układania programów operujących na tekstach.



22.1. Kody ASCII, czyli litera dla komputera to liczba

Co może oznaczać następujący ciąg cyfr?

6 33 4 2 22 2 5 8

Z pozoru to zadanie wydaje się bardzo trudne. Nie ma bowiem żadnych wskazówek ani skojarzeń. Spróbujmy przeanalizować ten ciąg znaków.

1. Grupy cyfr są oddzielone spacją i nie są jednakowo liczebne.
2. W każdej grupie występują takie same cyfry.
3. Jest 8 grup cyfr.

Nie trzeba być Sherlockiem Holmesem (słynny detektyw i mistrz dedukcji), żeby zauważyć te cechy. Ale jak je połączyć i rozwiązać problem?

Może to szyfr? A skoro szyfr, to zapewne jest to zamieniony na cyfry napis składający się z liter. Jeśli tak, to zapewne jest ich 8, czyli tyle, ile grup cyfr. Do rozwiązania tego problemu może przyczynić się przypadek lub twoja dobra pamięć.

Podpowiedź:

Czy na co dzień posługujesz się pilotem do TV lub telefonem stacjonarnym albo pamiętasz, jak pisało się SMS-y na starych telefonach komórkowych (rys. 22.1.)?

Zwróć uwagę na klawiaturę. Teraz jest już chyba wszystko jasne, prawda? Projektanci takiego rozwiązania wprowadzania tekstu do urządzeń komputerowych

Rys. 22.1. Stary telefon komórkowy i pilot zdalnego sterowania telewizorem



(takich jak telefon komórkowy i telewizor) rozwiązyali problem dużej liczby klawiszy. Czy teraz już możesz odczytać wyraz ukryty pod ciągiem cyfr?

6 33 4 2 22 2 5 8

Uzyskany wyraz to m e g a b a j t?

Zatem możliwe jest przedstawienie liter za pomocą liczb. Potrzebny jest do tego pewien kod lub klucz szyfru. W tym przypadku miałeś/miałaś do czynienia z kodem, który zamieniał przyciskanie jednego z 10 klawiszy z cyframi w celu uzyskania litery na ekranie. Taki kod nie przyda się do komputerowego przetwarzania tekstów, ale wytycza pewien kierunek (rys. 22.2.).



Rys. 22.2. Schemat komputerowego przetwarzania tekstu

Skoro wiesz już, jaką metodą należy się posłużyć, by tekst był zrozumiały dla komputera operującego w rzeczywistości wyłącznie na liczbach binarnych, możesz poznać kod, który mu to umożliwia.

Uwaga!

ASCII (skrót od *American Standard Code for Information Interchange*) to kod, w którym kolejnym liczbom przyporządkowano litery, cyfry, znaki przestankowe i symbole oraz polecenia sterujące wyświetlaniem lub drukowaniem.

kod ASCII

W rzeczywistości powstało wiele kodów o tej nazwie, dlatego że w wielu krajach stosuje się znaki narodowe, takie jak w języku polskim: ą, ę, ł.

Zapoznaj się z tabelą kodów ASCII (tab. 22.1.). Kolumny kolejno zawierają:

- liczbę w postaci binarnej (BIN),
- liczbę w postaci dziesiętnej (DEC),
- liczbę w postaci szesnastkowej (HEX),
- znak odpowiadający liczbie (Znak).

znaki narodowe



BIN	DEC	HEX	Znak
0010 0000	32	20	Spacja
0010 0001	33	21	!
0010 0010	34	22	„
0010 0011	35	23	#
0010 0100	36	24	\$
0010 0101	37	25	%
0010 0110	38	26	&
0010 0111	39	27	,
0010 1000	40	28	(
0010 1001	41	29	)
0010 1010	42	2A	*
0010 1011	43	2B	+
0010 1100	44	2C	.
0010 1101	45	2D	-
0010 1110	46	2E	.
0010 1111	47	2F	/
0011 0000	48	30	0
0011 0001	49	31	1
0011 0010	50	32	2
0011 0011	51	33	3
0011 0100	52	34	4
0011 0101	53	35	5
0011 0110	54	36	6
0011 0111	55	37	7
0011 1000	56	38	8
0011 1001	57	39	9
0011 1010	58	3A	:
0011 1011	59	3B	;
0011 1100	60	3C	<
0011 1101	61	3D	=

BIN	DEC	HEX	Znak
0011 1110	62	3E	>
0011 1111	63	3F	?
0100 0000	64	40	@
0100 0001	65	41	A
0100 0010	66	42	B
0100 0011	67	43	C
0100 0100	68	44	D
0100 0101	69	45	E
0100 0110	70	46	F
0100 0111	71	47	G
0100 1000	72	48	H
0100 1001	73	49	I
0100 1010	74	4A	J
0100 1011	75	4B	K
0100 1100	76	4C	L
0100 1101	77	4D	M
0100 1110	78	4E	N
0100 1111	79	4F	O
0101 0000	80	50	P
0101 0001	81	51	Q
0101 0010	82	52	R
0101 0011	83	53	S
0101 0100	84	54	T
0101 0101	85	55	U
0101 0110	86	56	V
0101 0111	87	57	W
0101 1000	88	58	X
0101 1001	89	59	Y
0101 1010	90	5A	Z
0101 1011	91	5B	[



BIN	DEC	HEX	Znak	BIN	DEC	HEX	Znak
0101 1100	92	5C	\	0110 1110	110	6E	n
0101 1101	93	5D	]	0110 1111	111	6F	o
0101 1110	94	5E	^	0111 0000	112	70	p
0101 1111	95	5F	_	0111 0001	113	71	q
0110 0000	96	60	`	0111 0010	114	72	r
0110 0001	97	61	a	0111 0011	115	73	s
0110 0010	98	62	b	0111 0100	116	74	t
0110 0011	99	63	c	0111 0101	117	75	u
0110 0100	100	64	d	0111 0110	118	76	v
0110 0101	101	65	e	0111 0111	119	77	w
0110 0110	102	66	f	0111 1000	120	78	x
0110 0111	103	67	g	0111 1001	121	79	y
0110 1000	104	68	h	0111 1010	122	7A	z
0110 1001	105	69	i	0111 1011	123	7B	{
0110 1010	106	6A	j	0111 1100	124	7C	
0110 1011	107	6B	k	0111 1101	125	7D	}
0110 1100	108	6C	l	0111 1110	126	7E	~
0110 1101	109	6D	m	0111 1111	127	7F	Delete

Tab. 22.1. Standardowe kody ASCII odpowiadające znakom

Liczby w tabeli rozpoczynają się od 32. Kody poniżej tej liczby dotyczą sterowania wyświetlaniem i drukiem, np. przejściem do nowej linii, zakończeniem druku, sygnałem dźwiękowym itp. Jeśli chcesz poznać znaczenie tych kodów, znajdź odpowiednie informacje w internecie.

Zauważ, że w kolumnie BIN wszystkie liczby na pozycji najstarszej mają wartość 0. Oznacza to, że w klasycznym kodzie ASCII do zakodowania wszystkich znaków i poleceń wystarczy 128 liczb.

W tabeli nie ma polskich znaków narodowych, a jednak system Windows je wyświetla. Nietrudno odgadnąć, że przydzielono im kody powyżej 127.

Do reprezentacji cyfrowej znaków używanych między innymi w języku polskim system Windows stosuje tzw. stronę kodową Windows-1250 CE. Odpowiednie kody znajdziesz w tabeli 22.2.



BIN	DEC	HEX	Znak
10100100	164	A4	Ą
10001111	143	8F	Ć
10101000	168	A8	Ę
10011101	157	9D	Ł
11100011	227	E3	Ń
11100000	224	E0	Ó
10010111	151	97	Ś
10001101	141	8D	Ż
10111101	189	BD	Ž

BIN	DEC	HEX	Znak
10100101	165	A5	ą
10000110	134	86	ć
10101001	169	A9	ę
10001000	136	88	ł
11100100	228	E4	ń
10100010	162	A2	ó
10011000	152	98	ś
10101011	171	AB	ż
10111110	190	BE	ž

kody
Windows-1250

Tab. 22.2. Kody Windows-1250 odpowiadające polskim znakom narodowym

Jak sprawdzić, czy rzeczywiście system Windows tak właśnie koduje litery? Możesz wykonać doświadczenie, wykorzystując klawiaturę z wydzieloną częścią numeryczną (rys. 22.3.).



Rys. 22.3. Sposób wpisywania znaku za pomocą kodów i klawiatury numerycznej

Z tabeli ASCII (tab. 22.1.) i Windows-1250 wynotuj litery odpowiadające kodom: 112 165 99 122 101 107

Czy otrzymałeś/otrzymałaś wyraz *pączek*?

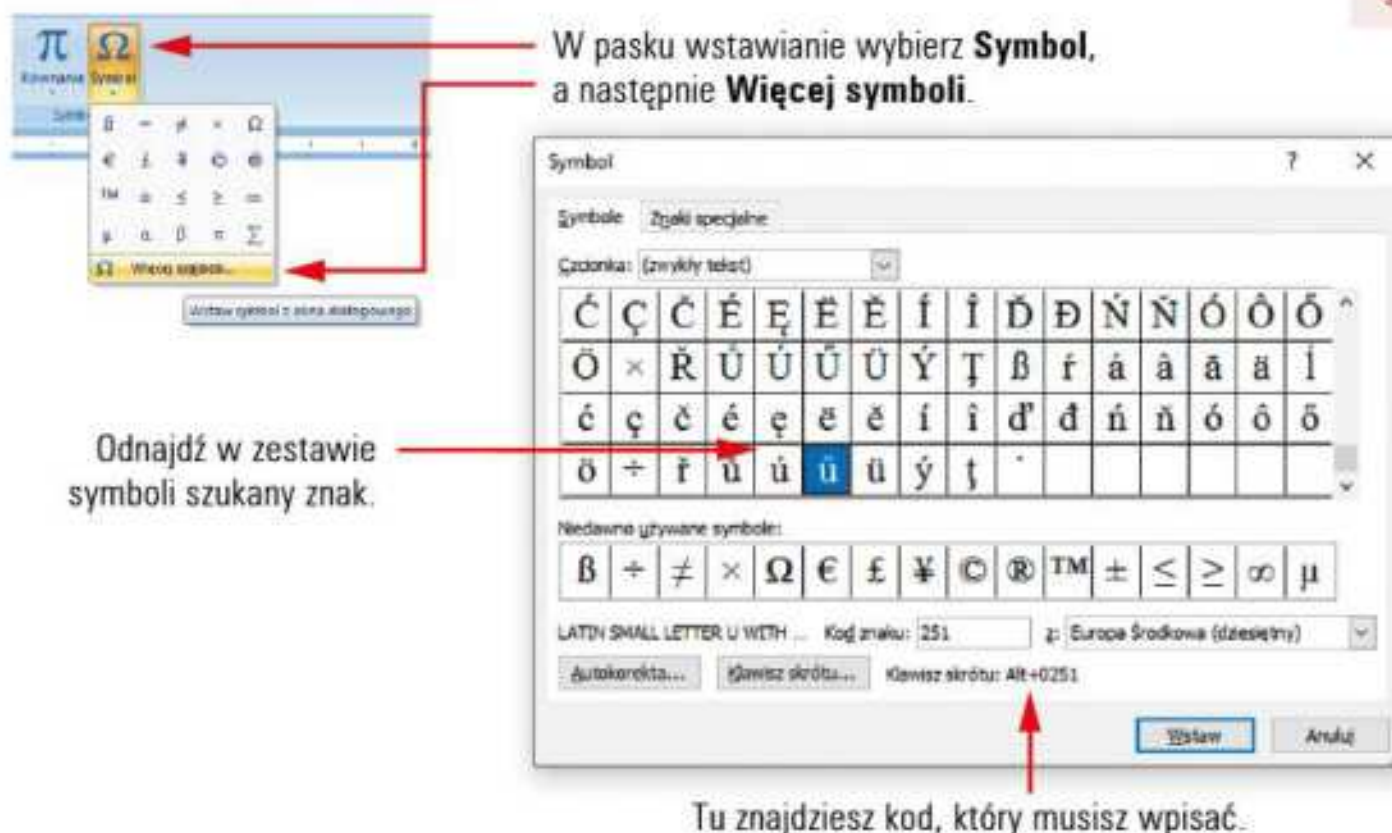
Uruchom edytor tekstu w systemie Windows i korzystając z instrukcji (rys. 22.3.), wprowadź te kody do komputera. Czy w edytorze wyświetliły się kolejne litery słowa *pączek*? Teraz już wiesz, że system Windows używa kodów do edycji tekstu.

Umiejętność wprowadzania kodów przydaje się programistom, ale także wszystkim przy edytowaniu tekstu. Oczywiście do wprowadzania tekstu służy standardowa klawiatura, jednak niektórych znaków na niej nie ma. Przykładem mogą być litery używane w języku niemieckim. Na niemieckiej klawiaturze je znajdziesz, ale na polskiej nie, a jednak czasami trzeba takiego znaku użyć. Wprowadź np. kody – 0220 0252, a otrzymasz znaki Ü ü.



Skąd można się dowiedzieć, jakie kody odpowiadają danym znakom w Windows? Użyj np. opcji **Wstaw Symbol** w MS Word (rys. 22.4.).

wstawianie
symboli



Rys. 22.4. Wstawianie symbolu w MS Word i kod znaku

Oczywiście wstawianie pojedynczego znaku jest łatwiejsze za pomocą myszy, ale jeśli używa się danego symbolu często, warto zapamiętać jego kod i wprowadzać go z klawiatury.

22.2. Plagiaty, czyli komputer szuka złodzieja

cytowanie innych
autorów

Kopiowanie tekstów nie zawsze jest legalne. Prawo dopuszcza cytowanie materiałów innych autorów. Cytat musi jednak spełniać kilka warunków, a przede wszystkim powinno istnieć uzasadnienie dla jego wykorzystania. Poniższy tekst jest cytatem:

Cytowanie poddane jest pewnym ograniczeniom. Możemy we własnej pracy przytoczyć całość bądź fragment innego dzieła (wszystko jedno, czy jest to tekst, zdjęcie czy film), ale tylko wtedy, gdy jest to uzasadnione celami takimi jak wyjaśnianie, polemika, analiza krytyczna lub naukowa, nauczanie lub prawa gatunku twórczości.

W literaturoznawczej książce naukowej autor legalnie przytacza obszernie fragmenty współczesnej prozy w celu ilustracji i omówienia stawianych przez siebie tez.

Źródło: <https://prawokultury.pl/kurs/prawo-cytatu>

Jeśli autor jest nieuczciwy i nie oznaczy odpowiednio tekstu zaczerpniętego z innego dzieła, to popełnia **plagiat**. Przepisanie wypracowania, rozwiązania zadania itp. też jest plagiatem. Zdarza się niestety, że pracownicy naukowcy taki plagiat popełniają. Skąd wiadomo, że to zrobili? Oczywiście autor stwierdzi tak po przeczytaniu „dzieła”.



Jednolity System Antyplagiatowy

Obecnie na uczelniach w Polsce badane są wszystkie prace dyplomowe oraz doktorskie. Z pomocą przychodzą oczywiście rozwiązania informatyczne. Promotorzy takich prac muszą obowiązkowo korzystać z JSA, czyli **Jednolitego Systemu Antyplagiatowego**. To bardzo silne narzędzie informatyczne i umocowane w prawie ustawowo (rys. 22.5.).

Jakie prace dyplomowe badane są przez Jednolity System Antyplagiatowy?

Zgodnie z treścią art 76 ust. 4 Ustawy z dnia 20 lipca 2018 r. Prawo o szkolnictwie wyższym i nauce w systemie JSA badane są wszystkie pisemne prace dyplomowe przed egzaminem dyplomowym: licencjackie, inżynierskie, magisterskie oraz doktorskie.

Należy mieć na względzie, iż w Ustawie 2.0 w kontekście systemu antyplagiatowego oraz studiów zakończonych dyplomem i uzyskaniem stopnia lub tytułu naukowego, mowa jest o studiach pierwszego, drugiego stopnia, jak również trzeciego stopnia.

Permalink

Informacja ze strony: <https://jsa.opi.org.pl/centrum-pomocy/faq-kategoria/adm/> dotycząca obowiązku badania prac przez JSA.

Rys. 22.5. Jakie prace bada JSA?

Dzięki temu, że każdy promotor musi korzystać z systemu JSA (rys. 22.6.), znacznie zmniejszyła się liczba plagiatów w pracach dyplomowych.

Czy każdy promotor musi być wprowadzony do rejestru pracowników w POL-on 2.0?

Zgodnie z interpretacją MNiSW: Tak.

Pełnienie funkcji promotora wiąże się z prowadzeniem zajęć, a co najmniej seminarium. W zakres obowiązków promotora wchodzi opieka nad studentem. W związku z tym, taka osoba powinna być wprowadzona do rejestru w systemie POL-on 2.0 jako "inna osoba prowadząca zajęcia".

Permalink

Informacja ze strony: <https://jsa.opi.org.pl/centrum-pomocy/faq-kategoria/mnisw/> dotycząca obowiązku rejestrowania promotorów w JSA.

Rys. 22.6. Promotor musi być zarejestrowany w systemie JSA



ZADANIA DO WYKONANIA

- Wykonaj polecenie: 80 111 100 97 106 32 107 111 100 121 32 65 83 67 73 73 32 84 119 111 105 99 104 32 105 110 105 99 106 97 136 162 119. Polecenie odczytaj za pomocą klawiatury numerycznej.
- Zapisz swoje imię w postaci kodów ASCII. Skorzystaj z tabeli 22.1.
- Ile liczb – kodów ASCII należy użyć, by zakodować tekst:
Litwo! Ojczyzno moja! Ty jesteś jak zdrowie.
Ile cię trzeba cenić, ten tylko się dowie,
Kto cię stracił.
 Nie uwzględniaj przejść do nowej linii.
- Zapoznaj się z zagadnieniami opisanymi na stronie jsa.opi.org.pl dotyczącymi plagiatów i wykonaj prezentację uwzględniającą najważniejsze informacje na ten temat. Uzasadnij swój wybór.

PODSUMOWANIE LEKCJI

- Kody ASCII to liczby odpowiadające znakom drukowanym lub wyświetlanym. Każdemu znakowi przyporządkowano jeden bajt. s. 173
- W rzeczywistości powstało wiele kodów ASCII zawierających znaki charakterystyczne dla danego języka, np. polskie ą, ę, ó, ł itp. s. 173
- W systemie Windows stosuje się stronę kodową Windows-1250 CE, będącą odpowiednikiem ASCII dla krajów Europy Środkowo-Wschodniej. s. 176
- Wprowadzając odpowiednie kody z klawiatury numerycznej (z lewym Alt), można do tekstu wstawiać znaki i symbole. s. 177
- We własnych opracowaniach można cytować fragmenty innych dzieł. s. 177
- Każda praca naukowa i dyplomowa na studiach wyższych musi być zweryfikowana przez Jednolity System Antyplagiatowy. s. 178



23. Metoda naiwna, czyli szukamy wzorca w tekście

NA TEJ LEKCJI:

- poznasz jedną z metod wyszukiwania wzorca w tekście;
- ułożysz algorytm zgodny z metodą naiwną.

Z wyszukiwaniem wzorców znaków w tekście spotykasz się bardzo często. Opcja taka jest dostępna między innymi w edytorach tekstu, przeglądarkach internetowych i na stronach w sieci. Najbardziej zaawansowane algorytmy wspomagają ten proces w wyszukiwarkach internetowych. Algorytmy skutecznego i szybkiego wyszukiwania są ciągle modyfikowane. Jeden z najprostszych opisuje metodę naiwną.

23.1. Zapisujemy metodę, czyli tworzymy opis metody naiwnej

wzorzec tekstu

Zdefiniujmy pojęcie wzorca tekstu.

Wzorzec występujący w ciągu znaków jest spójnym jego fragmentem. Oznacza to, że nie może być rozdzielony innymi fragmentami, np. w sekwencji znaków tekstu „Litwo! Ojczyzno moja!” wzorcem może być „two”, ale „twoj” nie. Nie jest to bowiem spójny ciąg znaków. Analogicznie każdy spójny ciąg znaków, **który chcemy znaleźć w tekście**, możemy nazwać wzorcem.

idea algorytmu
naiwnego

Idea algorytmu naiwnego wyszukiwania wzorca w tekście polega na przeglądaniu danego tekstu sekwencyjnie. Przesuwamy wzorzec po kolei po jednym znaku i sprawdzamy, czy kolejne znaki wzorca są identyczne ze znakami tekstu. Łatwo to sobie wyobrazisz, jeśli szukany wzorzec zapiszesz na kartce i przyłożysz do tekstu, jak to pokazano na rysunku 23.1.

Pierwszy znak wzorca został przyrównany z pierwszym znakiem tekstu.

↓
Litwo! Ojczyzno moja!
[two]→

Po wykryciu niezgodności pierwszego znaku wzorca z tekstem następuje przesunięcie wzorca i porównanie pierwszego znaku z następną literą tekstu.

↓
Litwo! Ojczyzno moja!
Litwo!
[two]→

Wykryta zgodność pierwszego znaku wzorca z tekstem. Następuje sprawdzanie drugiego znaku.

↓
Litwo! Ojczyzno moja!
Litwo!
[two]

↓
Litwo! Ojczyzno moja!
Litwo!
[two]

↓
Litwo! Ojczyzno moja!
Litwo!
[two]

Drugi i trzeci znak także są zgodne z tekstem.
Wykryto zatem występowanie wzorca w tekście.

Rys. 23.1. Przykład ilustrujący metodę naiwną wyszukiwania wzorca w tekście



Zacznij od ułożenia karteczki z wzorcem pod pierwszymi literami tekstu, zaczynając od lewej. i nie jest równe t . Przesuwamy więc karteczkę z wzorcem o jedną literę w prawo. Widzimy, że i także nie jest równe t . Następne przesunięcie przyniesie pozytywny wynik. Można więc sprawdzać zgodność drugiej litery wzorca z drugą literą sprawdzanego obszaru tekstu. Nastąpiła zgodność. Pozostaje już tylko sprawdzić ostatni znak. Jeśli zależy nam na znalezieniu jedynie pierwszego występowania wzorca w tekście, na tym możemy zakończyć działania.

W przeciwnym przypadku porównania (wyszukiwanie) należy kontynuować. Ale czy do ostatniej litery tekstu? Nie. Sprawdzanie ostatnich liter nie ma sensu, gdy pozostanie ich mniej niż we wzorcu. Jeśli $dlTekstu$ będzie liczbą jego znaków, a $dlWzorca$ liczbą znaków we wzorcu, to porównywanie będzie trwało aż do sprawdzenia

$dlTekstu - dlWzorca + 1$ znaków tekstu.

Powyższy opis tłumaczy sposób funkcjonowania metody naiwnej. Można także zapisać ten przepis w postaci algorytmu. Zanim do tego przystąpimy, ustalmy nazwy zmiennych.

$t[]$ – tablica typu `char` zawierająca tekst

$w[]$ – tablica typu `char` zawierająca wzorec

$dlWzorca$ – liczba znaków we wzorcu

$dlTekstu$ – liczba znaków w tekście

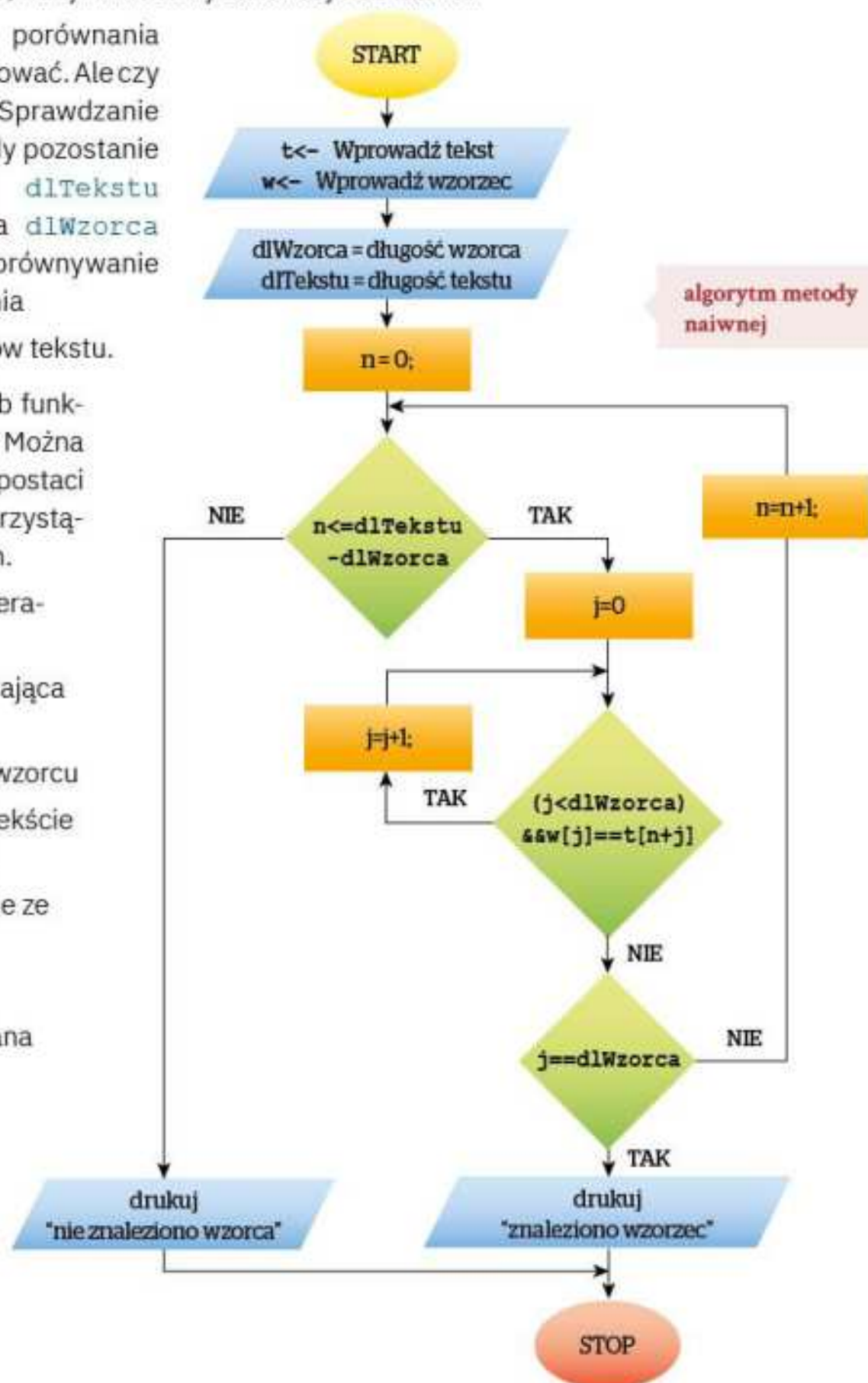
n – pozycja w tekście, od której zaczyna się aktualne porównanie ze wzorcem

j – pozycja znaku we wzorcu

$t[j+n]$ – aktualnie porównywana z wzorcem litera tekstu

$w[j]$ – aktualnie porównywana z tekstem litera wzorca

Pierwszy znak tekstu i wzorca mają index 0 zgodnie z zasadami numerowania elementów tablicy, a ostatnie znaki odpowiednio: $dlTekstu-1$ i $dlWzorca-1$.



Rys. 23.2. Przykładowy algorytm metody naiwnej wyszukiwania wzorca w tekście



test algorytmu

Przeanalizujemy algorytm dla przykładu z rysunku 23.2.

Tekst:

Litwo! Ojczyzno moja!

Wzorzec:

two

Wczytanie tekstu do tablicy t[] oraz wzorca do tablicy w[]	
Obliczenie liczby znaków w tekście dlTekstu=21 oraz liczby znaków we wzorcu dlWzorca=3	
Zerowanie zmiennej pomocniczej	n=0;
Badanie warunku:	0<=21-3
n<=dlTekstu-dlWzorca	TAK
j=0;	
Badanie warunku:	(0<3) && w[0]=t ≠ t[0+0]=L
(j<dlWzorca) && w[j]==t[n+j]	NIE
Badanie warunku:	j=0 ≠ 3
j==dlWzorca	NIE
n=n+1;	n=0+1=1
Badanie warunku:	1<=21-3
n<=dlTekstu-dlWzorca	TAK
j=0;	
Badanie warunku:	(0<3) && w[0]=t ≠ t[1+0]=i
(j<dlWzorca) && w[j]==t[n+j]	NIE
Badanie warunku:	j=0 ≠ 3
j==dlWzorca	NIE
n=n+1;	n=1+1=2
Badanie warunku:	2<=21-3
n<=dlTekstu-dlWzorca	TAK
j=0;	
Badanie warunku:	(0<3) && w[0]=t = t[2+0]=t
(j<dlWzorca) && w[j]==t[n+j]	TAK
j=j+1;	j=0+1=1
Badanie warunku:	(1<3) && w[1]=w = t[2+1]=w
(j<dlWzorca) && w[j]==t[n+j]	TAK
j=j+1;	j=1+1=2
Badanie warunku:	(2<3) && w[2]=o = t[2+2]=o
(j<dlWzorca) && w[j]==t[n+j]	TAK
j=j+1;	j=2+1=3
Badanie warunku:	(3<3)
(j<dlWzorca) && w[j]==t[n+j]	NIE
Badanie warunku:	j=3 = dlWzorca=3
j==dlWzorca	TAK
Znaleziono wzorzec	STOP

Tab. 23.1. Przykład testu algorytmu wyszukiwania wzorca metodą naiwną



ZADANIA DO WYKONANIA

1. Sprawdź działanie algorytmu z rysunku 23.2. dla kolejnych dwóch wersów inwokacji i wzorca `ceni`. Sporządź odpowiednią tabelę na wzór tabeli 23.1.
2. Sprawdź działanie algorytmu (rys. 23.2.) dla wzorca niezawierającego się w tekście. Sporządź odpowiednią tabelę na wzór tabeli 23.1.
3. Zapisz w pliku tekstowym algorytm wyszukiwania wzorca w tekście metodą naiwną w postaci listy. Sprawdź jego działanie na przykładzie z tabeli 23.1.
4. Wykonaj prezentację omawiającą działanie metody naiwnej wyszukiwania wzorca w tekście. Użyj narzędzi edytora do wizualnego przedstawienia tej metody.
- 5*. Zmodernizuj algorytm z rysunku 23.2. w taki sposób, by po znalezieniu wzorca zapamiętał jego miejsce w tekście i kontynuował poszukiwania aż do znalezienia wszystkich jego wystąpień.
- 6*. Ułóż algorytm metody naiwnej wyszukiwania wzorca w tekście w postaci pseudokodu. Sprawdź jego działanie na przykładzie z tabeli 23.1.

PODSUMOWANIE LEKCJI

- **Wzorzec występujący** w ciągu znaków jest spójnym jego fragmentem. s. 180
- **Idea algorytmu naiwnego wyszukiwania wzorca w tekście** polega na przeglądaniu danego tekstu sekwencyjnie. Przesuwamy wzorzec po kolei po jednym znaku i sprawdzamy, czy kolejne znaki wzorca są identyczne ze znakami tekstu. s. 180
- **Zakończenie działania algorytmu metody naiwnej** następuje w momencie, w którym w sprawdzanym tekście pozostaje mniej znaków niż zawiera wzorzec. s. 181
- **Algorytm metody naiwnej** szukania wzorca w tekście można zapisać w postaci graficznej. s. 181
- **Każdy algorytm należy przetestować** na wielu zestawach danych. s. 182



24. Realizacja algorytmu, czyli program szuka wzorca

NA TEJ LEKCJI:

- ułożysz program wyszukiujący wzorzec w tekście.

Znasz już metodę naiwną wyszukiwania wzorca w tekście. Wiesz, jak działa przykładowy algorytm. Na jego podstawie można ułożyć skuteczny program. Oczywiście problem zakodowania algorytmu można rozwiązać na wiele sposobów. Programy mogą się różnić nie tylko sposobem wyświetlania wyniku, lecz także zastosowaniem różnych instrukcji i rozkazów. Zapoznaj się z jednym z możliwych rozwiązań.

24.1. Według przepisu, czyli układamy program realizujący przykładowy algorytm

założenia

Ponieważ nasz program ma głównie realizować algorytm, uproścmy go, z góry zakładając maksymalną długość tekstu (`dlTekstu=100`) i wzorca (`dlWzorca=100`). Dodajmy za to ciekawy sposób wyświetlania wyniku (rys. 24.1.). Użyjemy w tym celu strumienia `cout`.

`cout.width`

Zastosowanie `cout.width(n+dlWzorca)` pozwoli określić, ile znaków łącznie będzie zawierał strumień. Dzięki temu wzorzec zostanie wyświetlony w odpowiednim miejscu (rys. 24.1.). Uwzględnia białe znaki, które są niewidoczne na ekranie. Przykładem może być spacja.

białe znaki

`cin.getline`

Dzięki użyciu `cin.getline` będzie możliwe wczytanie tekstu lub wzorca do momentu użycia enteru. Tablice `t[]` i `w[]` zostaną wypełnione wprowadzonym tekstem.

```
Wprowadź tekst: Litwo! Ojczyzno moja!  
Wprowadź wzorzec: two  
Znaleziono wzorzec  
Pierwsze wystąpienie wzorca od 3 znaku tekstu  
Litwo! Ojczyzno moja!  
two
```

Wzorzec jest wyświetlany w miejscu pierwszego wystąpienia w tekście.

Rys. 24.1. Przykładowy sposób wyświetlania wyniku działania programu wyszukiującego wzorzec w tekście



funkcja
strlen()

W programie zastosujemy także funkcję `strlen()` (zawartą w bibliotece `string.h`), która oblicza (zwraca) liczbę znaków tablicy przekazanej w nawiasie jako jej argument. Pozwoli to na ustalenie długości tekstu (`t[]`) i wzorca (`w[]`).

```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4
5 int main()
6 {
7     char t[100], w[100];
8     int n=0;
9     cout << "Wprowadź tekst: ";
10    cin.getline(t, 100);
11    cout << "Wprowadź wzorzec: ";
12    cin.getline(w, 100);
13
14    int dlTekstu = strlen(t);
15    int dlWzorca = strlen(w);
16
17    while (n<=dlTekstu-dlWzorca)
18    {
19        int j = 0;
20        while ((j<dlWzorca) && w[j]==t[n + j])
21            j=j+1;
22        if (j==dlWzorca)
23        {
24            cout << "Znaleziono wzorzec" << endl;
25            cout << "Pierwsze wystąpienie wzorca od " << n + 1 << " znaku tekstu" << endl << endl;
26            cout << t << endl;
27            cout.width(n+dlWzorca);
28            cout << w << endl;
29            return 0;
30        }
31        n=n+1;
32    }
33    cout << "nie znaleziono wzorca";
34    return 0;
35 }
36
```



```
Znaleziono wzorzec
Pierwsze wystąpienie wzorca od 3 znaku tekstu

Litwo! Ojczyzno moja!
two
```

Rys. 24.2. Przykładowy program poszukujący wzorca w tekście



ZADANIA DO WYKONANIA

1. Czy program z rysunku 24.2. będzie działał poprawnie, gdy wprowadzisz wzorzec dłuższy od tekstu? Uzasadnij swoją odpowiedź. Sprawdź ją doświadczalnie dla różnych danych.
2. Uzasadnij nadanie początkowych wartości 0 zmiennym `n` oraz `j`.
3. Opisz rolę zmiennej `j` w programie z rysunku 24.2.
- 4*. Ułóż program różniący się sposobem realizacji metody naiwnej wyszukiwania wzorca w tekście od przedstawionego na rysunku 24.2. Opisz jego działanie.
- 5*. Zmodyfikuj program z rysunku 24.2. w taki sposób, by odnaleziony wzorzec był wskazany w tekście kolorem czerwonym. Pomiń wyświetlanie wzorca pod tekstem.

PODSUMOWANIE LEKCJI

- s. 184 • Dla uproszczenia możesz przyjąć stałą wielkość tablic `t[]` oraz `w[]`.
- s. 184 • `cout.width` określi, ile znaków łącznie będzie zawierał strumień.
- s. 184 • Dzięki użyciu `cin.getline` będzie możliwe wprowadzenie tekstu lub wzorca do momentu użycia enteru.
- s. 184 • Białe znaki są znakami niewidocznymi na ekranie.
- s. 184 • Sposób wyświetlania efektu działania programu może wskazywać jednocześnie miejsce w tekście, w którym znaleziono wzorzec.
- s. 185 • Funkcja `strlen()` (zawarta w bibliotece `string.h`) oblicza (zwraca) liczbę znaków tablicy przekazanej w nawiasie jako jej argument.
- s. 185 • Przykładowy program ma ograniczenia, np. liczbę znaków w tekście lub wzorcu.



25. Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego

NA TEJ LEKCJI:

- przypomnisz sobie sposób obliczania wartości pierwiastka;
- dowiesz się, czym jest szacowanie pierwiastka kwadratowego;
- dowiesz się, w jaki sposób komputer oblicza wartość pierwiastka i dlaczego jest to wartość przybliżona.

Pojęcie pierwiastka jest ci doskonale znane. Zdajesz sobie sprawę, że jego wartość nie zawsze jest dokładnie określona. Dla liczb rzeczywistych są to najczęściej wartości przybliżone. Można je podawać z określoną dokładnością. Programy komputerowe wyznaczają wartości przybliżone z określoną liczbą miejsc po przecinku.

25.1. Obliczanie wartości pierwiastka kwadratowego

Przypomnimy sobie, czym jest pierwiastek oraz jak się oblicza jego wartość. Pierwiastek stopnia 2 (kwadratowy) liczby a równa się b , gdy b podniesione do potęgi 2 jest równe a , czyli $\sqrt[2]{a} = \sqrt{a} = b$, ponieważ $b^2 = a$, czyli np.:

$$\sqrt[2]{16} = \sqrt{16} = 4, \text{ ponieważ } 4^2 = 16$$

Obliczanie wartości pierwiastka kwadratowego wydaje się łatwe, gdy mamy do czynienia z liczbą (pod pierwiastkiem) będącą wynikiem podniesienia do potęgi 2 pewnej liczby całkowitej. Niestety program komputerowy nie działa tak jak ludzki umysł i dla każdej liczby wykonuje obliczenia w ten sam sposób. W kolejnych krokach dokonuje coraz to dokładniejszych przybliżeń, aż do osiągnięcia oczekiwanej dokładności. W językach programowania dostępne są funkcje obliczające wartość pierwiastka. W C++ znajdują się one w bibliotece `math.h`.

Przykład.

Wyznaczanie pierwiastka gotową funkcją `pow()` w programie C++. Aby użyć tej funkcji, musimy zaimportować bibliotekę `math.h`.

```
#include <iostream>
#include <cstdlib>
#include <math.h>
using namespace std;
int main()
```

pierwiastek
kwadratowy


```

{
    float liczba, stopien;
    cout << "Wprowadź liczbę: ";
    cin >> liczba;
    cout << "Wprowadź stopień pierwiastka: ";
    cin >> stopien;
    cout << pow(liczba, 1/stopien);
    return 0;
}

```

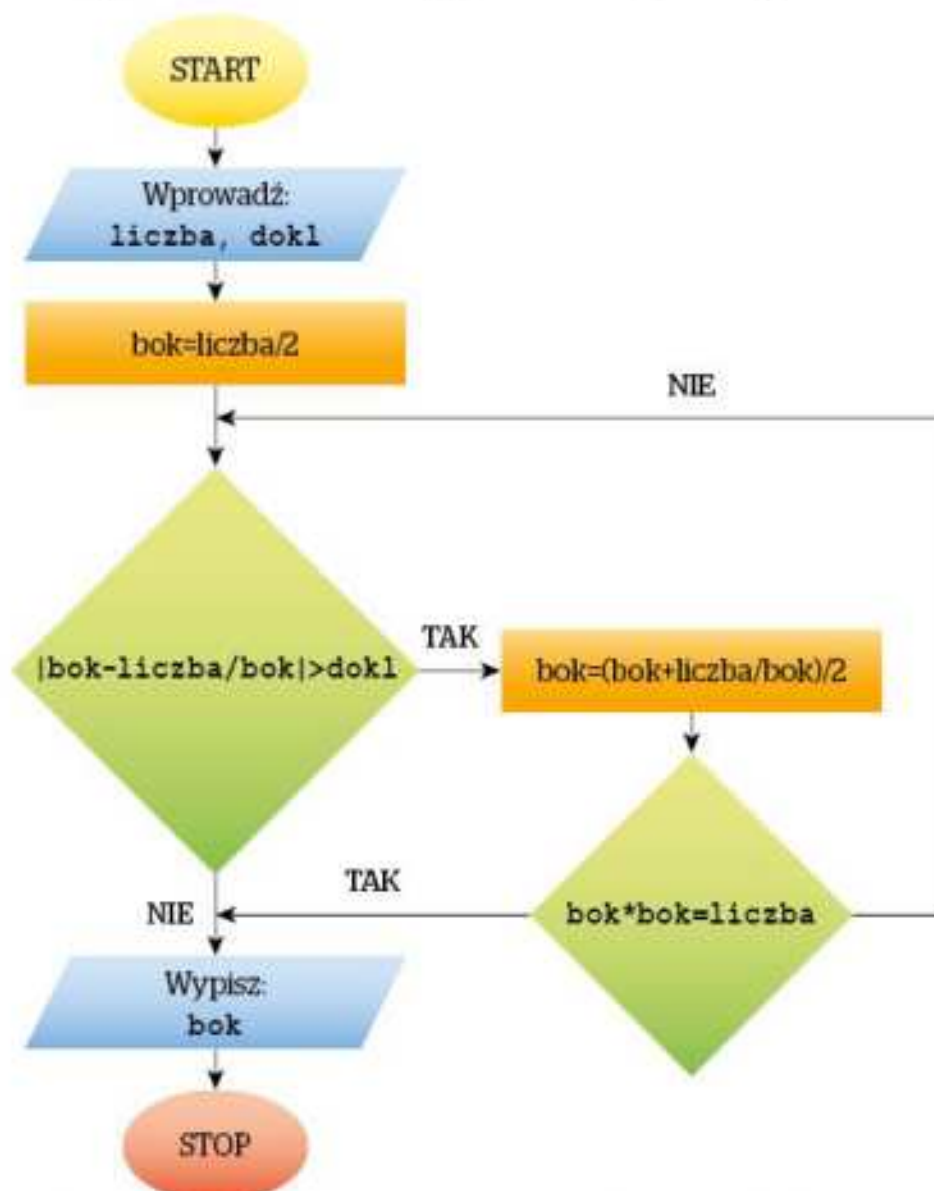
25.2. Obliczamy przybliżoną wartość pierwiastka kwadratowego

Twoja intuicja zapewne podpowiada ci, jak ułożyć algorytm obliczania pierwiastka kwadratowego. Pozostanie ci tylko rozwiązanie problemu uwzględnienia stopnia przybliżenia obliczeń (rys. 25.1).

liczba – zmienna typu `double` zawierająca zadaną liczbę rzeczywistą

dokl – zmienna typu `double` zawierająca dokładność obliczania

bok – zmienna typu `double` zawierająca informacje o długości



Rys. 25.1. Przykład algorytmu obliczającego wartość pierwiastka kwadratowego, uwzględniający liczbę pozycji po przecinku



Sprawdźmy działanie algorytmu na przykładzie liczby 36 z założoną dokładnością 0,0001 (tab. 25.1.).

liczba = 36, dokl = 0.0001	
bok = liczba/2 czyli 36 / 2 = 18	
Badamy warunek w pętli: bok - liczba / bok > dokl	czy 18 - 36 / 18 > 0.0001 TAK
Obliczanie zmiennej bok (bok+liczba/bok)/2	(18 + 36 / 18) / 2 = 10
Badanie warunku: bok * bok == liczba	czy 10 * 10 = 36 NIE
Badamy warunek w pętli: bok - liczba / bok > dokl	czy 10 - 36 / 10 > 0.0001 TAK
Obliczanie zmiennej bok (bok+liczba/bok)/2	(10 + 36 / 10) / 2 = 6.8
Badanie warunku: bok * bok == liczba	czy 6.8 * 6.8 = 36 NIE
Badamy warunek w pętli: bok - liczba / bok > dokl	6.8 - 36 / 6.8 > 0.0001 TAK
Obliczanie zmiennej bok (bok+liczba/bok)/2	(6.8 + 36 / 6.8) / 2 = 6.04706
Badanie warunku: bok * bok == liczba	czy 6.04706 * 6.04706 = 36 NIE
Badamy warunek w pętli: bok - liczba / bok > dokl	czy 6.04706 - 36 / 6.04706 > 0.0001 TAK
Obliczanie zmiennej bok (bok+liczba/bok)/2	(6.04706 + 36 / 6.04706) / 2 = 6.00018
Badanie warunku: bok * bok == liczba	czy 6.00018 * 6.00018 = 36 NIE
Badamy warunek w pętli: bok - liczba / bok > dokl	czy 6.00018 - 36 / 6.00018 > 0.0001 TAK
Obliczanie zmiennej bok (bok+liczba/bok)/2	(6.00018 + 36 / 6.00018) / 2 = 6
Badanie warunku: bok * bok == liczba	czy 6*6 = 36 TAK
Wyjście z pętli, wyświetlenie wartości zmiennej bok	STOP bok=6

Tab. 25.1. Przebieg obliczeń wykonanych przez badany algorytm



25.3. Funkcja obliczająca pierwiastek kwadratowy

Skoro nasz algorytm działa, czas ułożyć program i dobrze go przetestować. Samo obliczanie pierwiastka zazwyczaj jest częścią większego programu. Często musi być stosowane w wielu miejscach kodu. Najlepszym rozwiązaniem będzie więc utworzenie odpowiedniej funkcji. Oto przykład takiego programu:

```
#include <iostream>
#include <cmath>
using namespace std;
//funkcja obliczająca wartość pierwiastka
double pierwiastek(double liczba, double dokl)
{
    double wynik;
    wynik=liczba/2;
    while (fabs(wynik-liczba/wynik)>dokl)
    {
        wynik=(wynik+liczba/wynik)/2;
        if (wynik*wynik==liczba)
            break;
    }
    return wynik;
}
int main()
{
    double liczba;
    double dokl;
    cout << "Podaj liczbę rzeczywistą: " << endl;
    cin >> liczba;
    cout << "Podaj dokładność obliczeń: " << endl;
    cin >> dokl;
    //wywołanie funkcji obliczającej pierwiastek
    cout << pierwiastek(liczba,dokl);
    return 0;
}
```



Sprawdźmy działanie programu dla 36, czyli tej samej liczby, którą testowaliśmy algorytm (tab. 25.1).

Podaj liczbę rzeczywistą :

36

Podaj dokładność obliczeń :

0.0001

6

Dokładność obliczeń zależy od wartości zmiennej `dokl`.

Podaj liczbę rzeczywistą :

36

Podaj dokładność obliczeń :

0.01

6.00018310546875

Podaj liczbę rzeczywistą :

36

Podaj dokładność obliczeń :

0.1

6.0470590591430664062

Rys. 25.2. Wynik działania programu dla różnych wartości liczby określającej liczbę przybliżeń.

ZADANIA DO WYKONANIA

1. Sprawdź działanie algorytmu dla kilku liczb. Zbadaj, jak będą różniły się wyniki obliczeń dla różnych wartości zmiennej `dokl`. Określ progi, powyżej których wyniki przestaną się zmieniać.
2. Zmodyfikuj program z rozdziału w taki sposób, by wyświetlał pośrednie wyniki kolejnych przybliżeń. Przeanalizuj wynik jego działania.
3. Uruchom program z zadania 2 dla liczby 36 i przybliżenia 0.0001. Porównaj wyniki pośrednie z tabelą (tab. 27.1.) utworzoną na podstawie algorytmu. Wyciągnij wnioski.
4. Porównaj wyniki obliczeń programu z rozdziału z wynikami działania aplikacji Kalkulator. Opisz różnice, które zauważyłeś/zauważyłaś.
5. Napisz program w języku Java, który będzie obliczał wartość pierwiastka n -tego stopnia.

PODSUMOWANIE LEKCJI

- Pierwiastek z danej liczby obliczamy tak, że szukamy liczby, która podniesiona do drugiej potęgi da liczbę pod pierwiastkiem. Wartości pierwiastków liczb rzeczywistych są najczęściej przybliżone. Można je podawać z określoną dokładnością. Programy komputerowe wyznaczają ich wartości z określoną liczbą miejsc po przecinku.



26. Zbiór Cantora, czyli najprostsze tworzenie fraktali

NA TEJ LEKCJI:

- dowiesz się, czym jest fraktal oraz zbiór Cantora;
- poznasz, w jaki sposób tworzy się fraktale z wykorzystaniem zbioru Cantora.

Podzbiór prostej rzeczywistej, czyli osi liczbowej przedstawiającej zbiór liczb rzeczywistych, został opisany po raz pierwszy w 1883 roku przez niemieckiego matematyka Georga Cantora (czyt. kantora) i od jego nazwiska nazwany jest zbiorem Cantora. Naukowiec ten nie był jednak jego odkrywcą – zbiór ten został odkryty w 1875 roku przez brytyjskiego matematyka Henry’ego Johna Stephena Smitha (czyt. henrego dzona stiwen a smisa).

Zbiór Cantora jest najprostszym przykładem fraktala.

26.1. Fraktale

fraktale

Nazwa *fraktal* wywodzi się z języka łacińskiego i oznacza *złamany, ułamkowy, cząstkowy*. W języku potocznym nazywa się tak zwykle obiekt samopodobny, czyli taki, którego części są podobne do całości. Spotkamy się również z określeniem, że fraktal jest nieskończenie złożony, czyli ukazujący coraz bardziej złożone detale, jeżeli będziemy go powiększać.

Fraktale możemy dostrzec wszędzie – w projektach architektonicznych, w przyrodzie czy we wzornictwie przemysłowym. Charakteryzują się tym, że ich strukturę jesteśmy w stanie opisać bardzo prostymi wzorami matematycznymi. Są symetryczne. Ich wymiar nie jest liczbą całkowitą. Nie mają jednoznacznego kształtu. Określa się je jako zależności rekurencyjne, czyli cechujące się powtarzalnością wspomnianych wcześniej prostych wzorów matematycznych.

Fraktale znajdują swoje zastosowanie między innymi w przetwarzaniu i kodowaniu obrazów cyfrowych, badaniu struktury łańcuchów DNA i opisach procesów chaotycznych zachodzących w układach dynamicznych.



Rys. 26.1. Fraktal

26.2. Zbiór Cantora

Zbiór Cantora jest podzbiorem prostej rzeczywistej. Za jego pomocą możemy narysować bardzo ciekawy fraktal.

Zbiór Cantora rzędu 0 jest równy odcinkowi jednostkowemu $[0,1]$. Zbiór rzędu 1 uzyskujemy, dzieląc odcinek jednostkowy na trzy równe części i usuwając z niego odcinek środkowy. Składa się on zatem z dwóch fragmentów odcinka jednostkowego: $[0,1/3]$ i $[2/3,1]$.

zbiór Cantora

Zbiór Cantora rzędu $n+1$ stworzymy ze zbioru Cantora w następujący sposób: każdy z odcinków zbioru Cantora rzędu n dzielimy na 3 części i usuwamy odcinek środkowy, pozostawiając jego końce. Zgodnie z tą regułą, zbiór Cantora rzędu 2 będzie się składał z następujących odcinków: $[0,1/9]$, $[2/9,3/9]$, $[6/9,7/9]$, $[8/9,1]$.

Krok	Długość odcinka	Ilość odcinków	Obraz
0	1	1	
1	1/3	2	
2	1/9	4	
3	1/27	8	
k	$(1/3)^k$	2^k	

Tab. 26.1. Ilustracja działania algorytmu tworzącego fraktal na bazie zbioru Cantora



26.3. Kreślimy żółwiem fraktal na podstawie zbioru Cantora

Fraktale i programy je realizujące to zazwyczaj dobry przykład rekurencji. Do tej pory w języku C++ nie używaliście grafiki, a jej użycie wymagałoby poznania wielu nowych pojęć. Możecie jednak sprawdzić działanie algorytmu w innym języku. Dla przykładu i pokazania procesów rekurencyjnych użyjmy biblioteki `turtle` dla języka **Python**. Pozwala ona na kreślenie linii w podobny sposób jak to ma miejsce w języku **Logo**, w którym po ekranie porusza się **żółw** rysujący za sobą linie. Do ułożenia programu nie musicie jednak poznawać nowego języka, wystarczy znajomość instrukcji warunkowej `if` i pętli `for`. Pozostałe fragmenty programu będą zrozumiałe po przeanalizowaniu komentarzy (rys. 26.2.). Jeśli chcesz uruchomić program, możesz skorzystać ze środowiska programistycznego dla języka Python, np. znanego ci Eclipse (musisz doinstalować potrzebne komponenty, korzystając z opcji **Help/Install New Software**).

Uwaga!

W języku Python wcięcia tworzą bloki w strukturze programu podobnie jak nawiasy klamrowe w C++. Przepisując kod programu z podręcznika, zwróć na to baczną uwagę.

```
1 import turtle
2
3 t = turtle.Turtle()
4 # dl - długość rysunku
5 # wytnij - część do wycięcia
6 # n - ilość powtórzeń
7 def ZbiorCantor(dl, wytnij, n):
8     # jeśli n=0 to rysujemy tylko kreskę
9     if(n == 0):
10         t.pd()
11         t.forward(dl)
12         t.pu()
13     else:
14         nowa_dl = dl*(1 - wytnij)/2
15         ZbiorCantor(nowa_dl, wytnij, n - 1) # wywołanie rekurencyjne
16         t.forward(dl*wytnij)
17         ZbiorCantor(nowa_dl, wytnij, n - 1) # wywołanie rekurencyjne
18
19 window = turtle.Screen()
20 t.speed(500)
21 szerokosc = 800
22 wysokosc = 300
23 #ilość powtórzeń
24 n = 3
25 #ustawienie początkowe do rysowania
26 t.pu()
27 t.back(szerokosc/2)
28 t.lt(90)
29 t.forward(wysokosc/2)
30 # wywołanie
31 for i in range(n):
32     t.rt(90)
33     ZbiorCantor(szerokosc, 1/3, 1)
34     #ustawienie do rysowania kolejnej iteracji
35     t.back(szerokosc)
36     t.lt(90)
37     t.back(wysokosc / (n - 1))
38 window.exitonclick()
```

Wynik działania programu.



Rys. 26.2. Program tworzący fraktal na bazie zbioru Cantora



ZADANIA DO WYKONANIA

1. Ułóż algorytm (w postaci listy kroków), który poda liczbę odcinków w zbiorze Cantora rzędu n .
2. Ułóż program w języku C++, który poda liczbę odcinków w zbiorze Cantora dla rzędu n .
3. Ułóż program w języku Java, który poda liczbę odcinków w zbiorze Cantora dla rzędu n .
4. Ułóż algorytm (w postaci listy kroków), który obliczy długość jednego odcinka w zbiorze Cantora rzędu n .
5. Ułóż program w języku C++, który obliczy długość jednego odcinka w zbiorze Cantora rzędu n .
6. Ułóż program w języku Java, który obliczy długość jednego odcinka w zbiorze Cantora rzędu n .
7. Uruchom program z rysunku 28.2. Zbadaj, od której zmiennej zależy liczba wywołań (rekurencji). Zmodyfikuj program w taki sposób, by kreślił następne dwa poziomy odcinków.

PODSUMOWANIE LEKCJI

- Fraktal jest obiektem samopodobnym, czyli takim, którego części są podobne do całości. Tworzenie fraktali jest zazwyczaj procesem rekurencyjnym. s. 192
- Zbiór Cantora jest podzbiorem prostej rzeczywistej, który umożliwia tworzenie fraktali. Podstawowy zbiór Cantora to nieskończony zbiór punktów odcinka jednostkowego $[0,1]$. s. 193



27. Drzewo binarne, czyli hierarchiczna struktura danych

NA TEJ LEKCJI:

- dowiesz się, czym jest drzewo binarne;
- poznasz zastosowanie drzewa binarnego;
- ułożysz algorytm tworzenia drzewa binarnego.

Wiesz już, że dane można przechowywać w tabelach, tak jak to ma miejsce w bazie danych, np. utworzonej w Access lub Excel. Znasz zalety takiej struktury. Domyślasz się, że nie jest to jedyny sposób przechowywania danych. Jedną z możliwych struktur jest drzewo, a w szczególności drzewo binarne.

27.1. Jeden węzeł i dwoje dzieci, czyli drzewo binarne

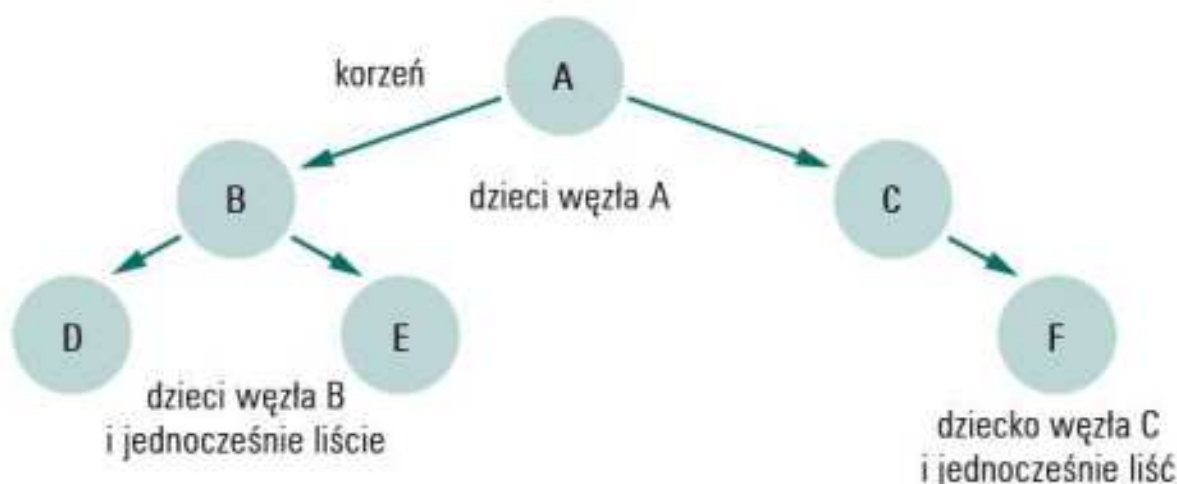
Przypomnij sobie, jak wygląda drzewo genealogiczne. Korzeń tworzy para najstarszych członków rodziny. Ich dzieci tworzą konary itd. Wyobraź sobie teraz analogiczne drzewo z danymi, w którym każdy ojciec ma dwóch synów, ci zaś następnie także stają się ojcami kolejnych **dwóch** itd. W podobny sposób powstaje **drzewo binarne**.

W drzewie binarnym z dowolnego węzła mogą wychodzić maksymalnie dwie gałęzie.

Składa się ono z **węzłów** połączonych krawędziami. Pierwszy nazywamy **korzeniem** (root). Następny węzeł powiązany z wyższym krawędzią (połączeniem) nazywany jest jego **dzieckiem** (lub synem), a węzły, które nie mają dzieci, to **liście** (rys. 27.1).

drzewo binarne

korzeń, węzeł,
dziecko i liść
drzewa binarnego



Rys. 27.1. Przykładowa struktura drzewa binarnego



Drzewo binarne pozwala na przechowywanie danych w sposób zhierarchizowany. Może ono zostać użyte nie tylko do przechowywania danych, lecz także do ich posortowania.

Implementacją (wdrożeniem) drzewa binarnego jest kopiec binarny zawierający liczby. Jest to kompletne drzewo binarne mające warunek, że żadne z dwojga dzieci węzła nie zawiera większej wartości niż ta w węźle. Z tego wniosek, że to jedna z metod sortowania danych.

kopiec binarny

Opis metody tworzenia kopca binarnego:

Dodawany element wstawiamy jako ostatni liść drzewa. Następnie sprawdzamy kolejno, czy jest on mniejszy od swojego „rodzica” lub mu równy. Jeśli nie, to zamieniamy wstawiony element z jego rodzicem. Operację sortowania kontynuujemy, aż znajdziemy rodzica większego od elementu lub mu równego albo dojdziemy do korzenia drzewa.

Algorytm dodawania nowego elementu do kopca binarnego:

Specyfikacja

Dane wejściowe:

Tablica – tablica zawierająca kopiec

ilosc – liczba elementów w kopcu

nowy – wartość dodawanego elementu

Dane wyjściowe:

Tablica – tablica zawierająca kopiec z dodanym nowym elementem

ilosc – liczba elementów w kopcu zwiększona o 1

Zmienne pomocnicze:

i, j – indeksy elementów w kopcu

Krok	Działanie	Opis
01	$i = \text{ilosc}$	ustawiamy indeks i na pozycję wstawianego elementu
02	$\text{ilosc} = \text{ilosc} + 1$	zwiększenie ilości elementów o 1
03	$j = (i - 1) \div 2$	obliczamy indeks rodzica
04	dopóki $i > 0$ oraz $\text{Tablica}[j] < \text{nowy element}$ wykonuj kroki 05..07	sprawdzanie warunku kopca
05	$\text{Tablica}[i] = \text{Tablica}[j]$	umieszczamy rodzica na miejscu syna
06	$i = j$	przenosimy się na pozycję ojca
07	$j \leftarrow (i - 1) \div 2$	obliczamy indeks ojca
08	$\text{Tablica}[i] = \text{nowy}$	wstawiamy element do kopca
09	koniec	

Tab. 27.1. Algorytm dodawania elementu do kopca binarnego



Przykładowa funkcja w języku C++ wstawiająca element do kopca

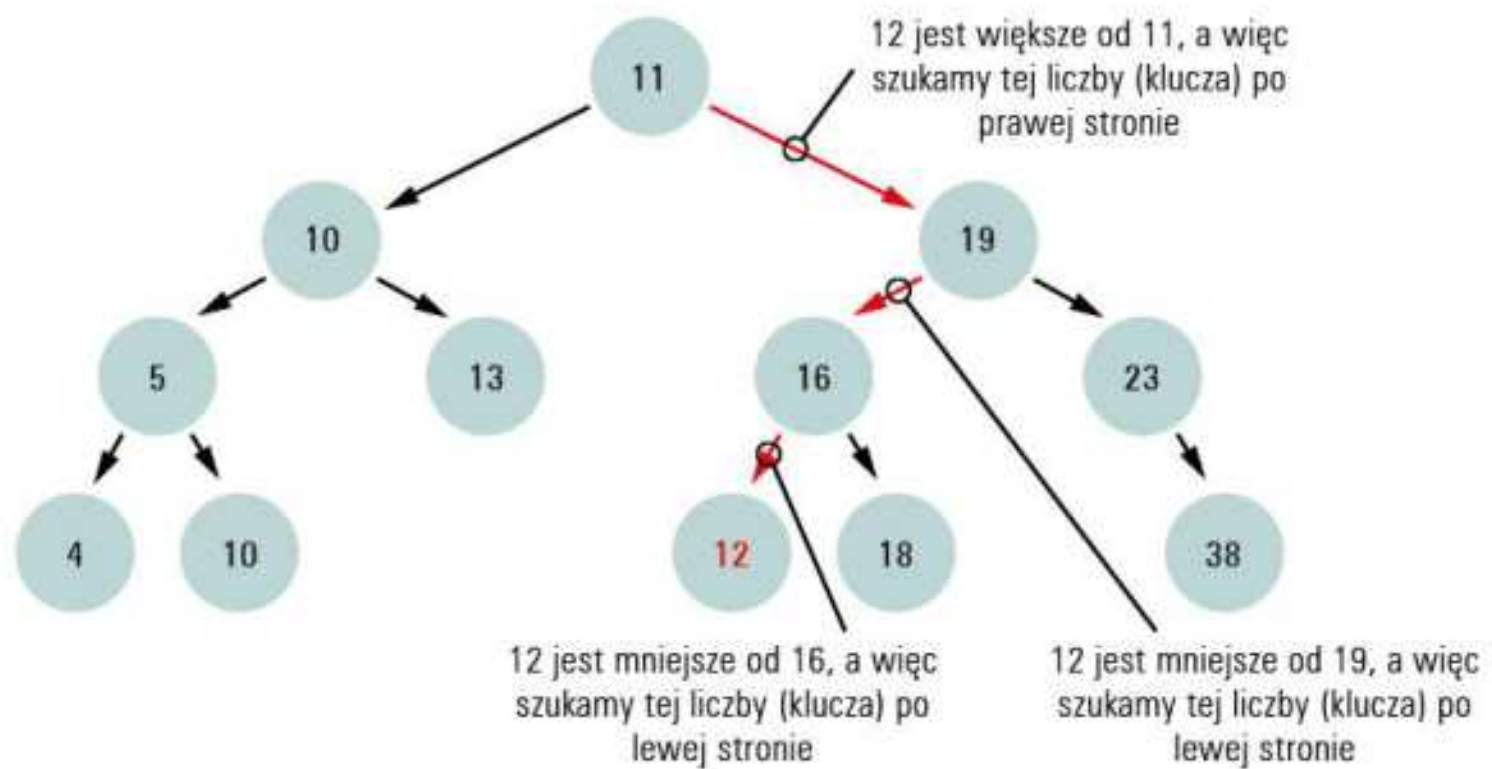
```
void wstaw_element (int nowy)
{
    int i, j;
    i = ilosc++;
    j = ( i - 1 ) / 2;
    while( i > 0 && Tablica[j] < nowy )
    {
        Tablica[i] = Tablica[j];
        i = j;
        j = ( i - 1 ) / 2;
    }
    Tablica[i] = nowy;
}
```

27.2. BST, czyli binarne drzewo poszukiwań

BST

BST (*binary search tree*) to binarne drzewo poszukiwań. Jego cechą charakterystyczną jest to, że spośród dwojga dzieci węzła, to o mniejszej wartości (większym kluczu) znajduje się po lewej stronie, a większe (lub równe) po prawej. Jest to jednak kwestia przyjęcia odpowiedniej konwencji. Można bowiem przyjąć, że będzie odwrotnie. Ważne, by dla całego BST obowiązywała ta sama zasada. Dzięki takiej strukturze możliwe jest skuteczne i szybkie wyszukiwanie danych.

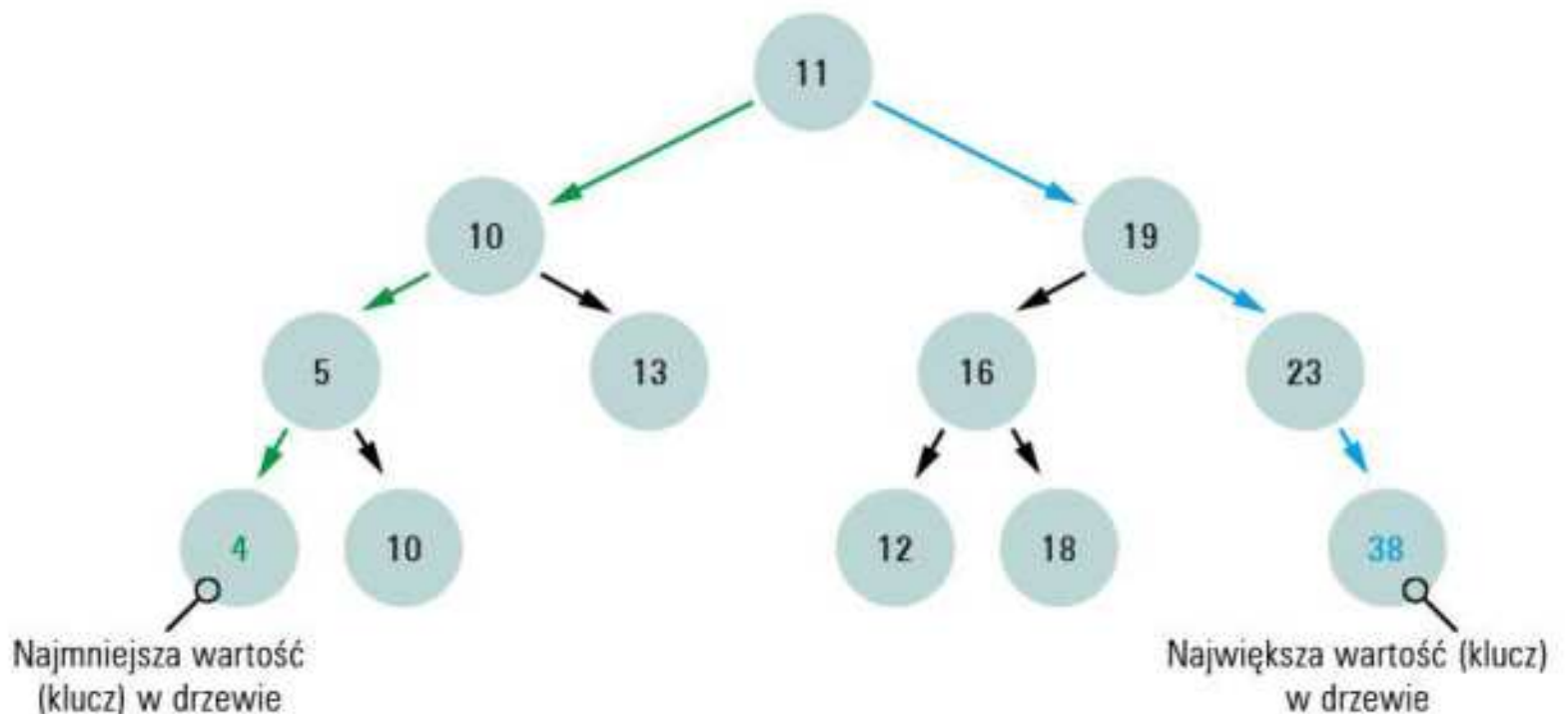
Założmy, że mamy już BST (rys. 27.2.) i większa wartość zawsze była wpisywana po prawej stronie. Problem do rozwiązania polega na wyszukaniu danej, której wartość (klucz) ma określoną wartość, np. 12. Wykorzystamy podstawową cechę binarnego drzewa poszukiwań. Zawsze będziemy poruszać się po drzewie wg zasady: wartość (klucz) większa od tej w węźle – wybieramy prawą gałąź (krawędź), mniejsza – lewą. Postępujemy tak do momentu, gdy wartość węzła będzie równała się szukanej.



Rys. 27.2. Wyszukiwanie wartości w BST

Analizując budowę binarnego drzewa poszukiwań, zauważymy, że łatwo odnaleźć największą lub najmniejszą wartość. Wystarczy poruszać się po „**zewnątrznych konarach**” (rys. 27.3.).

najmniejsza
i największa
wartość w BST



Rys. 27.3. Największa i najmniejsza wartość w BST



ZADANIA DO WYKONANIA

1. Ułóż algorytm w postaci schematu blokowego, który odnajdzie żądaną wartość w BST.
2. Utwórz w dowolnym edytorze lub w zeszycie BST zawierające następujące wartości: 3, 8, 12, 13, 14, 17, 22, 25, 33, 43, 46, 51, 67 i 69. Sprawdź poprawność wykonania zadania, wyszukując największą i najmniejszą wartość.
3. Ułóż algorytm tworzenia BST.
4. Ułóż algorytm umieszczający daną wartość w istniejącym BST z zachowaniem przemieszczenia także innych węzłów.

PODSUMOWANIE LEKCJI

- s. 196 • Drzewo binarne pozwala nam na przechowywanie danych w sposób zhierarchizowany.
- s. 196 • W drzewie binarnym każdy węzeł może mieć co najwyżej dwoje dzieci. Węzły bez dzieci to liście. Korzeń drzewa binarnego to pierwszy jego węzeł.
- s. 197 • Kopiec binarny jest implementacją drzewa binarnego BST.
- s. 198 • BST (*binary search tree*) to binarne drzewo poszukiwań. W zależności od przyjętej konwencji mniejsze wartości są umieszczane po jednej stronie węzła, a większe – po drugiej.
- s. 199 • W celu znalezienia najmniejszej lub największej wartości w BST należy poruszać się po zewnętrznych konarach.



28. Trójkątny fraktal, czyli trójkąt Sierpińskiego

NA TEJ LEKCJI:

- dowiesz się, czym jest trójkątny i kwadratowy dywan Sierpińskiego;
- ułożysz algorytm i program oparty na podejściu rekurencyjnym, tworzącym dywan Sierpińskiego.

Dywan Sierpińskiego opisał jeden z najwybitniejszych polskich matematyków, Wacław Sierpiński. Był on współtwórcą Polskiej Szkoły Matematycznej. Jego największe osiągnięcia dotyczą teorii liczb. Algorytmy tworzenia tych fraktali są bardzo podobne, dlatego w rozdziale zajmiemy się dokładnie jednym z nich – Trójkątem Sierpińskiego.

30.1. Algorytm tworzenia dywanu Sierpińskiego

Kwadratowy dywan Sierpińskiego otrzymamy, dzieląc kwadrat na dziewięć mniejszych kwadratów i usuwając środkowy (tak jakby była to wycinanka). Potem powtarzamy te same czynności dla każdego z pozostałych kwadratów. Jest więc to proces rekurencyjny. Można więc sformułować następujący algorytm.

kwadratowy dywan
Sierpińskiego

Lista kroków:

Krok 1

Rysujemy kwadrat, który dzielimy na dziewięć równych części i usuwamy środkowy kwadrat.

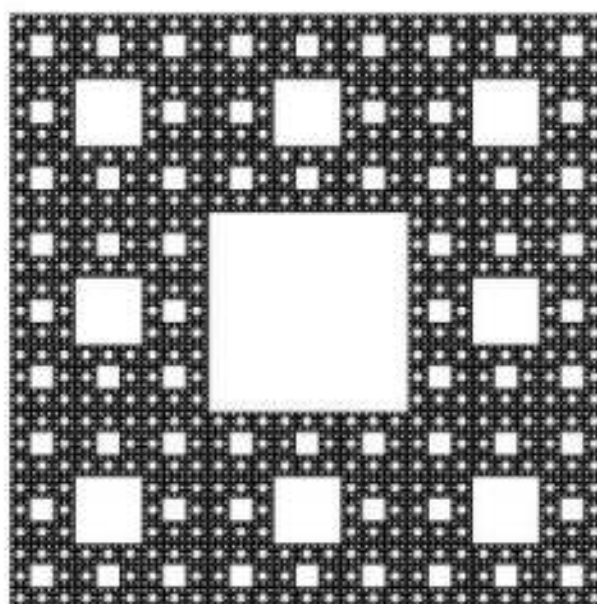
Krok 2

Każdy z pozostałych ośmiu kwadratów dzielimy ponownie na dziewięć części i usuwamy środkowe kwadraciki.

Krok n -ty

W kolejnych krokach postępujemy identycznie jak w krokach pierwszym i drugim. Po n krokach kwadrat będzie miał aż $1 + 8 + 8^2 + 8^3 + \dots + 8^{n-1}$ usuniętych kwadratów.

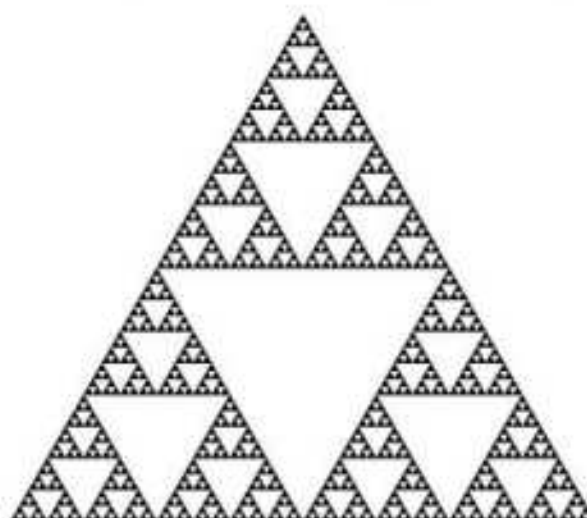
Kwadraty te będą różnej wielkości (rys. 28.1.).



Rys. 28.1. Kwadratowy dywan Sierpińskiego po 6 krokach

trójkątny dywan Sierpińskiego

Oprócz kwadratowego dywanu Sierpińskiego spotkasz się jeszcze z trójkątną jego wersją, którą otrzymamy z trójkąta równobocznego (rys. 28.2.). Wierzchołek każdego nowego trójkąta znajduje się w środku boku trójkąta powstałego w poprzednim kroku (poziomie). Zasady tworzenia obu dywanów są bardzo zbliżone.



Rys. 28.2. Trójkątny dywan Sierpińskiego

28.2. Rekurencja w kreśleniu, czyli program rysuje trójkąt Sierpińskiego

Do ułożenia programu kreślącego trójkąt Sierpińskiego użyjemy tej samej biblioteki i języka, który posłużył nam do wizualizacji zbioru Cantora. Możesz użyć środowiska programistycznego Replit (<https://bezpiecznylink.pl/replit>).

W komentarzach znajdziesz wszystkie informacje potrzebne do zrozumienia działania programu.

**Uwaga!**

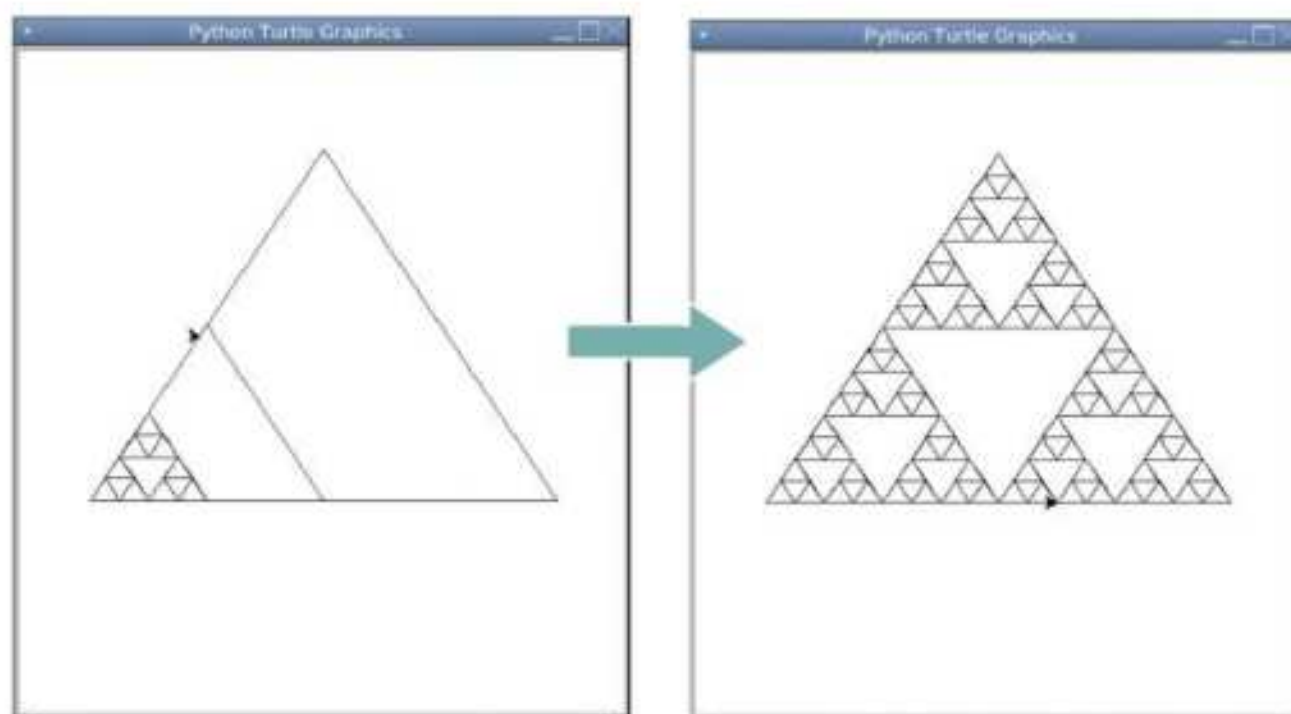
W języku Python wcięcia tworzą bloki w strukturze programu podobnie jak nawiasy klamrowe w C++. Przepisując kod programu z podręcznika, zwróć na to baczną uwagę.

```

main.py
1  #dołączenie biblioteki turtle
2  import turtle
3  t = turtle.Turtle()
4
5  #funkcja rysująca pojedynczy trójkąt
6  def rysujTrojkat(punkty):
7      t.fillcolor("black")
8      t.up()
9      t.goto(punkty[0][0],punkty[0][1])
10     t.down()
11     t.goto(punkty[1][0],punkty[1][1])
12     t.goto(punkty[2][0],punkty[2][1])
13     t.goto(punkty[0][0],punkty[0][1])
14
15     #funkcja obliczania położenia wierzchołka trójkąta
16     def polowa(p1,p2):
17         return ( (p1[0]+p2[0]) / 2, (p1[1] + p2[1]) / 2)
18
19     #funkcja rysowania trójkątów wykorzystująca rekurencję
20     def sierpinski(punkty,liczbaPoziomow):
21         rysujTrojkat(punkty)
22         if liczbaPoziomow > 0:
23             sierpinski([punkty[0], polowa(punkty[0], punkty[1]), polowa(punkty[0], punkty[2])], liczbaPoziomow-1)
24             sierpinski([punkty[1], polowa(punkty[0], punkty[1]), polowa(punkty[1], punkty[2])], liczbaPoziomow-1)
25             sierpinski([punkty[2], polowa(punkty[2], punkty[1]), polowa(punkty[0], punkty[2])], liczbaPoziomow-1)
26
27     #początek programu
28     #aktywacja okna graficznego biblioteki turtle
29     myWin = turtle.Screen()
30     #współrzędne wierzchołków pierwszego trójkąta
31     mojepunkty = [[-200,-100],[0,200],[200,-100]]
32     #wywołanie funkcji rysującej trójkąty z 4 poziomami trójkątów
33     sierpinski(mojepunkty,4)
34     myWin.exitonclick()
35

```

Rys. 28.3. Przykładowy kod programu rysującego trójkąt Sierpińskiego



Rys. 28.4. Efekt pośredni i końcowy działania programu rysującego trójkąt Sierpińskiego



ZADANIA DO WYKONANIA

1. Ułóż algorytm w formie schematu blokowego rysowania trójkątnego dywanu Sierpińskiego.
2. W przykładowym programie liczba rekurencji wynosi 4. Sprawdź, jak będzie wyglądał trójkąt powstały na trójkącie Sierpińskiego dla zmiennej liczbaPoziomów równej 1 oraz 2, 3, 5.
3. W linii 31 przykładowego programu z rysunku 30.3. wprowadzono współrzędne wierzchołków pierwszego trójkąta. Zmień te wartości w taki sposób, by uzyskać możliwie największy trójkąt mieszczący się w oknie Python Turtle Graphics. Sprawdź, dla jakiej wartości zmiennej liczbaPoziomów kolejne trójkąty będą jeszcze wyraźnie widoczne. Od czego jeszcze zależeć będzie wynik obserwacji?
4. Ułóż program kreślący kwadratowy dywan Sierpińskiego.

PODSUMOWANIE LEKCJI

- s. 201 • Dywan Sierpińskiego jest fraktalem, który powstaje na zasadzie rekurencji.
- s. 201 • Kwadratowy dywan Sierpińskiego otrzymamy, dzieląc kwadrat na dziewięć mniejszych kwadratów i usuwając środkowy, i powtarzając tę czynność dowolną ilość razy.
- s. 202 • Trójkątny dywan Sierpińskiego powstaje na bardzo podobnej zasadzie jak jego kwadratowa wersja. Wierzchołki kolejnych trójkątów znajdują się w punktach będących środkami boków trójkątów kreślonych w poprzednim poziomie rekurencji.



29. Krzywa fraktalna, czyli płatek Kocha

NA TEJ LEKCJI:

- dowiesz się, czym jest krzywa fraktalna zwana też krzywą Kocha;
- odkryjesz, jak powstaje płatek Kocha;
- ułożysz program kreślący płatek Kocha.

Przyroda potrafi nas zadziwić wielością form i zjawisk. Czy zwróciliście uwagę na misterne kształty płatków śniegu? Niels Fabian Helge von Koch (czyt. nils fabian helge fon koch) pokusił się o opracowanie matematycznego wzorca stworzenia tego cudu natury.

29.1. Krzywa fraktalna, czyli krzywa Kocha

Krzywa fraktalna to funkcja, za pomocą której jesteśmy w stanie narysować fraktale samopodobne. Krzywa ta może być nieskończenie długa, ale będzie się mieściła na skończonej powierzchni, co umożliwi nam narysowanie jej przybliżenia. Pierwszą osobą, która opisała ten problem, był Niels Fabian Helge von Koch. Dokonał tego w 1904 roku.

krzywa fraktalna

Krzywa Kocha, nazywana również krzywą płatka śniegu, ma kilka interesujących właściwości:

- krzywa ta jest połączona w taki sposób, że nie znajdziemy w niej żadnych przerw, ale nie jest gładka i równa, ponieważ ma nieskończoną liczbę ostrych narożników;
- jeżeli położymy naszą krzywą płatka śniegu pod mikroskopem, będzie wyglądać tak samo, jak gdybyśmy oglądali ją bez powiększenia, czyli jest samopodobna;
- krzywa ta ma nieskończony obwód, ale skończoną powierzchnię.

29.2. Algorytm tworzenia płatka Kocha

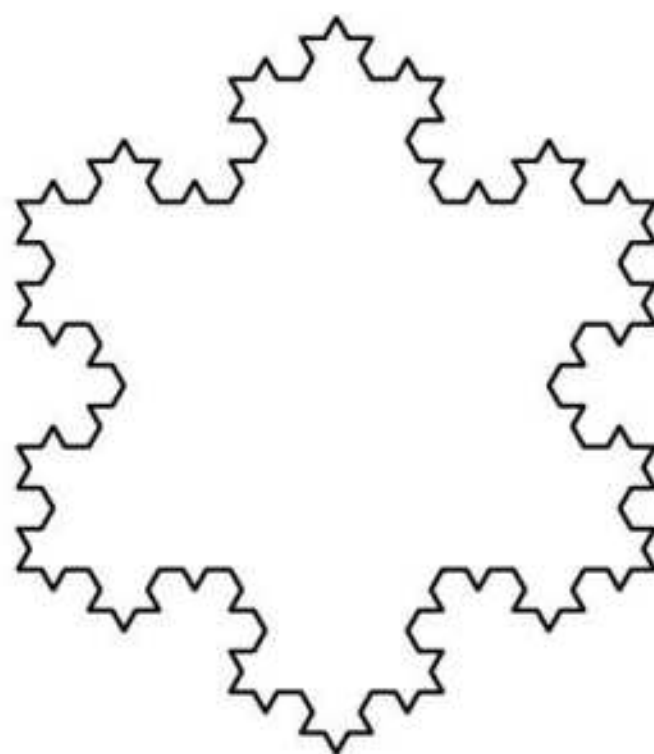
Krzywą Kocha, zwaną również krzywą fraktalną i krzywą płatka śniegu, otrzymamy w wyniku rekurencji wykonywanych wstępnie na odcinku. Powstaje ona poprzez podzielenie go na 3 części i zastąpienie środkowej ząbkami (o ramieniu o długości równej $1/3$ odcinka) takim, że wraz z usuwaną częścią tworzy trójkąt równoboczny. Krok ten jest powtarzany w nieskończoność dla każdego fragmentu odcinka.

Lista kroków dla algorytmu rysowania krzywej Kocha stopnia n o boku a :

1. jeśli $n = 1$, narysuj odcinek o długości a i zatrzymaj wykonywanie,
2. narysuj krzywą Kocha stopnia $n - 1$ o boku $a/3$,



3. obróć się w lewo o 60 stopni,
4. narysuj krzywą Kocha stopnia $n - 1$ o boku $a/3$,
5. obróć się w prawo o 120 stopni,
6. narysuj krzywą Kocha stopnia $n - 1$ o boku $a/3$,
7. obróć się w lewo o 60 stopni,
8. narysuj krzywą Kocha stopnia $n - 1$ o boku $a/3$.



Rys. 29.1. Płatek Kocha

**płatek Kocha
i krzywa Kocha**

Konstrukcję tę możemy powtarzać nieskończoną liczbę razy. Z powodu ograniczonej rozdzielczości ekranu oraz ułomności ludzkiego oka kolejne obrazki nie będą się już wiele różniły od tego, co zobaczyliśmy. Figurę przypominającą płatek śniegu nazywamy płatkem Kocha, a jej brzeg – krzywą Kocha.

Płatek Kocha, chociaż powstał z sumowania nieskończenie wielu trójkątów, ma skończone pole. Jego brzeg jest krzywą o nieskończonej długości ograniczającej obszar o skończonym polu.



29.3. Fraktal na ekranie, czyli program tworzy płatek Kocha

Użyjmy ponownie biblioteki **turtle** dla języka **Python**. Po doświadczeniach z poprzednich rozdziałów zapewne szybko zrozumiesz działanie przykładowego programu (rys. 29.2.).

Uwaga!

W języku Python wcięcia tworzą bloki w strukturze programu podobnie jak nawiasy klamrowe w C++. Przepisując kod programu z podręcznika, zwróć na to baczną uwagę.

```

main.py
1  from turtle import *
2  #import turtle
3  # funkcja rysująca krzywą Kocha
4  def krzywaKocha(dlugoscBok, poziomy):
5      #jeśli poziom 0 to rysujemy odcinek
6      if poziomy == 0:
7          forward(dlugoscBok)
8          return
9      #Obliczanie 1/3 długości odcinka
10     dlugoscBok /= 3.0
11     #kreślenie odcinków i zmiana kierunku kreślenia
12     krzywaKocha(dlugoscBok, poziomy-1)
13     left(60)
14     krzywaKocha(dlugoscBok, poziomy-1)
15     right(120)
16     krzywaKocha(dlugoscBok, poziomy-1)
17     left(60)
18     krzywaKocha(dlugoscBok, poziomy-1)
19 # program główny
20 #prędkość rysowania
21 speed(500)
22 #długość boku
23 dlugosc = 500.0
24 #podniesienie póra
25 penup()
26 #przesunięcie aby wyśrodkować obrazek w oknie
27 backward(dlugosc/2.0)
28 # opuszczenie pióra
29 pendown()
30 #rysowanie płatków
31 for i in range(3):
32     krzywaKocha(dlugosc, 4)
33     right(120)
34 mainloop()
35

```

Rys. 29.2. Przykładowy kod programu kreślący płatek Kocha



ZADANIA DO WYKONANIA

1. Ułóż algorytm w formie blokowej rysowania płatka Kocha.
2. Przeanalizuj program z rysunku 31.2. Ustal, który parametr odpowiada za liczbę trójkątów (ząbków) płatka. Zmień ją i obserwuj efekt.
3. Ułóż algorytm tworzenia płatka Kocha w postaci pseudokodu.

PODSUMOWANIE LEKCJI

- s. 205 • Krzywa Kocha to krzywa fraktalna, która powstaje z odcinka w wyniku podzielenia go na 3 części i zastąpienia środkowej z nich takim ząbkem, który wraz z usuwaną częścią tworzy trójkąt równoboczny.
- s. 206 • Płatek Kocha to figura, która powstaje z sumowania nieskończenie wielu trójkątów równobocznych.



PODSUMOWANIE PRZED SPRAWDZIANEM

20. Przypomnij sobie, czyli wracamy do środowiska programistycznego

- Środowisko programistyczne Eclipse może być wykorzystane do programowania w języku C++. Należy jednak pobrać i zainstalować **odpowiednią wersję, uwzględniając także system operacyjny**, w którym ma być zainstalowana. s. 158
- Do kompilacji C++ w Eclipse są potrzebne narzędzia pozwalające na kompilację programów w środowisku Windows. Jednym z powszechnie stosowanych jest **MinGW**. Należy go zainstalować przed instalacją Eclipse dla C++. s. 158
- Eclipse najlepiej pobrać ze strony domowej. Instalacja przebiega automatycznie. Należy jednak pamiętać o wybraniu odpowiedniej wersji (dla C++) i ścieżki do folderu, w którym będą gromadzone projekty. s. 160
- Eclipse ułatwia stawianie pierwszych kroków w programowaniu w C++ między innymi poprzez przykładowe programy pokazujące poprawną strukturę. Dzięki temu można ćwiczyć kompilację i uruchamianie programów. s. 161
- Dyrektywy oznaczone **#** są wykonywane przed kompilacją i powodują dołączenie odpowiednich bibliotek (zbiorów funkcji). s. 163
- **using namespace std;** to informacja dla kompilatora, że w programie zostanie użyta standardowa przestrzeń **std**. Przy układaniu dużych programów mogą nastąpić konflikty nazw, czyli różne obiekty mogą być nazwane tak samo. Można je umieszczać w różnych przestrzeniach i dzięki temu nie będą ze sobą kolidowały. s. 163
- Program zaczyna działanie od kodu zawartego w funkcji **main()**. s. 163
- **cout<<** należy rozumieć jako „wyprowadź komunikat”. s. 163

21. Przypomnij sobie, czyli podstawy języka programowania

- **Kompilacja** polega na dołączeniu modułów umożliwiających uruchomienie programu w systemie operacyjnym i utworzeniu pliku z kodem maszynowym zrozumiałym dla procesora. s. 165
- **Operatory relacyjne** są używane w instrukcji warunkowej lub pętlach. s. 167
- W pętli **do while** instrukcje są wykonywane **przynajmniej jeden** raz bez względu na prawdziwość warunku. s. 170
- **Funkcja** jest fragmentem programu, który nazwano i który jest wykonywany poprzez podanie jego nazwy oraz ewentualnych argumentów (w nawiasie za nazwą). s. 170

22. Czy litery to liczby, czyli kod ASCII i porównanie tekstów

- Kody ASCII to liczby odpowiadające znakom drukowanym lub wyświetlanym. Każdemu znakowi przyporządkowano jeden bajt. s. 173
- W rzeczywistości powstało wiele kodów ASCII zawierających znaki charakterystyczne dla danego języka, np. polskie **ą, ę, ó, ł** itp. s. 173
- W systemie Windows stosuje się stronę kodową Windows-1250 CE, będącą odpowiednikiem ASCII dla krajów Europy Środkowo-Wschodniej. s. 176



- s. 177 • Wprowadzając odpowiednie kody z klawiatury numerycznej (z lewym Alt), można do tekstu wstawiać znaki i symbole.
- s. 177 • We własnych opracowaniach można cytować fragmenty innych dzieł.
- s. 178 • Każda praca naukowa i dyplomowa na studiach wyższych musi być zweryfikowana przez Jednolity System Antyplagiatowy.

23. Metoda naiwna, czyli szukamy wzorca w tekście

- s. 180 • **Wzorzec występujący** w ciągu znaków jest spójnym jego fragmentem.
- s. 180 • **Idea algorytmu naiwnego wyszukiwania wzorca w tekście** polega na przeglądaniu danego tekstu sekwencyjnie. Przesuwamy wzorzec po kolei po jednym znaku i sprawdzamy, czy kolejne znaki wzorca są identyczne ze znakami tekstu.
- s. 181 • **Zakończenie działania algorytmu metody naiwnej** następuje w momencie, w którym w sprawdzanym tekście pozostaje mniej znaków niż zawiera wzorzec.
- s. 181 • **Algorytm metody naiwnej** szukania wzorca w tekście można zapisać w postaci graficznej.
- s. 182 • **Każdy algorytm należy przetestować** na wielu zestawach danych.

24. Realizacja algorytmu, czyli program szuka wzorca

- s. 184 • Dla uproszczenia możesz przyjąć stałą wielkość tablic `t[]` oraz `w[]`.
- s. 184 • `cout.width` określi, ile znaków łącznie będzie zawierał strumień.
- s. 184 • Dzięki użyciu `cin.getline` będzie możliwe wprowadzenie tekstu lub wzorca do momentu użycia enteru.
- s. 184 • **Białe znaki** są znakami niewidocznymi na ekranie.
- s. 184 • **Sposób wyświetlania** efektu działania programu może wskazywać jednocześnie miejsce w tekście, w którym znaleziono wzorzec.
- s. 185 • **Funkcja `strlen()`** (zawarta w bibliotece `string.h`) oblicza (zwraca) liczbę znaków tablicy przekazanej w nawiasie jako jej argument.
- s. 185 • Przykładowy program ma ograniczenia, np. liczbę znaków w tekście lub wzorcu.

25. Przybliżona wartość, czyli komputer oblicza wartość pierwiastka kwadratowego

- s. 187 • Pierwiastek z danej liczby obliczamy tak, że szukamy liczby, która podniesiona do drugiej potęgi da liczbę pod pierwiastkiem.

26. Zbiór Cantora, czyli najprostsze tworzenie fraktali

- s. 192 • **Fraktal** jest obiektem samopodobnym, czyli takim, którego części są podobne do całości. Tworzenie fraktali jest zazwyczaj procesem rekurencyjnym.
- s. 193 • **Zbiór Cantora** jest podzbiorem prostej rzeczywistej, który umożliwia tworzenie fraktali. Podstawowy zbiór Cantora to nieskończony zbiór punktów odcinka jednostkowego $[0,1]$.



27. Drzewo binarne, czyli hierarchiczna struktura danych

- Drzewo binarne pozwala nam na przechowywanie danych w sposób zhierarchizowany. s. 196
- W drzewie binarnym każdy węzeł może mieć co najwyżej dwoje dzieci. Węzły bez dzieci to liście. Korzeń drzewa binarnego to pierwszy jego węzeł. s. 196
- Kopiec binarny jest implementacją drzewa binarnego BST. s. 197
- BST (*binary search tree*) to binarne drzewo poszukiwań. W zależności od przyjętej konwencji mniejsze wartości są umieszczane po jednej stronie węzła, a większe – po drugiej. s. 198
- W celu znalezienia najmniejszej lub największej wartości w BST należy poruszać się po zewnętrznych konarach. s. 199

28. Trójkątny fraktal, czyli trójkąt Sierpińskiego

- Dywan Sierpińskiego jest fraktalem, który powstaje na zasadzie rekurencji. s. 201
- Kwadratowy dywan Sierpińskiego otrzymamy, dzieląc kwadrat na dziewięć mniejszych kwadratów i usuwając środkowy, i powtarzając tę czynność dowolną ilość razy. s. 201
- Trójkątny dywan Sierpińskiego powstaje na bardzo podobnej zasadzie jak jego kwadratowa wersja. Wierzchołki kolejnych trójkątów znajdują się w punktach będących środkami boków trójkątów kreślonych w poprzednim poziomie rekurencji. s. 202

29. Krzywa fraktalna, czyli płatek Kocha

- Krzywa Kocha to krzywa fraktalna, która powstaje z odcinka w wyniku podzielenia go na 3 części i zastąpienia środkowej z nich takim ząbkiem, który wraz z usuwaną częścią tworzy trójkąt równoboczny. s. 205
- Płatek Kocha to figura, która powstaje z sumowania nieskończenie wielu trójkątów równobocznych. s. 206

IV. Wykorzystanie algorytmów w rozwiązywaniu problemów i programowaniu

Jak to zapisać...

Mniej lub więcej ósemek...

Iloczyn i iloraz oktalny...

Nie tylko dwójkowy i dziesiętny...

Iloczyn i iloraz w systemie innym...

Wyciąganie elementów z listy...

Elementy występujące kolejno...

Ta sama kolejność, czyli szukanie...

Z której strony, czyli położenia...

Algorytmy badające własności...

Skrzyżowanie dróg, czyli badanie ...

Wewnątrz czy na zewnątrz...

Wyznaczanie pola ograniczonego ...

Znajdowanie elementów w zbiorze...

Szukanie połówek...

Ustawianie kart, czyli sortowanie...

Regula falsi, czyli przybliżone...

Geometria obliczeniowa...

Algorytm wyszukujący...

Podejście zachłanne...

Z miasta A do miasta B...

Pakowanie plecaka...

Metoda haszowania...

Metoda Monte Carlo...





30. Jak to zapisać, czyli zamiana systemów liczbowych z ósemkowego na szesnastkowy

NA TEJ LEKCJI:

- przypomnisz sobie, czym są systemy liczbowe ósemkowy i szesnastkowy;
- powtórzysz sobie, jak zapisuje się liczby w tych systemach;
- dowiesz się, jak zamieniać liczby zapisane w różnych systemach.

Codziennie posługujemy się systemem dziesiętnym zwanym inaczej decymalnym. Jest to system pozycyjny, co oznacza, że liczba składa się z cyfr, których wartość zależy od pozycji oraz podstawy systemu. W tym systemie każde 10 jednostek niższego rzędu tworzy 1 jednostkę następnego, wyższego rzędu. Oprócz systemu dziesiętnego w informatyce wykorzystujemy system ósemkowy, zwany oktalnym, oraz szesnastkowy, zwany heksadecymalnym. W tym ostatnim do zapisu oprócz cyfr od 0 do 9 używamy również liter od A do F, które przyjmują wartości w systemie dziesiętnym od 10 do 15.

30.1. System ósemkowy

W informatyce wykorzystujemy również systemy ósemkowy i szesnastkowy, ponieważ bardzo łatwo możemy dokonać przeliczeń między nimi a systemem dwójkowym, który jest uważany za podstawowy.

system ósemkowy

System ósemkowy to pozycyjny system liczbowy, do którego zapisu używamy cyfr od 0 do 7. W informatyce system ten jest wykorzystywany przez system Linux w poleceniu `chmod`, które ustawia prawa dostępu do pliku.

Tak jak w pozostałych systemach, liczby zapisujemy w nim jako ciągi cyfr, z których każda jest mnożnikiem kolejnej potęgi liczby będącej podstawą systemu, np. gdybyśmy chcieli w systemie ósemkowym zapisać liczbę, która w systemie dziesiętnym ma wartość 250, to przybrałaby ona postać 372, ponieważ:

$$3 \cdot 8^2 + 7 \cdot 8^1 + 2 \cdot 8^0 = 192 + 56 + 2 = 250$$

Gdybyśmy chcieli zamienić liczbę zapisaną w systemie dziesiętnym na liczbę zapisaną w systemie ósemkowym, musielibyśmy zastosować następującą listę kroków:

Krok 1

Zapisujemy liczbę w systemie dziesiętnym, np. jako liczbę L.

**Krok 2**

Liczbę L dzielimy przez podstawę systemu docelowego.

Krok 3

Za kolejną od końca cyfrę przyjmujemy resztę z dzielenia.

Krok 4

Za nową liczbę L przyjmujemy wynik dzielenia całkowitoliczbowego.

Krok 5

Jeśli liczba L jest różna od zera, operację powtarzamy. W przeciwnym razie wyświetlamy otrzymane cyfry w kierunku odwrotnym od ich wyznaczania.

Przykład

Mamy liczbę 25 000 zapisaną w systemie dziesiętnym i chcemy ją zamienić na system ósemkowy:

$$25\,000 : 8 = 3125, \text{ reszta } 0$$

$$3125 : 8 = 390, \text{ reszta } 5$$

$$390 : 8 = 48, \text{ reszta } 6$$

$$48 : 8 = 6, \text{ reszta } 0$$

$$6 : 8 = 0, \text{ reszta } 6$$

$$25\,000_{10} = 60\,650_8$$

30.2. System szesnastkowy

Szesnastkowy system liczbowy, nazywany też heksadecymalnym, to również system pozycyjny, który wykorzystuje do swojego zapisu 16 cyfr. Pierwsza dziesiątka jest taka sama, jak w systemie 10, natomiast kolejne 6 cyfr to litery alfabety od A do F, które przyjmują kolejno wartości od 10 do 15.

system
szesnastkowy

W systemie szesnastkowym liczby zapisujemy tak samo, jak w pozostałych pozycyjnych systemach liczbowych – jako ciąg znaków, w którym każdy kolejny z nich jest mnożnikiem kolejnej potęgi stanowiącej podstawę systemu. W tym przypadku będzie to potęga liczby 16.

Mamy liczbę 250 zapisaną w systemie dziesiętnym. Natomiast w systemie szesnastkowym jej zapis to FA, ponieważ:

$$15 \cdot 16^1 + 10 \cdot 16^0 = 240 + 10 = 250$$

Cyfra F w systemie szesnastkowym przyjmuje wartość 15, a cyfra A – wartość 10.

Zasada zamiany liczby dziesiętnej na liczbę szesnastkową jest identyczna jak w przypadku zamiany z liczby dziesiętnej na ósemkową, tylko tutaj podstawą, przez którą będziemy dzielić liczbę dziesiętną, będzie 16.



Przykład:

$$25\,000 : 16 = 1562, \text{ reszty } 8$$

$$1562 : 16 = 97, \text{ reszty } 10 - \text{ w zapisie szesnastkowym będzie to A}$$

$$97 : 16 = 6, \text{ reszty } 1$$

$$6 : 16 = 0, \text{ reszty } 6$$

$$25\,000_{10} = 61A8_{16}$$

30.3. Przeliczanie z systemu ósemkowego na szesnastkowy

W poprzednich rozdziałach poznaliśmy zasady przeliczania z systemu dziesiętnego na ósemkowy, z dziesiętnego na szesnastkowy. Kolejnym poziomem będzie przeliczanie z systemu ósemkowego na system szesnastkowy.

Założmy, że mamy liczbę 255_8 i chcemy ją przeliczyć na system szesnastkowy. Dokonujemy tego w dwóch etapach. Pierwszym będzie przeliczenie naszej liczby na system dziesiętny, a następnie powstałą liczbę w systemie dziesiętnym przeliczymy na system szesnastkowy.

$$255_8 = 2 \cdot 8^2 + 5 \cdot 8^1 + 5 \cdot 8^0 = 128 + 40 + 5 = 173$$

Liczba 255 zapisana w systemie ósemkowym to 173 w systemie dziesiętnym. Teraz liczbę 173 zamienimy na liczbę zapisaną w systemie szesnastkowym.

$$173 : 16 = 10, \text{ reszty } 13 \text{ (czyli D w systemie szesnastkowym)}$$

$$10 : 16 = 0, \text{ reszty } 10 \text{ (czyli A w systemie szesnastkowym)}$$

Liczba 173 w systemie dziesiętnym daje nam liczbę AD w systemie szesnastkowym, czyli liczba 255 zapisana w systemie ósemkowym to AD w systemie dziesiętnym.

Przełożmy to na język programowania.

Ze względu na to, że będziemy pracować na różnych liczbach, będziemy je przechowywać w tablicy. Tym sposobem możemy wprowadzić liczby w systemie szesnastkowym.

Stwórzmy sobie funkcję pomocniczą, która pozwoli w przyszłości wykonywać operacje arytmetyczne.

```
int getIntValue (char p) {  
    if (p >= '0' && p <= '9')  
        return p - '0';  
    if (p >= 'A' && p <= 'F')  
        return 10 + p - 'A';  
    return 0;  
}
```

przeliczanie
między dowolnymi
systemami



Będziemy potrzebować również funkcji, która na podstawie wartości cyfry określi jej reprezentację za pomocą znaku:

```
char getCharValue (int a){
    if (q >= 0 && q <= 9)
        return '0' + q;
    if (q >= 10 && q <= 15)
        return 'A' + q - 10;
    return ' ';
}
```

Skoro mamy potrzebne funkcje, zaczniemy przeliczanie – na początek na system dziesiętny:

```
int anyToDecimal (char* q, int s)
{
    int liczba = 0;
    for (int i = strlen(q) - 1, pot = 1, i >= 0, i--)
    {
        liczba += getIntValue(q[i]*pot);
        pot *= s;
    }
    return liczba;
}
```

Kolejnym etapem będzie przeliczenie liczby z systemu dziesiętnego na dowolny system liczbowy:

```
char* wynik = new char[d + 1];
for (int i = d, pot = 1; i >= 0; i--)
{
    wynik[i] = getCharValue(q % s);
    q /= s;
}
wynik[d] = '\0';
return wynik;
}
```

Na koniec stworzymy funkcję, która przetestuje nam te funkcje, które wcześniej stworzyliśmy:

```
int main () {
    char* q = '63';
    int m = anyToDecimal(q, 16);
    cout << m << endl;
    char* p = decimalToAny(m, 8);
    cout << p << endl;
    system("pause");
    return 0;
}
```




ZADANIA DO WYKONANIA

1. Napisz program w języku Java, który przeliczy zadaną przez użytkownika liczbę z systemu dziesiętnego na dwójkowy.
2. Napisz program w języku Java, który przeliczy zadaną przez użytkownika liczbę z systemu dwójkowego na dziesiętny.
3. Napisz program w języku C++, który przeliczy zadaną przez użytkownika liczbę z systemu dziesiętnego na dwójkowy.
4. Napisz program w języku C++, który przeliczy zadaną przez użytkownika liczbę z systemu dwójkowego na dziesiętny.
5. Dostosuj przedstawione w rozdziale 32.3. funkcje `getIntValue()` i `getCharValue()` tak, aby było można dokonać konwersji liczb z i do systemu dwudziestkowego.

PODSUMOWANIE LEKCJI

- s. 214 • System ósemkowy jest pozycyjnym systemem liczbowym składającym się z cyfr od 0 do 7. W informatyce wykorzystywany jest przez system Linux w poleceniu `chmod`, które ustawia prawa dostępu do pliku.
- s. 215 • System szesnastkowy jest pozycyjnym systemem liczbowym, który składa się z cyfr od 0 do 9 oraz liter od A do F, i w którym A przyjmuje w systemie dziesiętnym wartość 10, a F wartość 15.
- s. 216 • Przeliczenia między dowolnymi systemami możemy dokonać w dwóch podstawowych krokach. Zamieniamy liczbę z dowolnego systemu na system dziesiętny, a następnie z systemu dziesiętnego na kolejny system.



31. Mniej lub więcej ósemek, czyli dodawanie i odejmowanie w systemie ósemkowym

NA TEJ LEKCJI:

- dowiesz się, w jaki sposób dodaje się i odejmuje liczby w systemie ósemkowym.

Dodawanie i odejmowanie to dwa podstawowe działania arytmetyczne. W systemie ósemkowym również możemy wykonywać te działania.

31.1. Dodawanie w systemie ósemkowym

W systemie ósemkowym dodawanie wykonujemy za pomocą działań pisemnych. Dzięki temu będziemy mogli dokładnie zobaczyć, krok po kroku, jakie czynności podejmujemy.

dodawanie
w systemie
ósemkowym

Na początek dodamy do siebie liczbę 26_8 i liczbę 7_8 .

System przesunięcia nadmiaru działa identycznie jak w systemie binarnym, dając 1 tego systemu liczbowego.

$$\begin{array}{r}
 1 \\
 2 \quad 6 \\
 + \quad 7 \\
 \hline
 3 \quad 5
 \end{array}$$

Rys. 31.1. Dodawanie w systemie ósemkowym

Pamiętajmy, że system nadmiaru będzie występował tylko wtedy, gdy suma naszego dodawania przekroczy 7. W pierwszej kolumnie sumujemy liczby 6 i 7, co daje nam 13. Od 13 odejmujemy 8 (ponieważ dodajemy w systemie ósemkowym), więc otrzymamy 5, a nadmiar o wartości 1 przenosimy do kolumny drugiej. W drugiej kolumnie dodajemy 1 do 2, daje nam to 3, więc gdy dodamy w systemie ósemkowym 7 do 26_8 , to otrzymamy liczbę 35_8 .

Wykonajmy przykład kolejny. Zsumujmy w systemie ósemkowym liczby 274 i 337:

$$\begin{array}{r} 1 1 \\ 2 7 4 \\ + 3 3 7 \\ \hline 6 3 3 \end{array}$$

Rys. 31.2. Dodawanie w systemie ósemkowym

W powyższym przykładzie nadmiar występuje dwa razy. W pierwszej kolumnie dodajemy 4 i 7 – otrzymujemy 11, od 11 odejmujemy 8 i otrzymujemy 3, 1 przenosimy do kolumny drugiej. W drugiej kolumnie sumujemy 7 i 3 oraz 1 z nadmiaru – otrzymujemy znowu 11, więc 11 minus 8 daje nam 3, jeden przenosimy do kolumny 3. W kolumnie trzeciej sumujemy 2 i 3 oraz 1 z nadmiaru – otrzymujemy 6. Sumą naszych liczb w systemie ósemkowym będzie liczba 633.

33.2. Odejmowanie w systemie ósemkowym

odejmowanie
w systemie
ósemkowym

Odejmowanie w systemie ósemkowym wykonujemy w identyczny sposób, jak w przypadku dodawania, czyli pisemnie, z tą różnicą, że nie zwracamy nadmiaru, ale bierzemy pożyczkę.

Aby to zobrazować, odejmiemy liczbę 6 od liczby 73. Oczywiście obie liczby są w systemie ósemkowym.

$$\begin{array}{r} 73 \\ - 6 \\ \hline 65 \end{array}$$

Rys. 31.3. Odejmowanie w systemie ósemkowym



Odejmując 3 od 6 w pierwszej kolumnie, musimy wziąć pożyczkę 1 z drugiej kolumny. Wykonujemy działanie odejmowania w systemie ósemkowym, więc dodajemy 8 do liczby 3 – otrzymujemy 11, a następnie odejmujemy 6 i w pierwszej kolumnie mamy wynik 5. Ponieważ z drugiej kolumny braliśmy pożyczkę – 1, więc zostało nam tam tylko 6, które przepisujemy na dół. Wynikiem odejmowania liczby 6 od 73 w systemie ósemkowym jest 65.

Sprawdźmy to na kolejnym przykładzie. Odejmijmy 77 od 555.

$$\begin{array}{r}
 7+1 7+1 \\
 5 5 5 \\
 - 7 7 \\
 \hline
 4 5 6
 \end{array}$$

Rys. 31.4. Odejmowanie w systemie ósemkowym

Odejmując 5 od 7 w pierwszej kolumnie pożyczamy 1 z kolumny drugiej. Zgodnie z zasadą – 8 plus 5 daje nam 13, od tego odejmujemy 7 i otrzymujemy 6. W drugiej kolumnie mamy 4, ponieważ 1 przekazaliśmy do kolumny pierwszej, i 7; aby odjąć te liczby od siebie, musimy wziąć pożyczkę z kolumny trzeciej; 8 plus 4 daje nam 12, od których odejmujemy 7 i otrzymujemy 5. W kolumnie trzeciej pozostaje nam 4, ponieważ 1 pożyczaliśmy do kolumny drugiej. Przepisujemy je na dół. W wyniku odejmowania liczby 77 od 555 w systemie ósemkowym otrzymaliśmy liczbę 456.

31.3. Program dodaje ósemkowo, czyli jak zrealizować algorytm

Przykładowy program (rys. 31.5.) realizujący dodawanie liczb ósemkowych wykorzystuje właściwości liczb i zasady sumowania opisane w tym rozdziale.

Uwaga!

Dla uproszczenia kodu zrezygnowano ze sprawdzania, czy wprowadzona liczba jest rzeczywiście liczbą ósemkową. Program podaje prawidłowe wyniki jedynie dla liczb ósemkowych, czyli składających się z cyfr od 0 do 7.


```

1 #include <iostream>
2 using namespace std;
3 //funkcja dodająca liczby ósemkowe
4 int suma_oktalna(int a, int b)
5 {
6     int suma = 0, liczba = 0, przeniesc = 0, pozycja = 1;
7     while (a > 0 || b > 0 || przeniesc)
8     {
9         liczba = a % 10 + b % 10 + przeniesc;
10        // czy mamy przeniesienie
11        if (liczba > 7)
12        {
13            przeniesc = 1;
14            liczba %= 8;
15        }
16        else
17            przeniesc = 0;
18        suma += liczba * pozycja;
19        pozycja *= 10;
20        a /= 10;
21        b /= 10;
22    }
23    return suma;
24 }
25 int main()
26 {
27     int a,b;
28     cout << "Podaj liczbę ósemkową a : " << endl;
29     cin >> a;
30     cout << "Podaj liczbę ósemkową b : " << endl;
31     cin >> b;
32     cout << suma_oktalna(a, b);
33     return 0;
34 }
35

```

Rys. 31.5. Przykładowy program dodający dwie liczby ósemkowe



ZADANIA DO WYKONANIA

1. Wykonaj dodawanie następujących liczb w systemie ósemkowym: $216 + 173$; $155 + 63$; $707 + 63$.
2. Wykonaj odejmowanie następujących liczb w systemie ósemkowym: $311 - 225$; $202 - 57$; $70 - 12$.
3. Na podstawie wykonanych obliczeń ulóż algorytm w postaci listy kroków realizujący dodawanie w systemie ósemkowym.
4. Na podstawie wykonanych obliczeń ulóż algorytm w postaci listy kroków realizujący odejmowanie w systemie ósemkowym.
5. Przeanalizuj program z rysunku 33.5. i opisz działanie funkcji obliczającej sumę liczb ósemkowych. Sprawdź, czy działa ona podobnie do twojego algorytmu. Opisz ewentualne różnice.
6. Zmodyfikuj program z rysunku 33.5. w taki sposób, by uniemożliwił wprowadzenie liczb zapisanych znakami 8 lub 9.
7. Ulóż program realizujący odejmowanie liczb ósemkowych. Opisz jego działanie. Pamiętaj o podaniu specyfikacji programu.

PODSUMOWANIE LEKCJI

- Dodawanie w systemie ósemkowym wykonujemy identycznie jak w systemie binarnym czy dziesiętnym. W przypadku nadmiaru dokonujemy przeniesienia liczby 1. s. 219
- Odejmowanie w systemie dziesiętnym jest identyczną czynnością jak w systemach binarnym czy dziesiętnym. W przypadku niedoboru pożyczamy liczbę 1. s. 220

32. Iloczyn i iloraz oktalny, czyli mnożenie i dzielenie w systemie ósemkowym

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest system oktalny;
- dowiesz się, w jaki sposób mnoży się i dzieli w systemie ósemkowym.

Jak już wiesz, system ósemkowy, inaczej zwany oktalnym, jest stosowany również w informatyce, lecz rzadziej niż systemy dwójkowy (binarny) i szesnastkowy (heksadecymalny). Wykorzystuje się go zazwyczaj do przypisywania uprawnień w różnych dystrybucjach Linuksa oraz podczas programowania w językach C, C#, C++, Java, Pearl czy PHP.

32.1. Mnożenie w systemie ósemkowym

mnożenie
w systemie
ósemkowym

Zasady i schemat mnożenia liczb w systemie ósemkowym są bardzo zbliżone do mnożenia w systemie dziesiętnym. Musimy jednak pamiętać, że mamy do dyspozycji jedynie 8 cyfr, a nie 10.

Wykonajmy przykładowe działanie, mnożąc liczbę 14_8 przez 5_8 .

Aby zapoznać się z zasadami, jakimi musimy się kierować przy tym działaniu, wykonajmy następującą listę kroków:

Krok 1

Wykonujemy mnożenie dziesiętne – nie musimy dokonywać wcześniej zamiany liczb. Każdą cyfrę mnożnej mnożymy przez każdą cyfrę mnożnika.

Krok 2

Otrzymany wynik mnożenia zamieniamy na system ósemkowy.

Krok 3

Teraz zamienione na system ósemkowy liczby zapisujemy tak samo jak podczas mnożenia w systemie dziesiętnym, aby je zsumować.

Krok 4

Sumujemy liczby według zasad dodawania w systemie oktalnym.

1	4	
·		5
2	4	
+	5	
7	4	

$4 \cdot 5 = 20_{10} = 24_8$
 $1 \cdot 5 = 5_{10} = 5_8$

Rys. 32.1. Mnożenie w systemie oktalnym



Tabliczka mnożenia podczas mnożenia liczb w systemie ósemkowym będzie wyglądała inaczej niż podczas mnożenia w systemie dziesiętnym.

	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7
2	2	4	6	10	12	14	16
3	3	6	11	14	17	22	25
4	4	10	14	20	24	30	34
5	5	12	17	24	31	36	43
6	6	14	22	30	36	44	52
7	7	16	25	34	43	52	61

Rys. 32.2. Tabliczka mnożenia w systemie oktalnym

32.2. Dzielenie w systemie ósemkowym

Dzielenie w systemie ósemkowym, tak jak mnożenie, jest prawie identyczne jak w systemie dziesiętnym. Tutaj również musimy pamiętać, że działania wykonujemy na zestawie ośmiu cyfr – od 0 do 7.

dzielenie
w systemie
ósemkowym

Podzielmy 317_8 przez 14_8 .

$$\begin{array}{r}
 \\
 \hline
 3 7 : 1 \\
 - 3 \\
 \hline
 7 \\
 - 4 \\
 \hline
 R:
 \end{array}$$

$$\begin{aligned}
 31_8 &= 25_{10} \\
 14_8 &= 12_{10} \\
 12_{10} \cdot 2 &= 24_{10} \\
 24_{10} &= 30_8
 \end{aligned}$$

Rys. 32.3. Dzielenie w systemie oktalnym

Należy zwrócić uwagę, że nie możemy wykonać bezpośrednio mnożenia liczby 14_8 przez 2_{10} , ponieważ w systemie ósemkowym wyjdzie nam liczba 28_8 ; będzie się ona różniła od przekonwertowanych wartości i pomnożonych przez 2_8 , czyli 30_8 .

Musimy pamiętać o tym, aby przekonwertować poszczególne części dzielenia i mnożenia z liczb w systemie ósemkowym na liczby w systemie dziesiętnym. Następnie wykonujemy potrzebne działania oraz ponowną konwersję do systemu ósemkowego. Na koniec wykonujemy działania dodawania lub odejmowania. Jeżeli nie dokonamy tych konwersji, wynik wyjdzie nam błędny. Wynika to z różnic w liczbie cyfr w danym systemie.



32.3. Program dzieli i mnoży, czyli realizacja algorytmu w języku C++

Poniżej znajdziesz przykładowe kody programów w języku C++ zamieniające liczby z systemu ósemkowego na dziesiętny i odwrotnie. Można je wykorzystać do ułożenia programu mnożącego lub dzielącego liczby ósemkowe, zakładając, że wprowadzone liczby ósemkowe zostaną zamienione na postać dziesiętną, pomnożone lub podzielone, a wynik przedstawiony w postaci liczb ósemkowych.

Uwaga!

Funkcja `int main()` w przykładach z funkcjami konwersji została użyta w celu sprawdzenia poprawności działania konwersji.

Przykład kodu programu konwertującego liczby z postaci ósemkowej do dziesiętnej:

```
long octaDoDzies(long n)
{
    int reszta;
    long dziesiętna=0, i=0;
    while(n!= 0)
    {
        reszta = n%10;
        n = n/10;
        dziesiętna=dziesiętna+(reszta*pow(8,i));
        ++i;
    }
    return dziesiętna;
}

int main()
{
    long a;
    cout << "podaj liczbę ósemkową";
    cin >>a;
    cout << octaDoDzies(a);
}
```

Przykład kodu programu konwertującego liczby z postaci dziesiętnej do ósemkowej:



```

long dziesDoOkta(long n)
{
    int reszta;
    long oktalna=0, i=1;
    while(n!=0)
    {
        reszta = n%8;
        n=n/8;
        oktalna=oktalna+(reszta*i);
        i=i*10;
    }
    return oktalna;
}

int main()
{
    long a;
    cout << "podaj liczbę dziesiętną ";
    cin >>a;
    cout << dziesDoOkta(a);
}

```

Najtrudniejszy problem rozwiązany – pozostaje już tylko ułożenie programu wykorzystującego opisane funkcje do mnożenia lub dzielenia liczb ósemkowych.

Przykładowy kod programu mnożącego liczby ósemkowe:

```

#include <iostream>
#include <cmath>
using namespace std;
//Zamiana postaci dziesiętnej na ósemkową
long dziesDoOkta(long n)
{
    int reszta;
    long oktalna=0, i=1;
    while(n!=0)
    {
        reszta = n%8;
        n=n/8;

```




```
        oktalna=oktalna+(reszta*i);
        i=i*10;
    }
    return oktalna;
}
//Zamiana postaci ósemkowej na dziesiętną
long octaDoDzies(long n)
{
    int reszta;
    long dziesiętna=0, i=0;
    while(n!= 0)
    {
        reszta = n%10;
        n = n/10;
        dziesiętna=dziesiętna+(reszta*pow(8,i));
        ++i;
    }
    return dziesiętna;
}
//Funkcja wykonująca mnożenie liczb ósemkowych
long mnozenie(int c,int d)
{
    long iloczyn;
    iloczyn=dziesDoOkta(octaDoDzies(c)*octaDoDzies(d));
    return iloczyn;
}
int main()
{
    long a,b;
    cout<<"podaj pierwszą liczbę w kodzie ósemkowym ";
    cin>>a;
    cout<<"podaj drugą liczbę w kodzie ósemkowym ";
    cin>>b;
    cout<<mnozenie(a,b);
}
```



ZADANIA DO WYKONANIA

1. Wykonaj mnożenie następujących liczb w systemie ósemkowym: 250 i 3; 55 i 6; 502 i 3.
2. Wykonaj dzielenie następujących liczb w systemie ósemkowym: 371 przez 22; 252 przez 57; 170 przez 15.
3. Napisz algorytm w postaci listy kroków realizujący mnożenie w systemie ósemkowym.
4. Napisz algorytm w postaci blokowej realizujący mnożenie w systemie ósemkowym.
5. Zmodyfikuj program mnożący liczby ósemkowe w taki sposób, by niemożliwe było wprowadzanie liczb zawierających cyfry 8 i 9. Opracuj sposób informowania operatora o takim błędzie.
6. Na podstawie programu z rozdziału ulóż program dzielący liczby ósemkowe. Opisz problemy, jakie można spotkać przy rozwiązywaniu tego zagadnienia.
7. Używając przykładowych funkcji z rozdziału, napisz prosty kalkulator w C++ umożliwiający mnożenie, dzielenie, dodawanie i odejmowanie dwóch liczb ósemkowych.

PODSUMOWANIE LEKCJI

- Mnożenie w systemie ósemkowym jest identyczne jak w systemie dziesiętnym, z tą różnicą, że mamy do dyspozycji tylko osiem cyfr. s. 224
- Dzielenie w systemie ósemkowym jest czynnością identyczną jak w systemie dziesiętnym. Podczas wykonywania dzielenia musimy pamiętać o kolejności działań i konwersji liczb między systemami. s. 225

33. Nie tylko dwójkowy i dziesiętny, czyli suma i różnica w systemie innym niż dziesiętny

NA TEJ LEKCJI:

- przypomnisz sobie, czym są liczby w systemie pozycyjnym o danej zasadzie (podstawie);
- dowiesz się, w jaki sposób sumuje się i odejmuje w różnych systemach pozycyjnych.

Systemy pozycyjne to nic innego, jak metoda zapisywania liczb w taki sposób, aby w zależności od pozycji danej cyfry w ciągu oznaczała ona wielokrotność potęgi liczby uznawanej za bazę systemu, którym się aktualnie posługujemy.

33.1. Zapis liczb naturalnych w systemie pozycyjnym o dodatniej podstawie

system pozycyjny
liczb

W systemie pozycyjnym liczb każda pozycja zapisu liczby ma swoją określoną i niezmienną wagę liczbową. System pozycyjny liczb daje nam również możliwość zapisu ułamków. Poznaliśmy system binarny, oktalny, decymalny oraz heksadecymalny i wiemy, że systemy pozycyjne mają pojedyncze symbole, których używa się do ich zapisu. W zależności od systemu mogą to być same cyfry lub cyfry i litery alfabetu.

Najbardziej znanym systemem pozycyjnym jest system dziesiętny, w którym za bazę przyjmuje się liczbę 10. Tym samym liczbę 35 902 można przedstawić jako:

$$35902 = 3 \cdot 10^4 + 5 \cdot 10^3 + 9 \cdot 10^2 + 0 \cdot 10^1 + 2 \cdot 10^0$$

Zaletą systemów pozycyjnych jest ich przejrzystość i łatwość dokonywania złożonych operacji arytmetycznych. Poza tym dzięki różnym systemom liczbowym mamy również możliwość zapisu dowolnie dużej liczby.

33.2. Dodawanie liczb naturalnych zapisanych w dowolnym systemie pozycyjnym

Działania na liczbach zapisanych w różnych systemach liczbowych wykonuje się podobnie jak w przypadku działań w systemie dziesiętnym. Na początku zajmijmy się dodawaniem.

dodawanie
w systemach
pozycyjnych

Dodajemy liczby cyfry $a_0 + b_0$. Jeżeli $a_0 + b_0 < g$, to jako c_0 przyjmiemy $a_0 + b_0$. W przeciwnym wypadku, czyli gdy $a_0 + b_0 \geq g$, odejmujemy od $a_0 + b_0$ całkowitą



gdyż $6 - 3 = 3$

$1 < 2$, więc bierzemy jednostkę z poprzedniej kolumny $11_7 - 2 = 7 + 1 - 2 = 6$. W następnej kolumnie od 1 odejmujemy 4, toteż musimy wziąć jednostkę z kolumny poprzedniej i otrzymamy $11_7 - 4 = 8 - 4 = 4$.

ZADANIA DO WYKONANIA

- Wykonaj następujące działania:
 - dodawanie liczb w systemie trójkowym: $201102 + 2012$; $22101 + 2020$; $2002221 + 20220$;
 - dodawanie liczb w systemie szóstkowym: $5201 + 3242$; $2035 + 441$; $402332 + 23554$;
 - dodawanie w systemie czternastkowym: $DDA01 + ABC$; $98DD + AA2$; $782A1 + 258C$.
- Wykonaj następujące działania:
 - odejmowanie w systemie czwórkowym: $3202 - 223$; $20330 - 1023$; $22022 - 1233$;
 - odejmowanie w systemie szóstkowym: $5233201 - 254102$; $552412 - 3214$; $333521 - 2542$;
 - odejmowanie w systemie siódmkowym: $652335 - 5421$; $552365 - 21254$; $55445 - 555$.
- Napisz algorytm w postaci listy kroków realizujący dodawanie liczb w dowolnych systemach pozycyjnych.
- Napisz algorytm w postaci blokowej realizujący dodawanie liczb w dowolnych systemach pozycyjnych.
- Napisz program w języku C++, który będzie sumował dwie liczby rzeczywiste w dowolnym systemie pozycyjnym.
- Napisz algorytm w postaci listy kroków, który będzie odejmował dwie liczby rzeczywiste w dowolnym systemie pozycyjnym. Pamiętaj, aby zastrzec w algorytmie, że odejmujemy liczbę mniejszą od większej.
- Napisz algorytm w postaci blokowej, który będzie odejmował liczby rzeczywiste w dowolnym systemie pozycyjnym. Pamiętaj, aby zastrzec w algorytmie, że odejmujemy liczbę mniejszą od większej.
- Napisz program w języku Java, który będzie odejmował od siebie dwie liczby rzeczywiste w dowolnym systemie pozycyjnym. W przypadku odjęcia liczby większej od mniejszej w programie ma się pojawić komunikat: „Odjemna powinna być większa od odjemnika”.

PODSUMOWANIE LEKCJI

- s. 230 • System pozycyjny liczb jest takim systemem zapisu liczb, w którym każda pozycja zapisu liczby ma swoją określoną i niezmienną wagę liczbową.
- s. 230 • Dodawanie w pozycyjnym systemie liczb wykonujemy identycznie jak w systemie dziesiętnym. W przypadku nadmiaru dokonujemy przesunięcia do kolumny następnej.
- s. 231 • Odejmowanie w pozycyjnym systemie liczb wykonujemy podobnie jak w systemie dziesiętnym. Musimy jednak pamiętać, że gdy cyfra odjemnej jest mniejsza niż odpowiednia cyfra odjemnika, bierzemy jednostkę z poprzedniego miejsca.



34. Iloczyn i iloraz w systemie innym niż dziesiętny, czyli wielokrotności w systemach pozycyjnych

NA TEJ LEKCJI:

- dowiesz się, w jaki sposób mnoży się i dzieli w różnych systemach pozycyjnych.

Działania mnożenia i dzielenia są kolejnymi działaniami algebraicznymi wykonywanymi w systemie liczb dziesiętnych, które poznajemy, ucząc się matematyki. Na informatyce nasza wiedza poszerza się o umiejętność wykonywania tych działań w kolejnych systemach. Wiemy już, jak mnożyć i dzielić liczby w systemach binarnym, oktalnym i heksadecymalnym. W tym rozdziale poznamy ogólne zasady wykonywania tych działań w liczbowych systemach pozycyjnych.

34.1. Mnożenie w liczbowych systemach pozycyjnych

Mnożenie w liczbowych systemach pozycyjnych, tak samo jak w przypadku dodawania czy odejmowania, wykonuje się podobnie jak w systemie dziesiętnym. Musimy tylko pamiętać o tym, ile dany system zawiera liczb. Przed rozpoczęciem działań powinniśmy sobie przygotować tabelę mnożenia w takim systemie. Tabliczka mnożenia w systemie siódemkowym będzie się prezentowała następująco:

mnożenie
w liczbowych
systemach
pozycyjnych

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	2	4	6	11	13	15
3	3	6	12	15	21	24
4	4	11	15	22	26	33
5	5	13	21	26	34	42
6	6	15	24	33	42	51

Rys. 34.1. Tabliczka mnożenia w systemie siódemkowym

					1	2	4	3
.					2	1	2	5
	3	0	1	1	1	4	0	1

Mnożymy najpierw kolejne liczby przez 5 (korzystając z piątego wiersza tabliczki mnożenia). Otrzymamy następujący wynik:

$$\begin{array}{r} 21 \\ 26 \\ 13 \\ + 5 \\ \hline 6611 \end{array}$$

Wynik zapisujemy pod kreską mnożenia liczb 1243 przez 2125. Następnie powtarzamy czynność z kolejnymi liczbami. Mnożymy przez 2 – uzyskamy liczbę 25 16 i znów zapisujemy ją pod kreską zgodnie z zasadami mnożenia pisemnego, przesuwając o jedną pozycję w lewo. Gdy wymnożymy wszystkie liczby, przechodzimy do ich zsumowania.

[illegible]

234



34.2. Dzielenie liczb naturalnych zapisanych w dowolnym systemie pozycyjnym

Dzielenie, tak samo jak pozostałe działania arytmetyczne, wykonujemy w sposób bardzo zbliżony do dzielenia wykonywanego w systemie dziesiętnym. Musimy pamiętać, w jakim systemie liczbowym tę czynność wykonujemy, i brać pod uwagę dostępny zakres cyfr w danym systemie.

dzielenie
w liczbowych
systemach
pozycyjnych

Podzielmy liczbę 612654 przez 12 w systemie siódemkowym:

$$\begin{array}{r}
 455 \\
 \hline
 6164 : 1 \\
 5 \\
 \hline
 12 \\
 6 \\
 \hline
 6 \\
 \\
 \hline
 5 \\
 4 \\
 \hline
 1 \\
 0 \\
 \hline
 6
 \end{array}$$

Rys. 34.5. Dzielenie w systemie siódemkowym

Powyższe dzielenie wykonamy w następujących krokach:

Krok 1

Sprawdzamy (wykorzystując tabliczkę mnożenia w systemie siódemkowym z podrozdziału 34.1.), ile razy liczba 12 zmieści się w liczbie 61:

$1 \cdot 12 = 12$, $2 \cdot 12 = 24$, $3 \cdot 12 = 36$, $4 \cdot 12 = 4 \cdot 10 + 4 \cdot 2 = 40 + 11 = 51$, $5 \cdot 12 = 50 + 13 = 63$, czyli 12 w 61 mieści się cztery razy. Pierwszą cyfrą naszego ilorazu będzie 4.

Krok 2

Odejmujemy 51 od pierwszych cyfr dzielnej i do różnicy dopisujemy następną cyfrę z dzielnej – otrzymujemy 102.

**Krok 3**

Sprawdzamy – korzystając znowu z tabliczki mnożenia w systemie siódemkowym – ile razy 12 mieści się w 102:

$6 \cdot 12 = 60 + 15 = 105$, czyli pięć razy. Zapisujemy 5 jako drugą cyfrę naszego ilorazu.

Krok 4

Sprawdzamy, ile razy 12 mieści się w liczbie 66. Z poprzednich obliczeń wiemy, że pięć razy. Ponownie zapisujemy 5 jako kolejną liczbę ilorazu.

Kroki powtarzamy aż do zakończenia dzielenia. Uzyskujemy wynik 45526 i reszty 6.

ZADANIA DO WYKONANIA

- Wykonaj następujące działania:
 - mnożenie liczb w systemie piątkowym:
 $2043 \cdot 412$; $3011 \cdot 321$; $33202 \cdot 2213$;
 - mnożenie liczb w systemie szóstkowym:
 $5241 \cdot 242$; $2045 \cdot 431$; $45332 \cdot 554$;
 - mnożenie w systemie dziewiątkowym:
 $8801 \cdot 125$; $75885 \cdot 236$; $6685 \cdot 254$.
- Wykonaj następujące działania:
 - dzielenie w systemie czwórkowym:
2230102 przez 23; 2033011 przez 13; 3122022 przez 32;
 - dzielenie w systemie piątkowym:
243241 przez 24; 442412 przez 32; 333401 przez 42;
 - dzielenie w systemie siódemkowym:
652635 przez 21; 555365 przez 54; 5544665 przez 15.
- Napisz algorytm w postaci listy kroków realizujący mnożenie liczb w dowolnych systemach pozycyjnych. Załóż, że dane wejściowe to liczby całkowite. Zasugeruj odpowiedni typ zmiennych dla C++.
- Napisz algorytm w postaci blokowej realizujący dzielenie liczb w dowolnych systemach pozycyjnych.

PODSUMOWANIE LEKCJI

- s. 233** • Mnożenie w liczbowych systemach pozycyjnych, tak samo jak w przypadku dodawania czy odejmowania, wykonuje się podobnie jak w systemie dziesiętnym. Musimy tylko pamiętać o tym, ile dany system zawiera liczb.
- s. 235** • Dzielenie, tak samo jak pozostałe działania arytmetyczne, wykonujemy w sposób bardzo zbliżony do dzielenia wykonywanego w systemie dziesiętnym. Tutaj również musimy pamiętać, w jakim systemie liczbowym tę czynność wykonujemy, i brać pod uwagę dostępny zakres cyfr w danym systemie.



35. Wyciąganie elementów z listy, czyli znajdowanie najdłuższego spójnego podciągu niemalejącego

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest ciąg i podciąg;
- dowiesz się, czym jest podciąg spójny, ciąg i podciąg rosnący oraz ciąg i podciąg niemalejący.

Wykrywanie zmian wprowadzanych np. w dokumentach tekstowych, plikach, kodach programów staje się łatwiejsze, gdy znasz metody wykrywania najdłuższego wspólnego podciągu (NWP). Może to także być pomocne w automatyzacji wykrywania plagiatów. Jak znajdować takie podciągi? To wyzwanie, które podejmiesz, czytając ten rozdział.

35.1. Najdłuższy podciąg rosnący

Z matematycznego punktu widzenia ciąg to przyporządkowanie wszystkim liczbom naturalnym z przedziału $[1, n]$ lub wszystkim dodatnim liczbom naturalnym elementów pewnego ustalonego zbioru. W pierwszym przypadku będzie to ciąg skończony, natomiast w drugim – nieskończony. Każdy ciąg możemy podzielić na mniejsze części i będziemy wtedy posługiwać się pojęciem podciągu. Ciągi możemy również nazywać łańcuchami i podłańcuchami.

Podciąg powstaje wtedy, gdy wybierzemy pewną liczbę (skończoną lub nieskończoną) elementów ciągu źródłowego, zachowując ich pierwotną kolejność występowania. Muszą one zachować kolejność występowania, np. w przypadku ciągu $-3, -2, -1, 0, 1, 2, 3$, podciągiem będzie $1, 2, 3$. Oczywiście ciąg nie musi się składać z liczb po kolei występujących po sobie.

Jeżeli elementy będą występowały obok siebie, będziemy wtedy taki podciąg nazywać podciągiem spójnym. W naszym ciągu wyjściowym podciągiem spójnym będą liczby $-2, -1, 0$, jak również wcześniej prezentowane $1, 2, 3$.

Z podciągiem rosnącym mamy do czynienia wtedy, gdy jego kolejny element jest większy od bezpośrednio go poprzedzającego, np. w ciągu $3, 2, 1, -2, 3, 5, 7$ podciągiem rosnącym będzie część $-2, 3, 5, 7$.

Najdłuższy podciąg rosnący – LIS (*longest increasing sequence*) – jest ciągiem, który powstał przez wybranie pewnej liczby elementów ciągu początkowego i jest on najdłuższy w całym ciągu.

Przyjmijmy, że mamy ciąg liczb a_n . Jeżeli z tego ciągu liczby wybierzemy niektóre z nich i ustawimy je tak, by występowały jedna po drugiej, to otrzymamy podciąg:

ciąg

podciąg

podciąg spójny

podciąg rosnący

Ciąg liczb $a_n = 3, 7, 5, 6, 5, 4, 1, 2, 2, 3, 6, 7, 9, 2, 0$

Podciągami ciągu a_n będą:

7, 5,

7, 6, 5, 1, 2, 2,

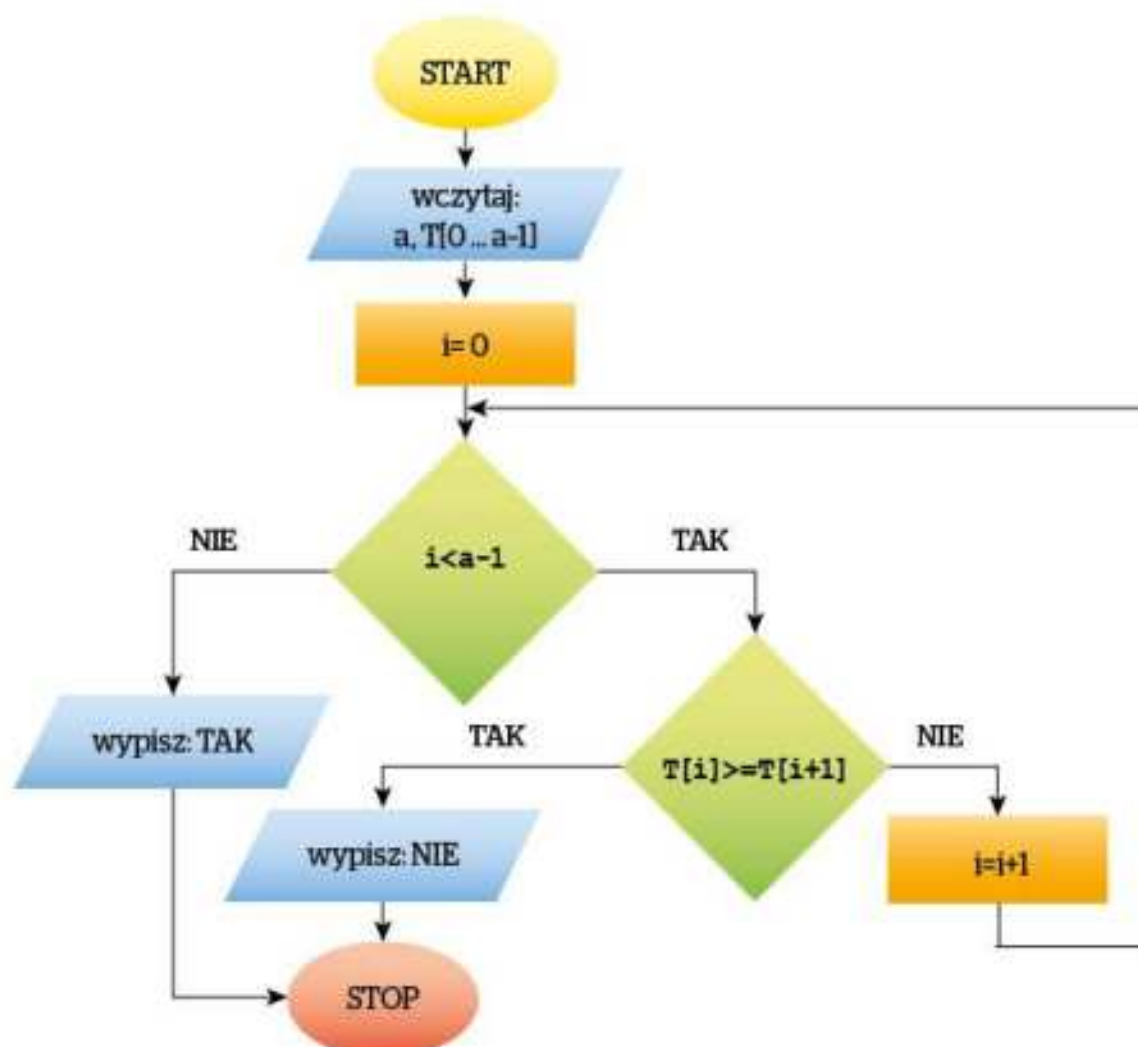
7, 7, 2, 0,

5, 5, 2, 2, 9, 2, 0

Jeżeli każda para sąsiadujących ze sobą liczb będzie spełniała warunek $b_i < b_{i+1}$, to podciąg b_n będziemy nazywali podciągiem rosnącym. Powyższy ciąg zawiera kilka podciągów rosnących, z których najdłuższy będzie następujący: 1, 2, 3, 6, 7, 9. Najdłuższym spójnym podciągiem rosnącym będzie natomiast 2, 3, 6, 7, 9.

Podsumowując, dla skończonego ciągu liczbowego $A = a_1 a_2 \dots a_n$ należy znaleźć podciąg $a_{i_1} a_{i_2} \dots a_{i_n}$ o maksymalnej długości $1 \leq i_1 < i_2 < \dots < i_n$ oraz $a_{i_1} < a_{i_2} < \dots < a_{i_n}$.

Przyjmijmy, że utworzyliśmy ciąg pięcioelementowy a_1, a_2, a_3, a_4, a_5 . Gdyby najdłuższy rosnący podciąg miał się rozpoczynać wyrazem ostatnim a_5 , to jego maksymalna długość wynosiłaby 1. Gdyby najdłuższy rosnący podciąg miał się rozpoczynać wyrazem a_4 , to mógłby mieć maksymalną długość 2, o ile $a_4 < a_5$ – jeśli tak nie jest, to maksymalna długość podciągu rosnącego, jaki możemy uzyskać, rozpoczynając go w tym miejscu, wynosi 1. Jeśli tę samą zasadę zastosujemy do wyrazu a_3 , to tylko nieznacznie utrudni nam szukanie najdłuższego podciągu rosnącego, ponieważ gdyby to a_3 miało być początkiem szukanego podciągu, to wówczas najdłuższy rosnący podciąg będzie o jeden dłuższy od najdłuższego rosnącego podciągu zaczynającego się od a_4 lub a_5 , pod warunkiem, że są one większe od a_3 .



Rys. 35.1. Schemat blokowy algorytmu sprawdzającego, czy ciąg jest rosnący



35.2. Najdłuższy spójny podciąg niemalejący

podciąg
niemalejący

Podciąg niemalejący będzie występował w ciągu, gdy jego kolejne elementy będą większe niż element poprzedzający bądź jemu równe – np. w ciągu 3, 2, 1, -2, -2, 3, 3, 4 podciągiem niemalejącym będzie część -2, -2, 3, 3, 4.

Najdłuższy spójny podciąg niemalejący jest odmianą problemu podciągu rosnącego. O ile w poprzednim wypadku szukany podciąg musiał być ściśle rosnący, to teraz wymagane jest, aby nasz podciąg był niemalejący.

Dla skończonego ciągu liczbowego $A = a_1 a_2 \dots a_n$ musimy znaleźć podciąg $a_{i_1} a_{i_2} \dots a_{i_n}$ o maksymalnej długości, taki, że $1 \leq i_1 < i_2 < \dots < i_n \leq n$ oraz $a_{i_1} \leq a_{i_2} \leq \dots \leq a_{i_n}$.

Aby znaleźć najdłuższy spójny podciąg niemalejący, wystarczy, że wyznaczymy sobie jego pierwszy element. Kolejnych elementów szukamy tak długo, aż będą spełniały warunek, że element je poprzedzający jest od nich mniejszy lub równy.

Algorytm

Wejście

$T[]$ – tablica przechowująca dane

Zmienne pomocnicze

n – długość tablicy $T[]$

max_dl – maksymalna długość ciągu

$poczatek$ – początek szukanego ciągu

dl, p – zmienne pomocnicze

Krok 1 wczytujemy tablicę $T[]$ n -znakową

Krok 2 dla $i < n$ wykonuj kroki Krok03..Krok05

Krok 3 jeżeli $T[i] \geq T[i-1]$ (sprawdzenie, czy następny jest większy od lub równy poprzedniemu) to $++dl$

Krok 4 jeżeli $dl > max_dl$ (sprawdzenie, czy aktualna długość rosnącego ciągu) to $max_dl = dl; poczatek = p$

Krok 5 jeżeli $dl \leq max_dl$, to $p = i; dl = 1$

Krok 6 od $i = poczatek$ do $i < poczatek + max_dl$ wyświetl $T[i]$

Krok 7 koniec



ZADANIA DO WYKONANIA

1. Napisz program w języku C++ wyznaczający najdłuższy podciąg rosnący.
2. Napisz program w języku Java wyznaczający najdłuższy podciąg rosnący z ciągu składającego się z 7 elementów.
3. Napisz algorytm w formie listy kroków wyznaczający najdłuższy spójny podciąg niemalejący.
4. Napisz program w języku C++ wypisujący najdłuższy spójny ciąg niemalejący.
5. Napisz program w języku Java wypisujący najdłuższy spójny ciąg niemalejący.

PODSUMOWANIE LEKCJI

- s. 237 • Ciąg to przyporządkowanie wszystkim liczbom naturalnym z przedziału $[1, n]$ lub wszystkim dodatnim liczbom naturalnym elementów pewnego ustalonego zbioru. W pierwszym przypadku będzie to ciąg skończony, a w drugim – nieskończony.
- s. 237 • Podciąg powstaje wtedy, gdy wybierzemy pewną liczbę (skończoną lub nieskończoną) elementów ciągu wyjściowego, zachowując ich pierwotną kolejność występowania.
- s. 237 • Spójność podciagu zachodzi wtedy, gdy elementy podciagu występują kolejno po sobie.
- s. 237 • Z podciągiem rosnącym mamy do czynienia wtedy, gdy jego kolejny element jest większy od bezpośrednio go poprzedzającego.
- s. 239 • Podciąg niemalejący będzie występował w ciągu, gdy jego kolejne elementy będą większe niż element poprzedzający bądź jemu równe.



36. Elementy występujące kolejno po sobie, czyli znajdowanie najdłuższego spójnego podciągu o największej sumie

NA TEJ LEKCJI:

- dowiesz się, czym jest podciąg o największej (maksymalnej) sumie;
- ułożysz programy komputerowe wyszukujące NSP, wykorzystując trzy różne metody.

W poprzednim rozdziale zapoznałeś/zapoznałaś się z algorytmami wyszukiwania NSP. Kontynuujmy te rozważania i ułożymy na ich podstawie programy. Użyjemy sum prefiksowych oraz metody dziel i zwyciężaj oraz naiwnej. Do dzieła.

36.1. Spójny podciąg o maksymalnej sumie

Spójny podciąg o maksymalnej sumie to taki fragment ciągu, którego suma jest największa. Jaki będzie zatem sposób na wyliczenie sumy ze spójnego podciągu o największej sumie?

spójny podciąg
o maksymalnej
sumie

Przyjmijmy, że dla ciągu n liczb mamy znaleźć podciąg o maksymalnej sumie.

Jednym z rozwiązań będzie rozwiązanie naiwne, w którym możemy znaleźć wszystkie podciągi, obliczyć ich sumy i na koniec wypisać największą z nich. Jak rozwiązać ten problem? Bierzemy wszystkie możliwe pary pozycji, które wyznaczają nam końce podciągów. Dla każdego z nich wyliczymy sumę i wypiszemy największą z nich.

Złożoność takiego podejścia do problemu wynosi $O(n^3)$.

Uwaga! W programach rozwiązujących problem użyto przykładowej tablicy z ciągiem liczb $\{4, -10, 3, 5, -3, 10, -2, 4\}$. Wynikiem działania programów powinna być suma równa 17.

Specyfikacja dla wszystkich przykładów:

$tab[]$ – tablica zawierająca przykładowy ciąg liczb $\{4, -10, 3, 5, -3, 10, -2, 4\}$

t – liczba elementów tablicy (ciągu) $tab[]$

$temp$ – suma tymczasowa

max – suma maksymalna

i, j, k – zmienne pomocnicze



```
#include <iostream>
using namespace std;
int main()
{
    int i, j, k;
    int temp, max = 0;
    int tab[] = {4, -10, 3, 5, -3, 10, -2, 4}, t=8;
    for(i=0; i<t; ++i)
        for(j=i; j<t; ++j)
        {
            for(temp=0, k=i; k<=j; ++k)
                temp+=tab[k];
            if(temp>max)
                max=temp;
        }
    cout << "Maksymalna suma: " << max;
}
```

Innym sposobem będzie rozwiązanie bardziej efektywne, dzięki któremu będziemy szybciej zliczać sumy podciągów. Skorzystamy tutaj z sum prefiksowych.

sumy prefiksowe

Sumy prefiksowe to sumy wszystkich wyrazów ciągu od jego początku, np. dla ciągu 1, 2, 3 sumy prefiksowe to 1, 3, 6.

Sumy prefiksowe obliczymy w czasie liniowym (iterując do tablicy), jeśli dodamy nowy element do poprzedniej sumy prefiksowej.

W taki oto sposób możemy obliczyć sumę dowolnego podciągu o końcach a i b w czasie stałym. Bierzemy element $T[b]$ i odejmujemy od niego $T[a-1]$ (jeżeli a będzie równe 0, w takim przypadku nic nie odejmujemy). $T[b]$ jest sumą wyrazów od początku do pozycji b . Musimy zatem pozbyć się sumy elementów przed pozycją a , więc $T[a-1]$.

Złożoność takiego podejścia do problemu wynosi $O(n^2)$.

```
#include <iostream>
using namespace std;
int main()
{
    int i, j;
    int temp, max=0;
    int tab[]={4, -10, 3, 5, -3, 10, -2, 4}, t=8;
    for(i=0; i<t; ++i)
    {
        if(i>0)
            tab[i]+=tab[i-1];
    }
```



```

for(i=0; i<t; ++i)
    for(j=i; j<t; ++j)
    {
        temp=tab[j];
        if(i>0)
            temp-=tab[i];
        if(temp>max)
            max=temp;
    }
cout << "Maksymalna suma: " << (max);
}

```

Inną efektywną metodą będzie rozwiązanie bardzo podobne do tego, z którym już się zapoznaliśmy. W poniższym sposobie również będziemy zliczać sumy w czasie stałym, ale będziemy to wykonywać na bieżąco.

Przyjmujemy, że nasz podciąg będziemy sprawdzać w następującej kolejności (dla $n = 4$, indeksujemy od 0): 0-0, 0-1, 0-2, 0-3, 1-1, 1-2, 1-3, 2-2, 2-3, 3-3. Dla pierwszej pary potrzebujemy tylko pierwszego elementu. Następnie dla dwóch pierwszych – dwóch elementów, potem trzech itd. Dzięki temu będziemy mogli na bieżąco dodawać tylko jeden element do sumy i gdy przejdziemy wszystkie podciągi dla danej pozycji startowej, będziemy zerować naszą sumę.

Złożoność tego problemu również wynosi $O(n^2)$.

```

int main()
{
    int t, i, j, k, x, max = 0, T[t];

    get(t);
    for(i = 0; i < t; ++i)
        get(T[i]);
    for(i = 0; i < t; ++i)
        for(x = 0, j = i; j < t; ++j) // zerujemy sumę
        {
            x += T[j]; // dodajemy najnowszy element do
sumy
            if(x > max)
                max = x;
        }
    print(max);
}

```




ZADANIA DO WYKONANIA

1. Zmodyfikuj kody programów z rozdziału, tak aby tablica była wprowadzana przez użytkownika i program obliczał długość tablicy (zmienna t).
2. Dany jest ciąg liczb o długości co najmniej 1, składający się z następujących elementów: 7, -12, 8, 5, -3, 28, 5, 3. Znajdź spójny podciąg o największej sumie elementów i wypisz jej wartość. Suma wszystkich elementów to 41. W celu rozwiązania zadania napisz program w języku Java.
3. Napisz program w języku Java, który znajdzie spójny podciąg o największej sumie elementów, i wypisz jej wartość. Ciąg składa się tylko z liczb całkowitych.
4. Napisz program wyszukiujący spójny podciąg o sumie, która nie będzie wyższa od 35. Ciąg składa się z całkowitych liczb ujemnych i dodatnich.
5. Napisz program wyszukiujący spójny podciąg o sumie, która nie będzie wyższa od wprowadzonej przez użytkownika. Ciąg składa się z całkowitych liczb ujemnych i dodatnich.

PODSUMOWANIE LEKCJI

- s. 241 • Spójny podciąg o maksymalnej sumie to taki fragment ciągu, którego suma jest największa.
- s. 242 • Sumy prefiksowe to sumy wszystkich wyrazów ciągu od jego początku – np. dla ciągu 1, 2, 3 sumy prefiksowe to 1, 3, 6. Sumy prefiksowe obliczymy w czasie liniowym (iterując do tablicy), jeśli dodamy nowy element do poprzedniej sumy prefiksowej.



37. Ta sama kolejność, czyli szukanie najdłuższego wspólnego podciągu

NA TEJ LEKCJI:

- dowiesz się, czym jest programowanie dynamiczne;
- poznasz zasady szukania najdłuższego wspólnego podciągu;
- odkryjesz dwie metody wyszukiwania najdłuższego podciągu: rekurencyjną oraz programowania dynamicznego.

Obecnie wszelką dokumentację, ze względów ekologicznych oraz dla oszczędności czasu, prowadzimy w formie elektronicznej. Dokumenty w takiej postaci łatwiej jest nam modyfikować lub znajdować w nich potrzebne informacje.

Przeglądając plik w formacie pdf lub na stronie WWW, korzystamy z kombinacji klawiszy CTRL+F i wyszukujemy interesujące nas informacje. Czy zastanawialiście się, w jaki sposób – za pomocą jakiego algorytmu – to się dzieje? Poniższy rozdział przybliży wam to zagadnienie.

37.1. Programowanie dynamiczne

Programowanie dynamiczne jest metodą projektowania algorytmów w taki sposób, aby problemy były rozwiązywane w sposób optymalny. Stanowi ono alternatywę do rozwiązywania zagadnień za pomocą algorytmów zachłannych. Wynalazcą tej techniki programowania był amerykański matematyk Richard Bellman (wym. riczard belman).

Programowanie dynamiczne opiera się na podziale rozwiązywanego zagadnienia na problemy ze względu na kilka parametrów. W odróżnieniu od metody „dziel i zwyciężaj”, problemy, na które podzielimy nasze zagadnienie, są spójne. Aby zaprojektować optymalny algorytm programowania dynamicznego, należy znaleźć równanie rekurencyjne opisujące optymalną wartość funkcji celu dla problemu, który chcemy rozwiązać za pomocą programowania dynamicznego. Optymalna funkcja celu powinna rozwiązywać problemy o mniejszych rozmiarach zawarte w naszym zagadnieniu.

programowanie
dynamiczne

37.2. Najdłuższy wspólny podciąg (NWP)

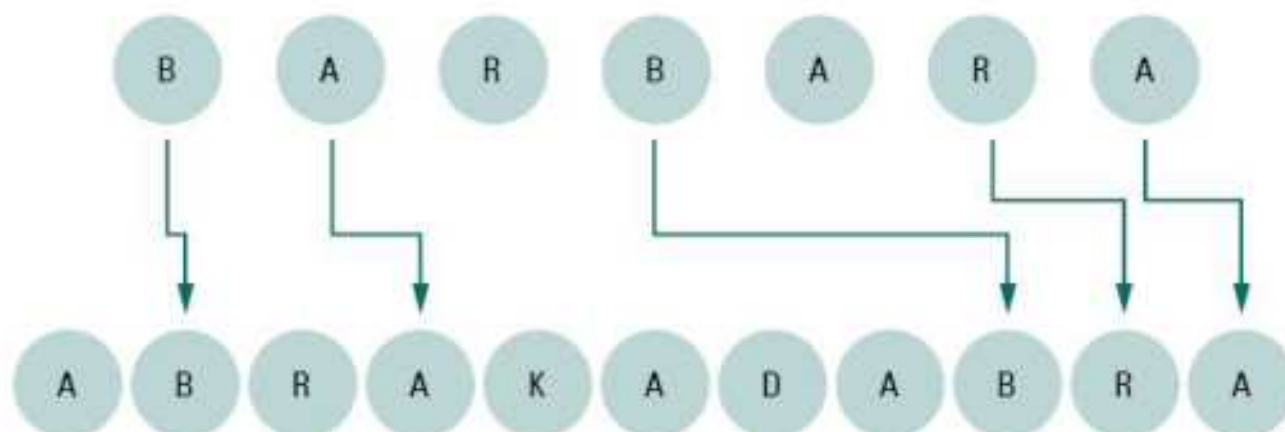
Najdłuższy wspólny podciąg (NWP; ang. *longest common subsequence*, LCS) to najdłuższy podciąg znaków, które występują w tej samej kolejności w dwóch porównywanych ze sobą łańcuchach. Elementy podciągów nie muszą występować obok siebie.

najdłuższy
wspólny podciąg



Algorytm, dzięki któremu wyznaczymy wspólny ciąg dwóch wyrazów, jest bardzo dobrym przykładem programowania dynamicznego.

Będziemy porównywać dwa łańcuchy: BARBARA i ABRAKADABRA. Połączymy strzałkami te znaki, które będą należały do najdłuższego wspólnego podciągu (NWP).



Rys. 37.1. Najdłuższy wspólny podciąg

Gdy szukamy w ten sposób wspólnego najdłuższego podciągu, istotne jest to, aby strzałki się nie krzyżowały, ponieważ nie byłaby zachowana kolejność znaków w przeszukiwanych ciągach. Jeżeli nasze ciągi zaczynają się od tych samych znaków, to znak ten będzie należał do NWP, natomiast jeżeli będą zaczynały się od różnych znaków, to oba początkowe znaki nie będą należały do NWP, ponieważ wówczas strzałki by się skrzyżowały i nie byłaby zachowana kolejność przeszukiwanych znaków. Jeżeli żaden z początkowych znaków się nie powtarza w naszych ciągach, usuwamy je – nasz problem skróci się do znalezienia najdłuższego wspólnego podciągu, który będzie pomniejszony o pierwsze znaki. Takie podejście do sprawy pozwoli nam rozwiązać ten problem za pomocą rekurencji.

Lista kroków rekurencyjnego znajdowania długości najdłuższego wspólnego podciągu NWP:

Długość NWP $\in \mathbb{N}$

a_1, a_2 – są ciągami, w których będziemy szukać NWP – na ich końcu dodamy wartownika

i – indeks w a_1 , od którego zaczniemy przeglądanie naszego łańcucha – $i \in \mathbb{N}$

j – indeks a_2 , od którego zaczniemy przeglądanie naszego łańcucha – $j \in \mathbb{N}$

Krok 1 jeżeli $a_1[i] = \text{wartownik}$, to zakończ z wynikiem 0

Krok 2 jeżeli $a_2[j] = \text{wartownik}$, to zakończ z wynikiem 0

Krok 3 jeżeli $a_1[i] = a_2[j]$, to zakończ z wynikiem $1 + \text{NWP}(i+1, j+1)$

Krok 4 zakończ z wynikiem $\max(\text{NWP}(i+1, j), \text{NWP}(i, j+1))$

Jeżeli będziemy mieli algorytm znajdowania najdłuższego wspólnego podciągu (NWP), to możemy stworzyć algorytm samego NWP. W tym przypadku oprzemy się na następujących zależnościach: jeżeli znaki $a_1[i]$ oraz $a_2[j]$ będą równe, to należą one do NWP, oraz jeżeli znaki $a_1[i]$ oraz $a_2[j]$ będą różne, to wyliczamy $\text{NWP}(i+1, j)$ oraz $\text{NWP}(i, j+1)$, a następnie sprawdzamy:



$NWP(i+1, j) \leq NWP(i, j+1)$ – odrzucamy $a_2[j]$, ponieważ bez niego będziemy mieli dłuższy NWP, w przeciwnym wypadku odrzucamy $a_1[i]$, z tego samego powodu – bez niego będziemy mieli dłuższy NWP.

Lista kroków znajdowania najdłuższego wspólnego podciągu:

a_1, a_2 – ciągi, w których będziemy szukać najdłuższego wspólnego podciągu (NWP) – na końcu umieszczamy wartownika o kodzie 0

i – indeks w a_1 , od którego zaczniemy przeglądanie naszego łańcucha – $i \in N$

j – indeks a_2 , od którego zaczniemy przeglądanie naszego łańcucha – $j \in N$

a_{NWP} – łańcuch, który zawiera NWP

Krok 1 $a_{NWP} \leftarrow ""$ //zerujemy NWP

Krok 2 $i \leftarrow 0$

Krok 3 $j \leftarrow 0$

Krok 4 dopóki $(a_1[i] \neq 0) \wedge (a_2[j] \neq 0)$: wykonaj kroki od Krok 5 do Krok 11

Krok 5 jeżeli $a_1[i] \neq a_2[j]$, to idź do Krok 10

Krok 6 $a_{NWP} \leftarrow a_{NWP} + a_1[i]$

Krok 7 $i \leftarrow i + 1$

Krok 8 $j \leftarrow j + 1$

Krok 9 wróć do Krok 4 i kontynuuj pętlę

Krok 10 jeżeli $NWP(i+1, j) \leq NWP(i, j+1)$, to idź do Krok 8

Krok 11 $i \leftarrow i+1$

Krok 12 zakończ

37.3. Najdłuższy wspólny podciąg (NWP) – algorytm dynamiczny

Stosując algorytm dynamiczny do wyznaczenia najdłuższego wspólnego podciągu, eliminujemy wywołania rekurencyjne, a wyliczone długości NWP zapamiętywane są w tablicach $T[(m+1) \times (n+1)]$. Element tablicy $T[i+1, j+1]$ będzie zawierać długość NWP dla podłańcuchów $a_1[0:i]$ oraz $a_2[0:j]$.

Lista kroków dynamicznego znajdowania długości najdłuższego wspólnego podciągu:

a_1, a_2 – są ciągami, w których będziemy szukać NWP – na ich końcu dodamy wartownika

m – długość ciągu a_1 , $m \in N$

n – długość ciągu a_2 , $n \in N$

T – tablica długości o indeksach od 0 do m oraz od 0 do n , $T \in N$

i, j – indeksy znaków w a_1 i a_2 – $i, j \in N$

Krok 1 $m \leftarrow |a_1|$

Krok 2 $n \leftarrow |a_2|$

Krok 3 dla $i = 0, 1, \dots, m$: wykonaj $T[i, 0] \leftarrow 0$

Krok 4 dla $j = 0, 1, \dots, n$: wykonaj $T[0, j] \leftarrow 0$

Krok 5 dla $i = 0, 1, \dots, m-1$: wykonuj kroki od Krok 6 do Krok 10

Krok 6 dla $j = 0, 1, \dots, n-1$: wykonuj kroki od Krok 7 do Krok 10

Krok 7 jeżeli $a_1[i] \neq a_2[j]$, to idź do Krok 10



Krok 8 $T[i+1, j+1] \leftarrow 1 + T[i, j]$
 Krok 9 wróć do Krok 6 i kontynuuj pętlę
 Krok 10 $T[i+1, j+1] \leftarrow \max(T[i+1, j], T[i, j+1])$
 Krok 11 zakończ z wynikiem $T[m, n]$

Następnym etapem będzie znalezienie najdłuższego wspólnego podciągu.

Lista kroków znajdowania najdłuższego wspólnego podciągu:

a_1, a_2 – są ciągami, w których będziemy szukać NWP

a_{NWP} – ciąg zawierający NWP

T – tablica długości NWP o indeksach od 0 do m oraz od 0 do n

i, j – indeksy znaków a_1 i a_2

Krok 1 utwórz i oblicz tablicę T

Krok 2 $a_{NWP} \leftarrow ""$ //zerujemy NWP

Krok 3 $i \leftarrow m - 1$

Krok 4 $j \leftarrow n - 1$

Krok 5 dopóki $(i \geq 0) > (j \geq 0)$, wykonuj kroki od Krok 6 do Krok 12

Krok 6 jeżeli $a_1[i] \neq a_2[j]$, to idź do Krok 11

Krok 7 $a_{NWP} \leftarrow a_1[i] + a_{NWP}$

Krok 8 $i \leftarrow i - 1$

Krok 9 $j \leftarrow j - 1$

Krok 10 wróć do Krok 5 i kontynuuj pętlę

Krok 11 jeżeli $T[i+1, j] > T[i, j+1]$, to idź do Krok 9

Krok 12 $i \leftarrow i - 1$

Krok 13 zakończ z wynikiem a_{NWP}

37.4. Program szuka najdłuższego wspólnego podciągu

Poniższy przykład programu reprezentuje podejście rekurencyjne.

Specyfikacja programu:

a_1, a_2 – zmienne przechowujące ciągi znaków

a_{NWP} – zmienna przechowująca najdłuższy wspólny podciąg

NWP – funkcja znajdująca najdłuższy wspólny podciąg

i, j – zmienne pomocnicze

```
#include <iostream>
#include <string>
using namespace std;
string a1, a2, aNWP;
int NWP (int i, int j)
{
    if(!a1[i]||!a2[j])
        return 0;
    else
        if(a1[i]==a2[j])
```



```

        return 1+NWP(i+1, j+1);
    else
        return max(NWP(i+1,j), NWP(i,j+1));
    }
int main( )
{
    int i, j;
    cout << "Podaj pierwszy ciąg znaków a1 :\n" ;
    cin >> a1;
    cout << "Podaj drugi ciąg znaków a2 :\n" ;
    cin >> a2;
    aNWP=""; i=j=0;
    while(a1[i]&&a2[j])
        if(a1[i]==a2[j])
        {
            aNWP+=a1[i]; i++; j++;
        }
        else if(NWP(i+1,j)<=NWP(i,j+1)) j++; else i++;
        cout<<"Najdłuższy wspólny podciąg: "<<aNWP<<": "<<aNWP.
length( )<<endl;
    return 0;
}

```

ZADANIA DO WYKONANIA

1. Zakładając, że a_1 i a_2 nie będą dłuższe niż 20 znaków, zmodyfikuj program z rozdziału w taki sposób, by w odpowiedzi wyświetlał a_1 i a_2 , a pod nimi wspólny podciąg dokładnie pod jego położeniem w a_1 i a_2 .
2. Przeanalizuj program z rozdziału i udowodnij tezę, że zastosowano podejście rekurencyjne.
3. Napisz program wyszukujący najdłuższy wspólny podciąg, wykorzystując algorytm dynamiczny.

PODSUMOWANIE LEKCJI

- Programowanie dynamiczne jest metodą projektowania algorytmów w taki sposób, aby problemy były rozwiązywane w sposób optymalny. Stanowi ono alternatywę do rozwiązywania zagadnień za pomocą algorytmów zachłannych. s. 245
- Najdłuższy wspólny podciąg (NWP) jest to najdłuższy podciąg znaków, które występują w tej samej kolejności w dwóch porównywanych ze sobą łańcuchach. Elementy podciągów nie muszą występować obok siebie. s. 245



38. Z której strony, czyli położenia punktów względem prostej

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest prosta, punkt i układ współrzędnych;
- poćwiczysz opisywanie prostych i punktów w geometrii dwuwymiarowej;
- dowiesz się, czym jest wzajemne położenie punktów i prostej oraz jak się je bada.

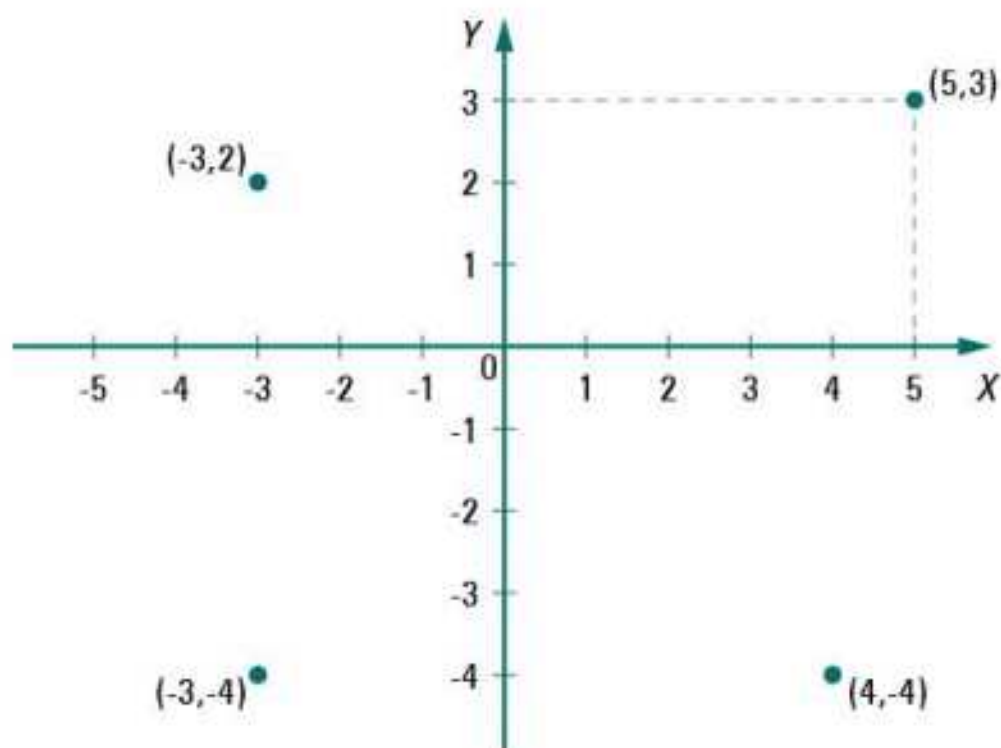
Badanie położenia punktów względem prostej było już omawiane na lekcjach matematyki. Aby zająć się tym problem od strony informatycznej – zbudować potrzebne algorytmy i napisać odpowiednie programy – należy przypomnieć sobie podstawowe zagadnienia, takie jak prosta, współrzędne i punkty.

38.1. Badanie położenia punktów względem prostej

Aby zbadać położenie punktów względem prostej, musimy zagłębić się w zagadnienia geometrii dwuwymiarowej, która opisuje relacje między punktami na płaszczyźnie. Relacje te najprościej można zobrazować za pomocą układu współrzędnych, na którym rozmieszczone są punkty. Punkt w układzie współrzędnych jest miejscem, które jest określone przez parę liczb (x, y) zwane współrzędnymi. Wartość x określa prostopadłe położenie naszego punktu względem osi X , a wartość y natomiast określa prostopadłe położenie względem osi Y .

Prosta to jedno z podstawowych pojęć geometrii. Jest nieograniczona z obu stron. Możemy ją opisać za pomocą równania w postaci kierunkowej $y = ax + b$. Punkty na płaszczyźnie dwuwymiarowej mogą mieć różne położenie względem prostej – mogą znajdować się poza nią lub na niej. Jeżeli będziemy rozpatrywać położenie dwóch punktów względem prostej, mogą one mieć te same współrzędne, czyli jeden będzie nachodził na drugi ($P1 = P2$), albo mogą mieć różne współrzędne.

układ
współrzędnych
i punkt



Rys. 38.1. Układ współrzędnych i punkty

Napişmy prosty program w języku C++, który będzie sprawdzał, czy dany punkt leży na prostej, czy poza nią.

Mamy daną prostą określoną równaniem $y = ax + b$, w którym a i b są liczbami rzeczywistymi. Naszym zadaniem jest sprawdzenie, czy punkt $P(x_p, y_p)$ leży na prostej. Aby to zrobić, musimy podstawić jego współrzędne do równania prostej i sprawdzić, czy jest ono spełnione.

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, x, y;
    cout<<"Podaj a i b równania prostej" <<endl;
    cin>>a>>b;
    cout<<endl<<"Podaj X i Y punktu P"<<endl;
    cin>>x>>y;
    if (y==a*x+b) cout<<endl<<"Punkt należy do prostej";
    else cout<<endl<<"Punkt nie należy do prostej";
}
```




ZADANIA DO WYKONANIA

1. Napisz algorytm, który będzie sprawdzał położenie dwóch punktów względem prostej.
2. Napisz program w języku C++, który będzie sprawdzał położenie dwóch punktów wczytanych przez użytkownika względem prostej.
3. Napisz program w języku Java, który będzie sprawdzał położenie dwóch punktów wczytanych przez użytkownika względem prostej.

PODSUMOWANIE LEKCJI

- s. 250 • Układ współrzędnych (na płaszczyźnie dwuwymiarowej) to dwie prostopadłe do siebie osie liczbowe – oś odciętych X i oś rzędnych Y .
- s. 252 • Położenie punktu w układzie współrzędnych określamy za pomocą współrzędnych – $P(x, y)$. Zawsze pierwsza współrzędna to odcięta, a druga to rzędna.



39. Algorytmy badające własności geometryczne, czyli przynależności punktu do odcinka

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest odcinek i nachylenie odcinka;
- dowiesz się, w jaki sposób sprawdzić, czy dany punkt należy do odcinka.

Wszeghobecne dzisiaj nawigacje, do których mamy dostęp w telefonie, zegarku czy samochodzie, pozwalają zaplanować i obliczyć drogę, którą chcemy przebyć z punktu *A* do punktu *B*.

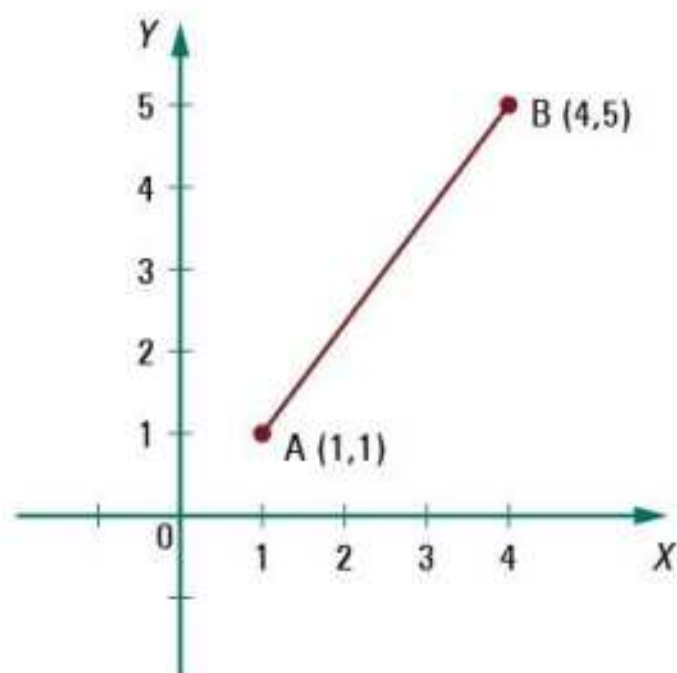
Wytyczając podróż, możemy również sprawdzić, czy punkt *C* będzie znajdował się na naszej drodze, czy też będziemy musieli z niej zboczyć, aby się do niego dostać. Możliwość sprawdzenia tego w urządzeniach mobilnych jest możliwa dzięki algorytmom. Takim algorytmem jest sprawdzanie, czy dany punkt przynależy do odcinka.

39.1. Wstępne założenia związane z przynależnością punktu do odcinka

W tym rozdziale zajmiemy się przypomnieniem sobie podstawowych informacji związanych z przynależnością punktu do odcinka. Przypomnijmy sobie najpierw, czym jest odcinek.

Odcinek jest najkrótszą linią, która łączy dwa punkty. Odcinkiem możemy też nazwać prostą, która jest ograniczona bądź zakończona dwoma punktami. Oczywiście punkty graniczne wchodzą w skład odcinka – wyznaczają np. jego długość bądź nachylenie.

odcinek



Rys. 39.1. Odcinek

Przyjmijmy, że mamy dwa punkty A i B o współrzędnych $A(x_1, y_1)$ i $B(x_2, y_2)$. Jeżeli połączymy je ze sobą linią, to otrzymamy odcinek. Aby obliczyć jego długość, musimy skorzystać ze wzoru:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

nachylenie odcinka

Chcąc sprawdzić, czym jest nachylenie odcinka, musimy na początku obliczyć różnicę poziomów $R_p = (y_2 - y_1)$ i różnicę przesunięcia $P_p = (x_2 - x_1)$. Stosunek tych różnic to nachylenie, które zapiszemy pod postacią:

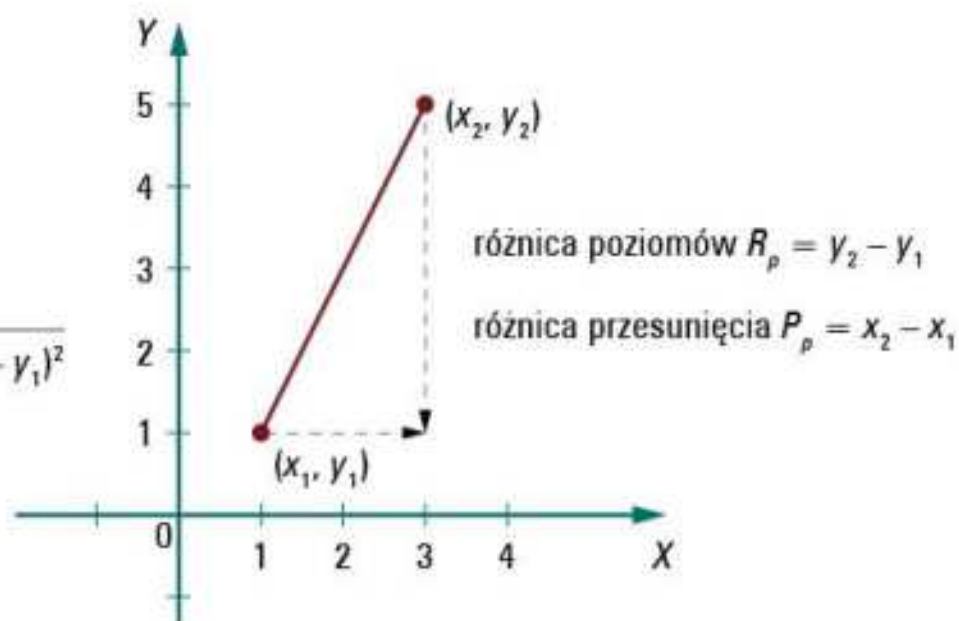
$$N = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

nachylenie:

$$N = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

długość:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Rys. 39.2. Nachylenie i długość odcinka



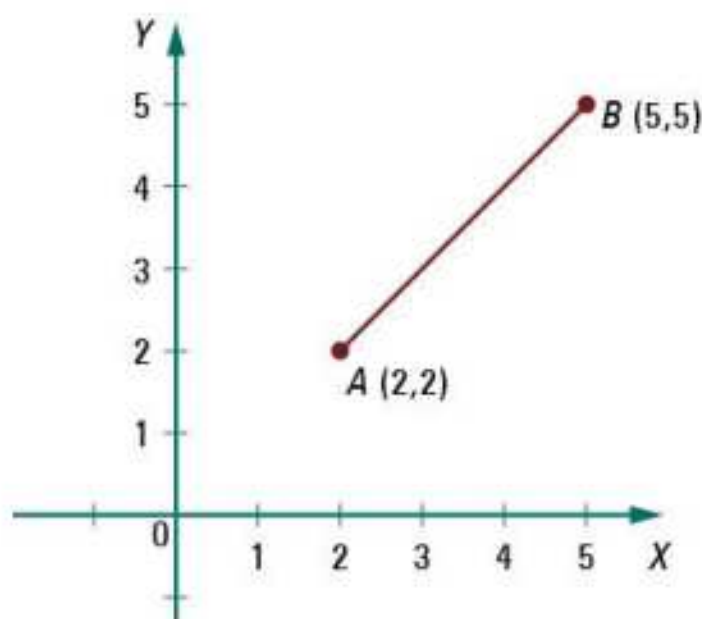
Przedstawiony na rysunku 39.2. odcinek będzie miał nachylenie dodatnie, ponieważ jeżeli punkt A ma współrzędne $(1,1)$, a punkt B współrzędne $(3,5)$, to N wyniesie $\frac{4}{2}$, czyli 2.

Jeżeli odcinek ma nachylenie równe 1, to jest to odcinek równoległy do prostej o równaniu $y = x$. Ma to miejsce wtedy, gdy różnica przesunięcia jest równa 0 – nie można obliczyć nachylenia, ponieważ nie wykonujemy dzielenia przez 0. Natomiast jeżeli nasz odcinek ma nachylenie równe 0, jest on odcinkiem poziomym. Ma to miejsce w przypadku, gdy różnica poziomów jest równa 0.

Odcinki mogą również mieć nachylenie ujemne – wtedy, gdy różnica poziomów R_p będzie mniejsza od 0.

39.2. Przynależność punktu do odcinka

Na wstępie założmy, że na układzie współrzędnych występują trzy punkty – punkt $A(x_1, y_1)$, punkt $B(x_2, y_2)$ i punkt $C(x_3, y_3)$. Przyjmijmy również, że dwa pierwsze z nich będą tworzyły odcinek $|AB|$ – np. $A(2,2)$ i $B(5,5)$.

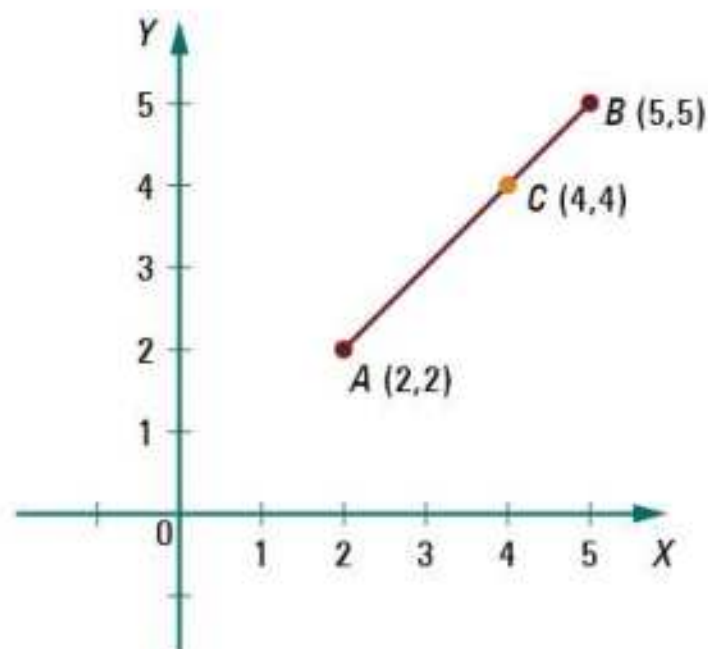


Rys. 39.3. Odcinek $|AB|$



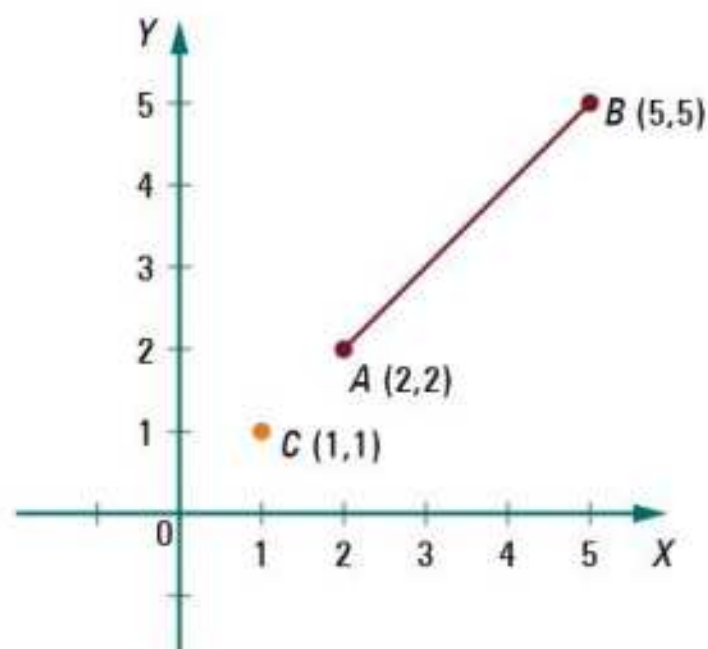
Przyjmijmy również trzy różne możliwe położenia punktu C względem naszego odcinka:

1. Punkt C leży na odcinku.



Rys. 39.4. Położenie punktu C na odcinku $[AB]$

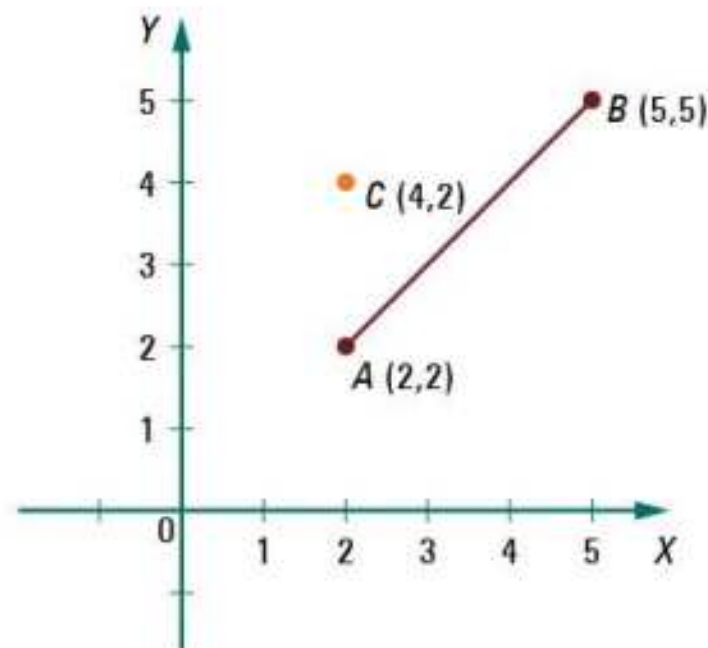
2. Położenie punktu C poza odcinkiem $[AB]$, ale na prostej, która przez niego przechodzi.



Rys. 39.5. Położenie punktu C na prostej, która przechodzi przez odcinek $[AB]$



3. Punkt C leży całkowicie poza odcinkiem $[AB]$.



Rys. 39.6. Położenie punktu C poza odcinkiem $[AB]$

Poniższy program, napisany w języku C++, będzie sprawdzał, czy punkt należy do odcinka, czyli czy jest spełniony pierwszy warunek. Na początku wprowadźmy dane naszego odcinka, czyli podajmy punkty A , B i C :

```
#include <iostream>

using namespace std;

int main()
{
    double x1, x2, x3, y1, y2, y3;

    cout << „Punkt A. Podaj x: „; cin >> x1;
    cout << „Punkt A. Podaj y: „; cin >> y1;
    cout << „Punkt B. Podaj x: „; cin >> x2;
    cout << „Punkt B. Podaj y: „; cin >> y2;
    cout << „Punkt B. Podaj x: „; cin >> x3;
    cout << „Punkt C. Podaj y: „; cin >> y3;

    return 0;
}
```

Kolejnym krokiem będzie sprawdzenie współliniowości, czyli czy punkty A , B i C leżą na jednej prostej. Oczywiście będzie to zbyt mało, aby stwierdzić, że punkt C leży na odcinku $[AB]$.

współliniowość



macierz

Abyśmy mogli zbadać współliniowość punktów, musimy wprowadzić pojęcie macierzy oraz jej wyznacznik. Bardziej szczegółowo zapoznajcie się z tymi pojęciami na matematyce, a tutaj tylko je sobie przypomnimy.

Macierzą nazywamy układ liczb lub symboli, które są ułożone w wiersze i kolumny. Zarówno w matematyce, jak i w informatyce możemy się z takim zapisem spotkać dość często (np. układ równań).

Nasza macierz będzie miała budowę 3×3 :

$$M = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix}$$

W tej macierzy występują jedynki, dlatego że posługujemy się przestrzenią trójwymiarową, a nasze punkty musimy sprowadzić do dwóch wymiarów.

Kolejnym etapem działań na macierzach są operacje po skosie. Na początku mnożymy przez siebie trzy pary liczb po skosie w dół i zsumujemy ich iloczyny, a następnie pomnożymy trzy pary liczby po skosie w górę i odejmiemy ich iloczyny.

Aby tego dokonać, rozszerzymy macierz o dwie linie w dół, kopiując dwa pierwsze wiersze.

$$M = \begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{bmatrix}$$

W kolejnym kroku od góry do dołu, na ukos, wymnożymy, a następnie zsumujemy trzy wartości x_1, y_2 i 1; x_2, y_3 i 1 oraz ostatnia trójka x_3, y_1 i 1.

Drugim etapem w tym kroku będzie wymnożenie, a następnie odjęcie elementów po skosie od dołu do góry x_2, y_1 i 1; x_1, y_3 i 1 oraz x_3, y_2 i 1 – co przedstawia następujące równanie:

$$M = (x_1 \cdot y_2 \cdot 1) + (x_2 \cdot y_3 \cdot 1) + (x_3 \cdot y_1 \cdot 1) - (x_2 \cdot y_1 \cdot 1) - (x_1 \cdot y_3 \cdot 1) - (x_3 \cdot y_2 \cdot 1)$$

```
double wyznacznik = x1*y2+x2*y3+x3*y1-x2*y1-x1*y3-x3*y2;
if( wyznacznik == 0) {
// tu będziemy sprawdzać, czy punkt nie leży poza krawcami
// odcinka
} else cout << „Niestety, punkt C nie leży na tym odcinku.
: ( „ ) ;
```

W równaniu tym możemy pominąć wartość 1 i zająć się wyliczaniem naszego wyznacznika. Sprawdzamy dwa warunki – pierwszy, czy wyznacznik jest różny od



0. Jeżeli ten warunek będzie spełniony, to znaczy, że punkt C w ogóle nie należy do prostej i odcinka; drugi – jeżeli współczynnik równa się 0, sprawdzamy, czy leży na wskazanym przez nas odcinku. Aby tego dokonać, musimy sprawdzić, czy x i y punktu C zawiera się pomiędzy najniższym i najwyższym X oraz najniższym i najwyższym Y . Warto zwrócić uwagę na to, że sprawdzając ten warunek, musimy zastosować znaki $\geq$ i $\leq$, ponieważ punkt C może pokrywać się z punktami A i B . Na koniec sprawdzamy opcję trzecią – czy punkt leży poza odcinkiem, ale należy do prostej, która przez ten odcinek przechodzi.

```
double wyznacznik = x1*y2+x2*y3+x3*y1-x2*y1-x1*y3-x3*y2;
if( wyznacznik == 0)
{
    if( (x3 <= min(x1, x2)) && (x3 <= max(x1, x2)) && (y3
    >= min(y1, y2)) && (y3 <= max(y1, y2)) ) cout << „C leży
    na |AB|” << endl;
    else cout << „C nie leży na |AB| choć leżą na jednej
    prostej” << endl;
}
else cout << „Niestety, punkt C nie leży na tym odcinku.
: („ << endl;
```

ZADANIA DO WYKONANIA

1. Napisz algorytm, który będzie obliczał długość odcinka.
2. Napisz algorytm, który będzie obliczał nachylenie odcinka.
3. Napisz algorytm, który będzie sprawdzał, czy dany punkt należy do odcinka.
4. Napisz program w języku C++, który będzie obliczał długość odcinka.
5. Napisz program w języku Java, który będzie obliczał długość odcinka.
6. Napisz program w języku C++, który będzie obliczał nachylenie odcinka.
7. Napisz program w języku Java, który będzie obliczał nachylenie odcinka.
8. Napisz program w języku Java, który będzie sprawdzał, czy dany punkt należy do odcinka.

PODSUMOWANIE LEKCJI

- Odcinek jest najkrótszą linią, która łączy dwa punkty.
- Nachylenie jest stosunkiem różnicy poziomów odcinka i różnicy jego przesunięcia.
- Współliniowość punktów występuje wtedy, gdy punkty A , B i C leżą na jednej prostej.
- Macierzą nazywamy układ liczb lub symboli, które są ułożone w wiersze i kolumny.

s. 253

s. 254

s. 257

s. 258

40. Skrzyżowanie dróg, czyli badanie przecinania się odcinków

NA TEJ LEKCJI:

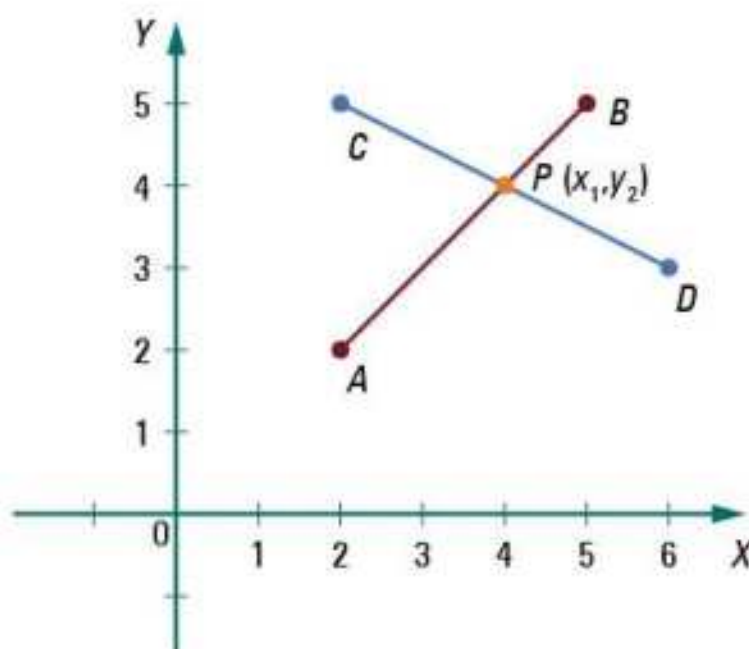
- dowiesz się, kiedy występuje przecinanie się odcinków;
- poznasz, czym jest „czyste przecięcie” oraz „przecięcie brzegowe”;
- nauczysz się badać przecinanie odcinków.

Na fizyce lub matematyce często spotykamy zadania typu: „Z punktu A do punktu B zaczyna poruszać się obiekt nr 1 oraz z punktu C do punktu D zaczyna poruszać się obiekt nr 2. Mając dane dotyczące prędkości i masy, oblicz, w którym miejscu przetną się ich drogi”. Do obliczania tego typu zadań możemy wykorzystać wiedzę na temat przecinania się odcinków. Bo skoro oba obiekty przemieszczają się ku sobie, to zakładamy, że przemieszczają się one po prostej, a skoro z punktu A do B i z punktu C do D , to na tych prostych mamy wyznaczone odcinki AB i CD .

40.1. Przecinanie się odcinków

przecinanie się
odcinków

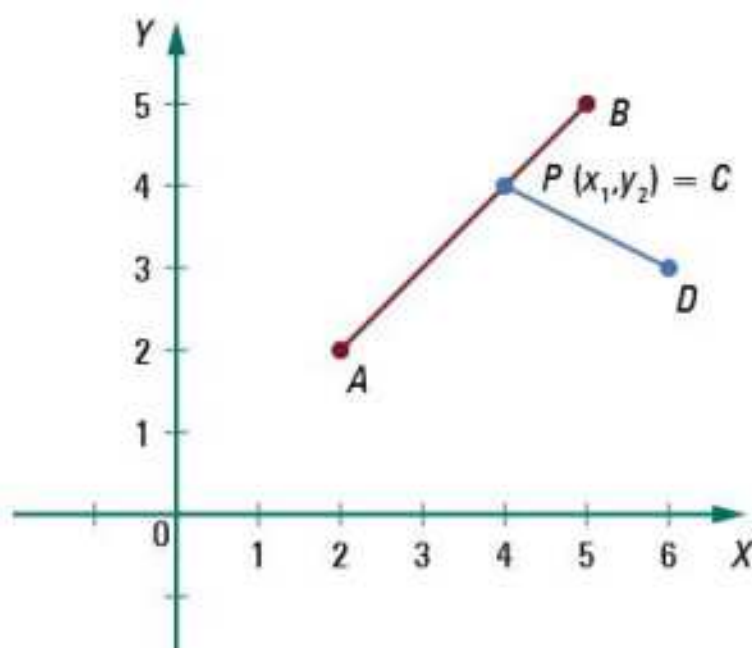
Jak sprawdzić, czy dwa odcinki się przecinają? Można powiedzieć, że odcinki się przecinają, gdy mają jeden wspólny punkt i nie zawierają się w jednej prostej. Oczywiście możemy mieć różne sytuacje przecięcia się odcinków. Dajmy na to, że mamy odcinek AB i odcinek CD , które mają wspólny punkt przecięcia $P(x_1, y_2)$ i żaden z końców odcinka nie należy do odcinków przecinających się.



Rys. 40.1. Przecięcie się odcinków w punkcie P



Inny przypadek przecinania się odcinków zachodzi wtedy, gdy jeden z końców odcinka należy do drugiego odcinka, czyli gdy punkt przecięcia P jest równy np. punktowi C .



Rys. 40.2. Przecięcie się odcinków – punkt przecięcia jest początkiem drugiego odcinka

40.2. Badanie przecinania się odcinków

To, czy dwa odcinki się przecinają, możemy sprawdzić, wyznaczając równania prostych przechodzących przez odcinki, znaleźć ich punkt przecięcia i sprawdzić, czy należy on do obu odcinków.

Możemy przyjąć, że odcinki się przecinają, gdy mają przynajmniej jeden wspólny punkt.

By sprawdzić, po której stronie prostej zawierającej odcinek AB znajduje się punkt C , kończący odcinek CD , sprawdzimy, po której stronie odcinka AB znajduje się punkt C . Tę samą czynność wykonamy w przypadku punktu D kończącego odcinek CD .

Policzymy iloczyn wektorowy:

$$S_1 = (C - A) \times (B - A)$$

oraz

$$S_2 = (D - A) \times (B - A)$$

Analogicznie będziemy postępować, sprawdzając, czy AB przecina prostą zawierającą CD . Obliczamy iloczyn:

$$S_3 = (A - C) \times (D - C)$$

oraz

$$S_4 = (B - C) \times (D - C)$$

czyste
przecięcie się
odcinków

przecięcie
brzegowe
odcinków

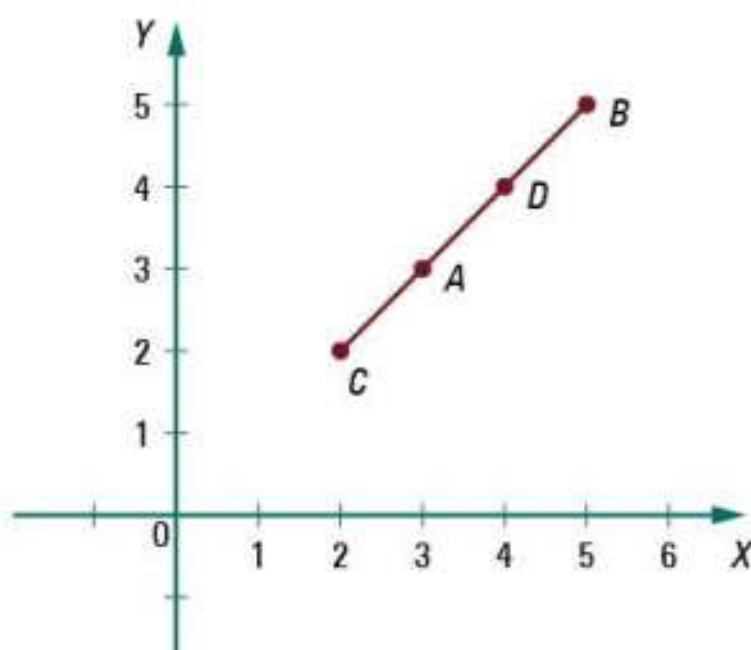
W przypadku, gdy iloczyny S_1 i S_2 mają przeciwne znaki i żaden z nich nie równa się zero, oraz gdy S_3 i S_4 mają przeciwne znaki i żaden z nich nie równa się zero, mamy tzw. czyste przecięcie – końce jednego odcinka nie należą do drugiego. Możemy wtedy stwierdzić, że odcinki się przecinają.

Jeżeli jeden z iloczynów jest równy zero, oznacza to, że jeden z końców odcinka leży na prostej zawierającej drugi odcinek – mówimy wtedy o „przecięciu brzegowym”. Musimy wtedy sprawdzić, czy punkt przecięcia leży na odcinku wchodzącym w skład prostej.

Wiedząc, że punkty są współliniowe, musimy zbadać, czy jedna ze współrzędnych tego punktu znajduje się między współrzędnymi końców tego odcinka. Dla odcinka AB wybierzemy punkt C , który musi spełnić warunek:

$$C \geq \min AB \text{ i } C \leq \max AB$$

Jeżeli ten warunek nie będzie spełniony, nie przesądza to jeszcze o tym, że odcinki się nie przecinają. Należy sprawdzić go dla wszystkich S_i , $i = 1, 2, 3, 4$, które wyniosą zero.



Rys. 40.3. Odcinki AB i CD przecinają się w nieskończonej liczbie punktów

Na rysunku 40.3. możemy zauważyć, że odcinki leżą na jednej prostej i „nakładają się na siebie”. Przyjmijmy, że obliczyliśmy $S_1 = 0$. Przykład zatem nie stanowi „czystego przecięcia”. Następnie sprawdzimy, czy punkt C leży między odcinkiem $|AB|$ – okazuje się, że nie. Gdy przerwiemy w tym momencie sprawdzanie i stwierdzimy, że odcinki się nie przecinają, nie będzie to prawdą, ponieważ mamy $S_2 = 0$, a punkt D leży na odcinku AB .

Przedstawmy to w pseudokodzie:

```
iloczyn(A, B, C) {
  return (x2 - x1)(y3 - y1) - (x3 - x1)(y2 - y1)
}
```



```
punkt_miedzy(A, B, C) {
  if min{x1, x2} ≥ x3 ≤ max {x1,x2} return true
}
```

Będziemy mogli wtedy zdefiniować funkcję:

```
przecina(A, B, C, D) {
  S1 = iloczyn (A, C, B)
  S2 = iloczyn (A, D, B)
  S3 = iloczyn (C, A, D)
  S4 = iloczyn (C, B, D)
  if ((S1 > 0 && S2 < 0) || (S1 < 0 && S2 > 0)) && ((S3 < 0
  && S4 > 0) || (S3 > 0 && S4 < 0)) return true
  else if S1 = 0 punkt_miedzy (A, B, C) return true
  else if S2 = 0 punkt_miedzy (A, B, D) return true
  else if S3 = 0 punkt_miedzy (C, D, A) return true
  else if S4 = 0 punkt_miedzy (C, D, B) return true
  else return false
}
```

Powyższy algorytm sprawdza, czy odcinki „przecinają się czysto” oraz czy koniec jednego z odcinków nie leży na drugim.

ZADANIA DO WYKONANIA

1. Napisz algorytm w postaci listy kroków, który będzie sprawdzał, czy odcinki się przecinają.
2. Napisz program w języku C++, który będzie sprawdzał, czy odcinki podane przez użytkownika się przecinają.
3. Napisz program w języku Java, który będzie sprawdzał, czy odcinki podane przez użytkownika się przecinają.

PODSUMOWANIE LEKCJI

- Odcinki się przecinają, gdy mają jeden wspólny punkt i nie zawierają się w jednej prostej. s. 260
- O „czystym przecięciu” mówimy wtedy, gdy odcinki się przecinają i końce jednego odcinka nie należą do drugiego. s. 262
- O „przecięciu brzegowym” mówimy wtedy, gdy jeden z końców odcinka leży na prostej zawierającej drugi odcinek. s. 262



41. Wewnątrz czy na zewnątrz, czyli badanie przynależności punktu do trójkąta

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest trójkąt z matematycznego punktu widzenia;
- dowiesz się, kiedy dany punkt jest przynależny do trójkąta;
- poznasz dwie metody implementacji, które sprawdzą, czy punkt jest przynależny do trójkąta.

Jednym z problemów omawianych na zajęciach matematyki podczas badania własności geometrycznych jest sprawdzenie położenia jakiegoś punktu względem konkretnej figury. Trójkąt jest najmniejszym wielokątem i na jego przykładzie najprościej jest sprawdzić, czy dany punkt leży w jego obszarze, czyli czy jest mu przynależny.

41.1. Trójkąt

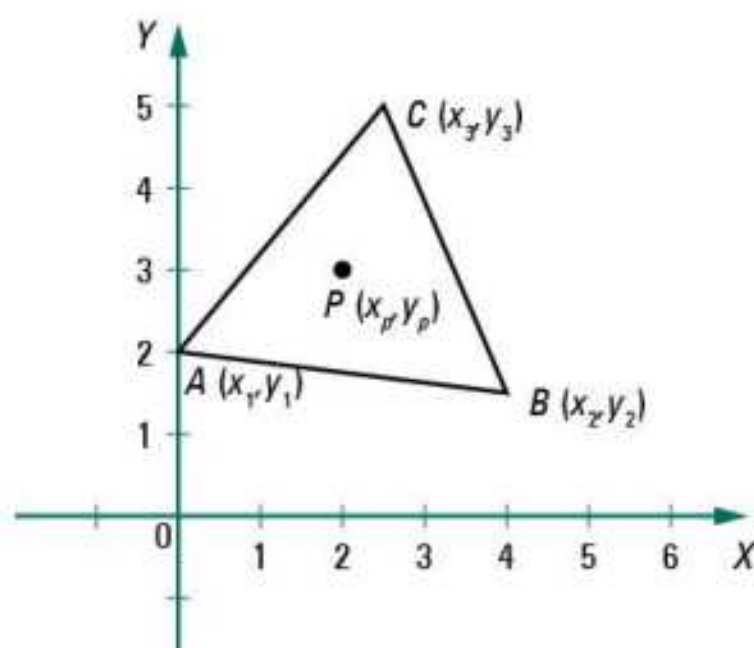
Definicję trójkąta poznajemy już w szkole podstawowej na matematyce. Ze względu na to, że było to dość dawno temu, przypomnijmy ją sobie.

trójkąt

Trójkąt to najmniejsza figura wypukła i domknięta, zawierająca pewne trzy ustalone i niewspółliniowe punkty płaszczyzny (otoczka wypukła wspomnianych trzech punktów). Odcinki tworzące łamaną nazywamy bokami, a punkty wspólne sąsiednich boków – wierzchołkami trójkąta.

W tym rozdziale zajmiemy się tym, w jaki sposób sprawdzić, czy punkt o danych współrzędnych przynależy do trójkąta, czy też jest poza nim.

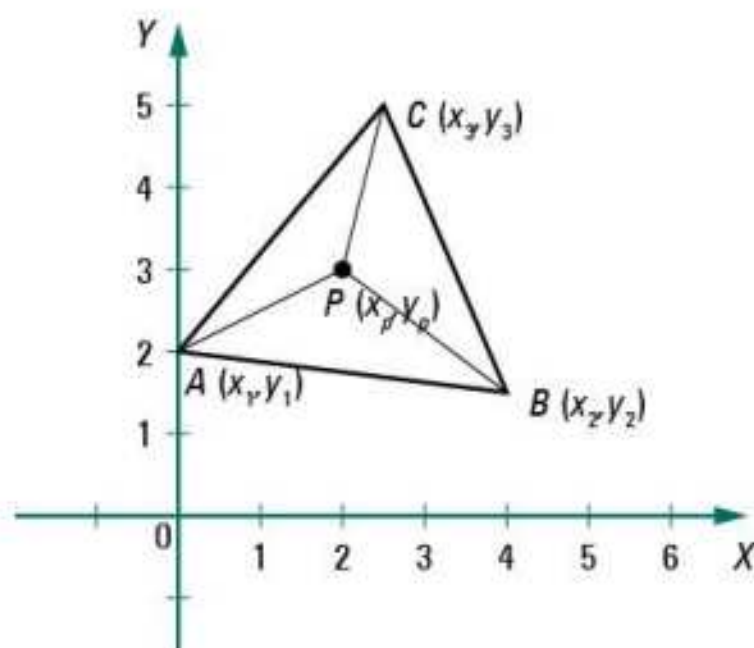
Mamy dany trójkąt T , który został opisany przez współrzędne trzech wierzchołków $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$, oraz punkt P o współrzędnych (x_p, y_p) . Sprawdzimy, czy punkt P przynależy do trójkąta T o wierzchołkach A , B i C .



Rys. 41.1. Przynależność punktu do trójkąta

41.2. Badanie przynależności punktu do trójkąta – metoda implementacji

Jednym z prostszych sposobów rozwiązania tego problemu jest podzielenie danego trójkąta na trzy trójkąty, których jednym z wierzchołków będzie punkt P , a pozostałymi będą wierzchołki należące do trójkąta ABC .

Rys. 41.2. Podział trójkąta ABC na trzy mniejsze

Suma pól wszystkich powstałych nowych trójkątów powinna być równa polu trójkąta podstawowego. Jeżeli tak nie jest, oznacza to, że punkt P leży poza danym trójkątem.

Wzór, którego należy użyć, musi się opierać na wyliczeniu pola trójkąta na podstawie jego współrzędnych:



$$T = \frac{1}{2} \cdot |(x_1 \cdot (y_2 - y_3) + x_2 \cdot (y_3 - y_1) + x_3 \cdot (y_1 - y_2))|$$

Wzór ten możemy zapisać w formie funkcji `WewnetrznePole()`:

```
double Wewne-
trznePole(int x1, int y1, int x2, int y2, int x3, int y3) {
    return 0.5*abs((x1*(y2 - y3) + x2*(y3 - y1) + x3*(y1
- y2)));
}
```

Aby sprawdzić, czy punkt P jest wewnątrz danego trójkąta, będziemy musieli zsumować pola wszystkich trójkątów, które powstały po połączeniu wierzchołków trójkąta ABC z punktem P , a następnie ustalić, czy suma tych pól równa się wielkości pola trójkąta ABC .

```
bool Przynaleznosc(int x1, int y1, int x2, int y2, int x3,
int y3, int xp, int yp) {
    double T = WewnetrznePole (x1, y1, x2, y2, x3, y3);
    double T1 = WewnetrznePole (xp, yp, x2, y2, x3, y3);
    double T2 = WewnetrznePole (x1, y1, xp, yp, x3, y3);
    double T3 = WewnetrznePole (x1, y1, x2, y2, xp, yp);
    return (T == T1 + T2 + T3);
}
```

Inną, bardziej efektywną metodą, dzięki której będziemy mogli sprawdzić, czy punkt P przynależy do trójkąta ABC , jest metoda z użyciem współrzędnych barycentrycznych.

współrzędne
barycentryczne

W dwuwymiarowym układzie współrzędnych każdy punkt na płaszczyźnie (w naszym przykładzie wierzchołki trójkąta i punkt wewnątrz trójkąta) jest opisany za pomocą współrzędnych (x, y) . Natomiast we współrzędnych barycentrycznych położenie punktu (w naszym przykładzie leżącego wewnątrz trójkąta) jest opisywane w odniesieniu do współrzędnych (x, y) wierzchołków.

Aby sprawdzić tą metodą, czy punkt P przynależy do trójkąta ABC , musimy na początku przekształcić punkty względem punktu A :

$$B = B - A, C = C - A, P = P - A.$$

Następnie wyliczymy $d = x_2 \cdot y_3 - x_3 \cdot y_2$. Ostatnią czynnością będzie sprawdzenie, czy następujące wartości znajdują się w przedziale $[0, d]$:

1. $x_p (y_2 - y_3) + y_p (x_3 - x_2) + x_2 \cdot y_3 - x_3 \cdot y_2$
2. $x_p \cdot y_3 - y_p \cdot x_3$
3. $y_p \cdot x_2 - x_p \cdot y_2$



```
bool Przynaleznosc(int x1, int y1, int x2, int y2, int x3,
int y3, int xp, int yp) {
    x2 -= x1; y2 -= y1;
    x3 -= x1; y3 -= y1;
    xp -= x1; yp -= y1;
    double d = x2*y3 - x3*y2;
    double wb1 = xp*(y2 - y3) + yp*(x3 - x2) + x2*y3 - x3*y2;
    double wb2 = xp*y3 - yp*x3;
    double wb3 = yp*x2 - xp*y2;
    return Zakres(wb1, 0, d) && Zakres(wb2, 0, d) &&
    Zakres(wb3, 0, d);
}
```

Funkcja `Zakres()` zwraca nam wartość logiczną `true` wtedy i tylko wtedy, gdy podana wartość `val` należy do zakresu `[min, max]`:

```
bool Zakres(double val, double min, double max) {
    return val >= min && val <= max;
}
```

ZADANIA DO WYKONANIA

1. Napisz algorytm w postaci listy kroków, który będzie sprawdzał, czy punkt przynależy do trójkąta.
2. Napisz program w języku C++, za pomocą którego użytkownik sprawdzi, czy podany przez niego punkt jest przynależny do trójkąta, którego dane wprowadzi również użytkownik.
3. Napisz program w języku Java za pomocą którego użytkownik sprawdzi, czy podany przez niego punkt jest przynależny do trójkąta, którego dane wprowadzi również użytkownik.

PODSUMOWANIE LEKCJI

- Trójkąt to najmniejsza figura wypukła i domknięta, zawierająca pewne trzy ustalone i niewspółliniowe punkty płaszczyzny (otoczka wypukła wspomnianych trzech punktów). Odcinki tworzące łamaną nazywamy bokami, a punkty wspólne sąsiednich boków – wierzchołkami trójkąta.
- We współrzędnych barycentrycznych położenie punktu opisywane jest w odniesieniu do współrzędnych (x, y) wierzchołków.

s. 264

s. 266



42. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą prostokątów

NA TEJ LEKCJI:

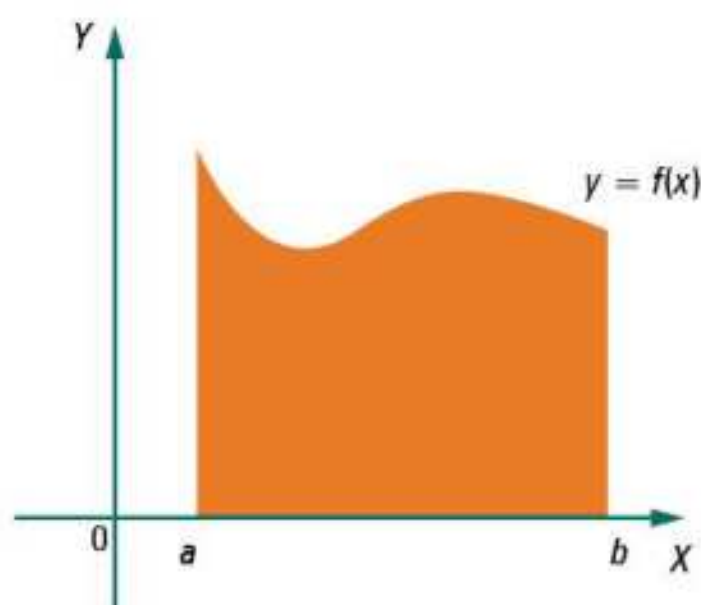
- przypomnisz sobie, czym jest całka oznaczona i całka nieoznaczona;
- dowiesz się, czym jest całkowanie numeryczne;
- nauczysz się całkowania numerycznego metodą prostokątów.

Czasami znajdziemy się w sytuacji, gdy musimy dokonać nietypowych obliczeń lub pomiarów. Jednym z takich działań jest obliczanie pól powierzchni, które zawierają różnego rodzaju zaokrąglenia lub krzywizny. Na ratunek przychodzi nam wtedy całkowanie. Obliczając pole obszaru ograniczonego funkcją, wykorzystamy całkowanie numeryczne.

42.1. Definicja całki

całka oznaczona

Całkowanie jest jedną z najważniejszych operacji w matematyce analitycznej, a definicję całki możemy zawrzeć w jednym zdaniu. Jest to funkcja pierwotna względem funkcji pochodnej lub różniczki. Jeżeli mówimy o całkach, to najczęściej przez całkę rozumiemy całkę oznaczoną lub całkę nieoznaczoną. Całka oznaczona jest również nazywana całką Riemanna – od nazwiska Bernharda Riemanna (wym. bernarda rimana), który w 1854 roku opisał ją w swojej pracy jako pole powierzchni między wykresem funkcji $f(x)$ w pewnym przedziale $[a, b]$ a osią odciętych.



Rys. 42.1. Prezentacja w formie graficznej całki oznaczonej dla funkcji $y=f(x)$ w przedziale $[a, b]$



42.2. Metoda prostokątów

W tym rozdziale stworzymy algorytm, dzięki któremu uda nam się w przybliżony sposób obliczyć wartość pola figury powstałej z zamknięcia obszaru między wykresem funkcji $y = f(x)$ a odcinkiem AB położonym na osi X i prostymi a i b (pole przedstawione na rys. 42.1.). Metoda ta pozwoli nam obliczyć pole powstałej figury z określoną dokładnością. Wyznaczenie tej przybliżonej wartości pola nazywamy całkowaniem numerycznym.

całkowanie
numeryczne

Żeby obliczyć przybliżoną wartość pola obszaru wyznaczonego na rysunku 42.1., musimy podzielić go na określoną liczbę figur, których pola nie jest zbyt trudno wyznaczyć. Najczęściej do tego celu stosujemy prostokąty. Suma pól prostokątów, których użyliśmy do podziału całego obszaru, będzie stanowiła rozwiązanie, czyli przybliżoną wartość pola tego obszaru.

Metoda prostokątów jest prostym i skutecznym sposobem wyznaczenia przybliżonej wartości całki oznaczonej, czyli oszacowania pola określonego obszaru. Dzielimy nasz przedział $[a, b]$ na określoną (n) liczbę jednakowych podprzedziałów – liczba podprzedziałów jest liczbą naturalną większą od 0.

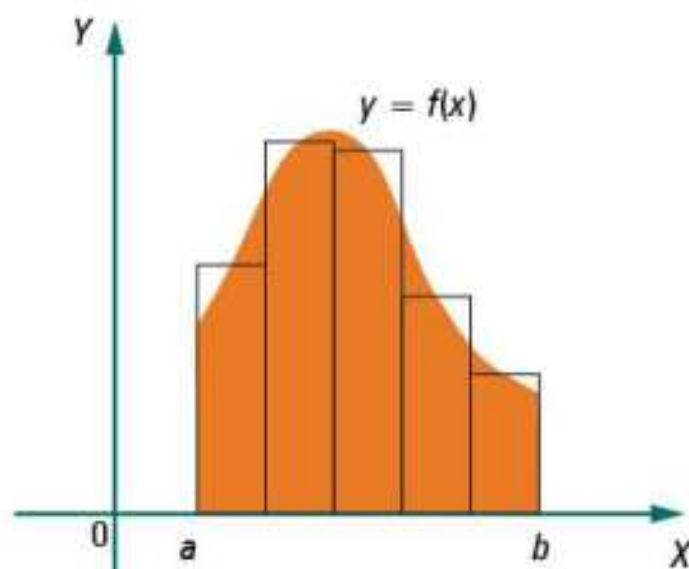
metoda
prostokątów

Nasz przedział $[a, b]$ jest równy przedziałowi $[x_0, x_n]$.

Długość każdego z podprzedziałów:

$$x_{i+1} - x_i \text{ dla } i=0,1, \dots, n-1 \text{ wynosi } \frac{b-a}{n}$$

Rysunek 42.2. przedstawia realizację całkowania numerycznego metodą prostokątów dla $n = 6$.



Rys. 42.2. Realizacja całkowania numerycznego metodą prostokątów

Im więcej prostokątów użyjemy do obliczeń, tym dokładniejszy będzie wynik, który otrzymamy. Długość przedziału równą szerokości podstawy prostokąta będziemy wyznaczać ze wzoru:

$$dl = \frac{b-a}{n}$$

n to liczba użytych prostokątów.



Wysokość prostokąta będzie natomiast równa wartości bezwzględnej funkcji $f(x)$ w środku przedziału – będziemy ją wyznaczać ze wzoru:

$$h = \left| f\left(a + dl \cdot i - \frac{dl}{2}\right) \right|$$

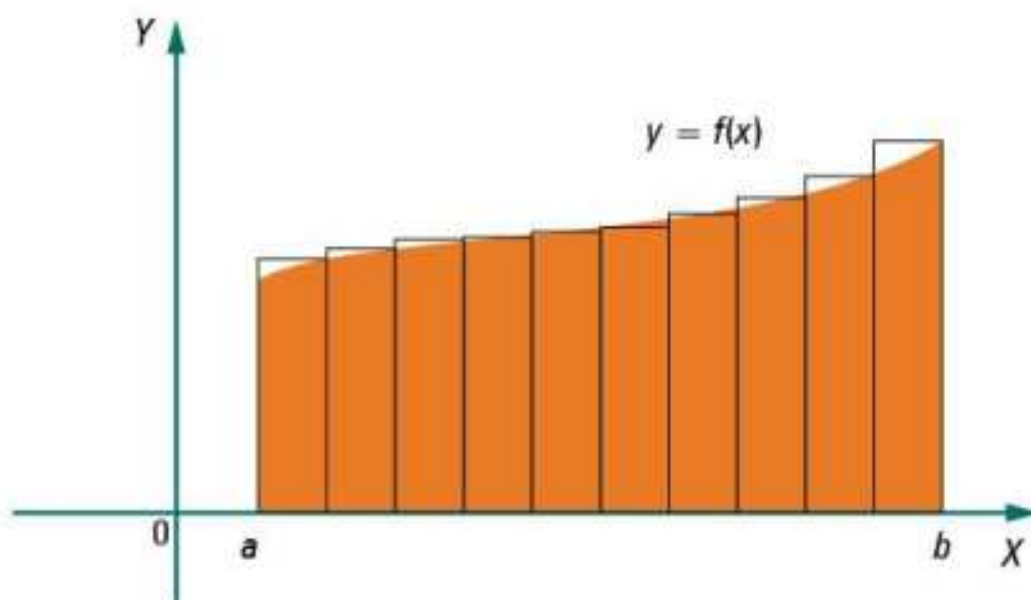
$i = 1, 2, \dots, n$. W podanym powyżej wzorze będziemy odejmowali $\frac{dl}{2}$ w celu przecięcia i -tego prostokąta z wykresem funkcji w środku jego podstawy.

Natomiast wykorzystanie wartości bezwzględnej pozwoli uzyskać wartości ujemne danej funkcji. To, czy wykres znajduje się nad czy pod osią OX , nie będzie miało wpływu na wielkość pola obszaru przez nas wyznaczanego. Istotna za to będzie odległość położenia funkcji od osi OX .

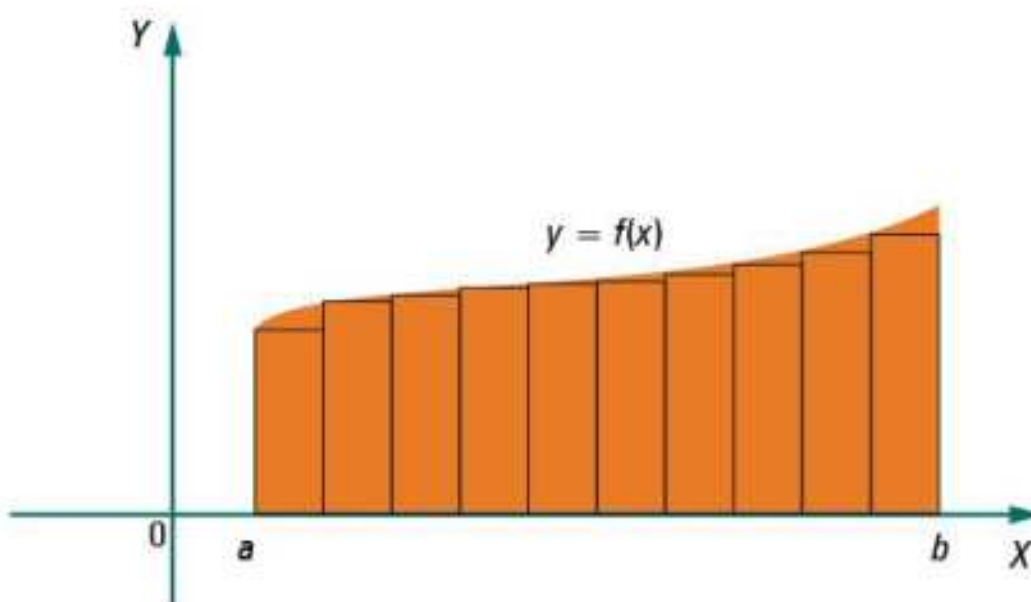
Przybliżone pole badanego obszaru będziemy opisywali funkcją:

$$s = dl \cdot \sum_{i=1}^n \left| f\left(a + dl \cdot i - \frac{dl}{2}\right) \right|$$

Rysunek 42.2. przedstawia realizację całkowania numerycznego metodą prostokątów z efektem nadmiaru i niedomiaru. Poniższe rysunki obrazują ten efekt jeszcze dokładniej.



Rys. 42.3. Realizacja całkowania numerycznego metodą prostokątów z nadmiarem



Rys. 42.4. Realizacja całkowania numerycznego metodą prostokątów z niedomiarem



Stwórzmy schemat blokowy realizujący obliczanie obszaru pola metodą prostokątów.

Specyfikacja algorytmu:

Zmienne główne:

a – początek przedziału

b – koniec przedziału

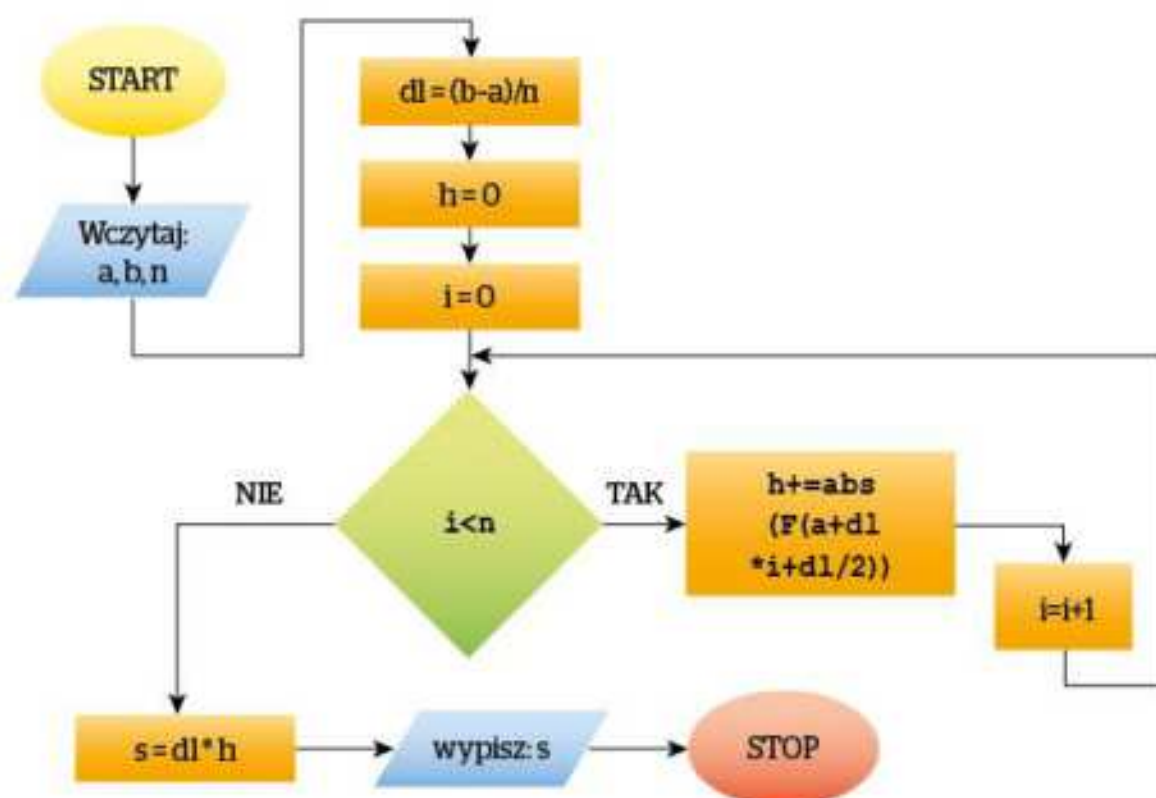
n – ilość prostokątów

Zmienne pomocnicze:

s – suma pól prostokątów

dl – pierwszy bok – każdy prostokąt ma taki sam

h – wysokość prostokąta



Rys. 42.5. Schemat blokowy realizujący obliczanie obszaru pola metodą prostokątów



ZADANIA DO WYKONANIA

1. Wyznacz pole powierzchni obszaru ograniczonego wykresem funkcji $f(x) = x^2 - x - 5$, osią OX oraz prostą a przecinającą oś X w punkcie 5 i prostą b przecinającą oś X w punkcie 7. Zastosuj metodę prostokątów dla $n = 15$.
2. Wyznacz pole powierzchni obszaru ograniczonego wykresem funkcji $f(x) = \cos(x) + 2$, osią OX oraz prostą a przecinającą oś X w punkcie 2 i prostą b przecinającą oś X w punkcie 4. Zastosuj metodę prostokątów dla $n = 5$.
3. Napisz program w języku C++, za pomocą którego użytkownik wyznaczy pole obszaru ograniczonego wykresem funkcji $f(x)$, osią OX oraz prostymi a i b z zastosowaniem metody prostokątów.
4. Napisz program w języku Java, za pomocą którego użytkownik wyznaczy pole obszaru ograniczonego wykresem funkcji $f(x)$, osią OX oraz prostymi a i b z zastosowaniem metody prostokątów.

PODSUMOWANIE LEKCJI

- s. 268 • Całka oznaczona, nazywana również całką Riemanna, pozwala obliczyć pole powierzchni między wykresem funkcji $f(x)$ w pewnym przedziale $[a, b]$ a osią odciętych.
- s. 268 • Całka nieoznaczona jest rozumiana jako pojęcie odwrotne do pochodnej funkcji.
- s. 269 • Całkowanie numeryczne w przybliżony sposób umożliwia obliczenie wartości pola figury, która powstała z zamknięcia obszaru między wykresem funkcji $y = f(x)$.
- s. 269 • Metoda prostokątów jest prostym i skutecznym sposobem wyznaczenia przybliżonej wartości całki oznaczonej poprzez podzielenie obliczanego pola na mniejsze obszary, zazwyczaj prostokąty.



43. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą trapezów

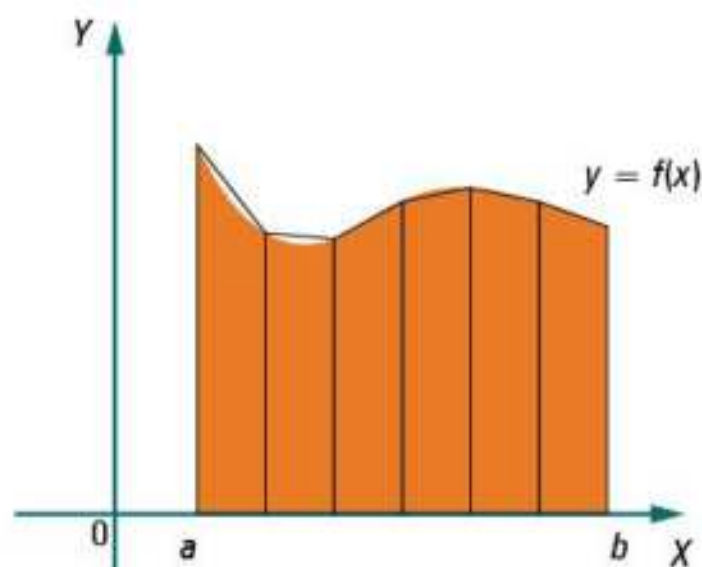
NA TEJ LEKCJI:

- poznasz, czym jest całkowanie numeryczne metodą trapezów.

Całkowanie numeryczne metodą prostokątów nie zawsze da satysfakcjonujące wyniki. Może się zdarzyć, że będziemy potrzebowali obliczeń bardziej precyzyjnych, zawierających mniejsze błędy lub odstępstwa. Wtedy powinniśmy sięgnąć po całkowanie numeryczne metodą trapezów.

43.1. Całkowanie numeryczne metodą trapezów

W poprzednim rozdziale omówiliśmy całkowanie numeryczne metodą prostokątów, które dawało nam możliwość obliczenia powierzchni pola obszaru ograniczonego z jednej strony funkcją $f(x)$. Wynik takich obliczeń był tylko przybliżony. Jeżeli prostokąty zastąpimy trapezami, to otrzymamy bardziej precyzyjne wyliczenia.



Rys. 43.1. Prezentacja w formie graficznej całkowania numerycznego metodą trapezów

Analizując rysunek 43.1., dojdziemy do wniosku, że metoda trapezów jest bardziej precyzyjna niż metoda prostokątów. Przy dużej liczbie trapezów błędy będą prawie niewidoczne.

Zapiszmy wzór pojedynczego trapezu, który będzie zawarty w przedziale $[x_i, x_{i+1}]$ dla $i = 0, 1, \dots, n-1$ (gdzie i określa kolejny trapez).

Na początku określimy wysokość i podstawę trapezu. Jego wysokość to szerokość jednego przedziału:

$$dl = \frac{b-a}{n}$$

metoda trapezów



Do rozważań założmy, że wykres $y = f(x)$ leży nad osią OX .

Podstawami trapezu będą wartości funkcji dla końców przedziałów:

$$p = f(a + dl \cdot (i-1))$$

$$q = f(a + dl \cdot i)$$

Wzór na pole trapezu będzie się przedstawiał następująco:

$$S_{\text{trapezu}} = \frac{p+q}{2} \cdot dl$$

Do obliczenia pola obszaru zastosujemy wzór:

$$s = \frac{f(a)+f(a+dl)}{2} \cdot dl + \frac{f(a+dl)+f(a+2 \cdot dl)}{2} \cdot dl + \dots$$

$$+ \frac{f(a+(n-1) \cdot dl) + f(a+n \cdot dl)}{2} \cdot dl$$

Zauważmy, że $f(a)$ i $f(b) = f(a+n \cdot dl)$ pojawia się tylko jeden raz, natomiast $f(a+dl)$, $f(a+2 \cdot dl)$, ..., $f(a+(n-1) \cdot dl)$ pojawiło się we wzorze dwukrotnie, dlatego wzór na obliczenie pola danego obszaru możemy zapisać w następujący sposób:

$$s = \frac{dl}{2} \left(f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a+i \cdot dl) \right)$$

Identycznie jak w metodzie prostokątów, również podczas tworzenia algorytmu obliczającego całkowanie numeryczne metodą trapezów musimy wziąć pod uwagę ujemne wartości funkcji. I tym razem na pole powierzchni danego obszaru nie będzie miało wpływu położenie funkcji $f(x)$ względem osi OX . Stosujemy więc we wzorze wartość bezwzględną:

$$s = \frac{dl}{2} \left(|f(a)| + |f(b)| + 2 \sum_{i=1}^{n-1} |f(a+i \cdot dl)| \right)$$

Stwórzmy zatem schemat blokowy obliczania pola danego obszaru metodą trapezów.

Specyfikacja algorytmu:

Zmienne główne:

a – początek przedziału

b – koniec przedziału

n – ilość trapezów

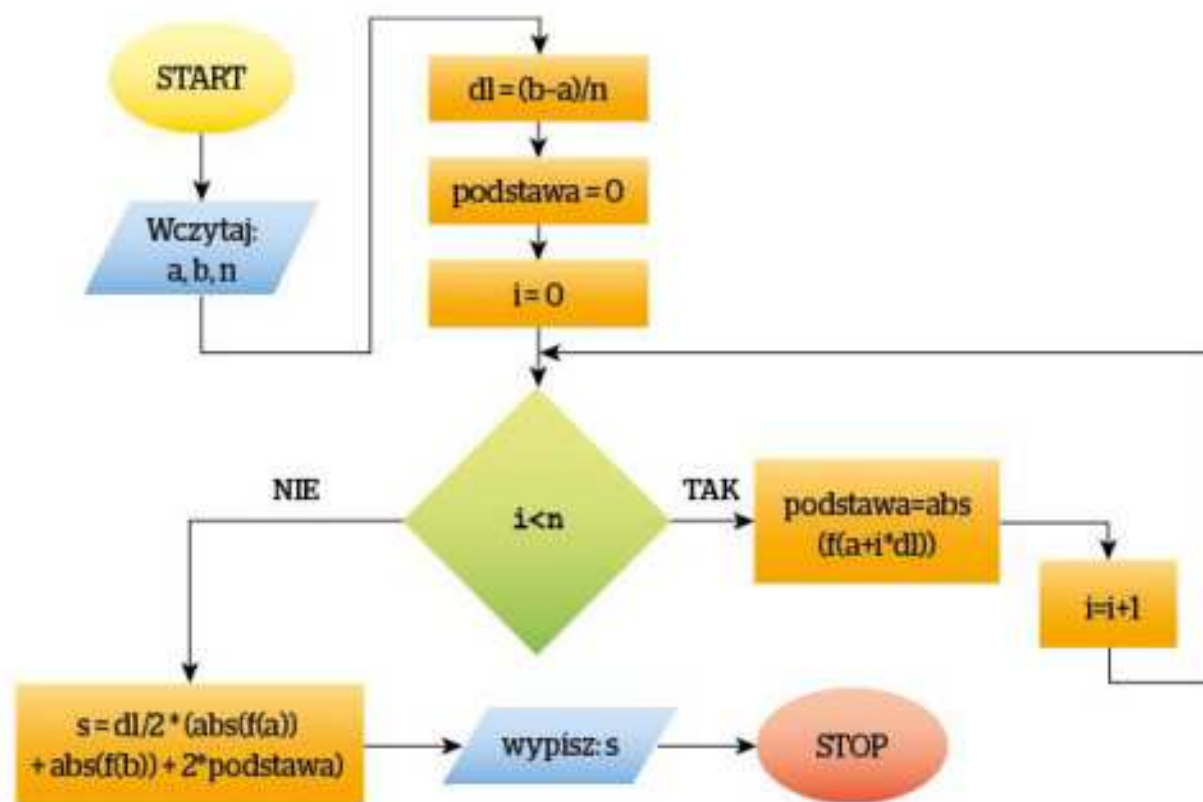
Zmienne pomocnicze:

s – suma pól trapezów

dl – wysokość trapezu

podstawa – podstawa trapezu

algorytm oblicza sumę pól n trapezów.



Rys. 43.2. Schemat blokowy obliczania pola obszaru metodą trapezów

ZADANIA DO WYKONANIA

1. Wyznacz pole powierzchni obszaru ograniczonego wykresem funkcji $f(x) = \cos(x)+1$, osią OX oraz prostą a przecinającą oś X w punkcie 2 i prostą b przecinającą oś X w punkcie 6. Zastosuj metodę trapezów dla $n = 100$.
2. Wyznacz pole powierzchni obszaru ograniczonego wykresem funkcji $f(x) = x^2 - x - 1$, osią OX oraz prostą a przecinającą oś X w punkcie 3 i prostą b przecinającą oś X w punkcie 7. Zastosuj metodę trapezów dla $n = 8$.
3. Wyznacz pole powierzchni obszaru ograniczonego wykresem funkcji $f(x) = x^3 - x^2 + 1$, osią OX oraz prostą a przecinającą oś X w punkcie 1 i prostą b przecinającą oś X w punkcie 9. Zastosuj metodę trapezów dla $n = 20$.
- 4*. Napisz program w języku C++, za pomocą którego użytkownik wyznaczy pole obszaru ograniczonego wykresem funkcji $f(x) = x^2 - x - 1$, osią OX oraz prostymi a i b z zastosowaniem metody trapezów.
- 5*. Napisz program w języku Java, za pomocą którego użytkownik wyznaczy pole obszaru ograniczonego wykresem funkcji $f(x) = x^2 - x - 1$, osią OX oraz prostymi a i b z zastosowaniem metody trapezów.

PODSUMOWANIE LEKCJI

- Obie metody dla małej liczby przedziałów są nieefektywne. Wówczas o tym, która jest lepsza, decyduje przypadek lub – określając to lepiej – rodzaj funkcji i szczęście w doborze punktów podziału. Przyjmuje się, że metoda trapezowa jest bardziej precyzyjna. W obu metodach przedziały muszą być tak małe, aby wartości tworzyły wykres w przybliżeniu liniowy, inaczej błąd może być duży.



44. Znajdowanie elementów w zbiorze, czyli wyszukiwanie liniowe

NA TEJ LEKCJI:

- przypomnisz sobie, czym są algorytmy naiwne;
- dowiesz się, na czym polega wyszukiwanie liniowe.

Wielokrotnie zapewne przeszukujecie strony internetowe oferujące streaming w celu odnalezienia swojego ulubionego utworu, albumu danego wykonawcy czy podcastu ulubionego autora. O ile jest łatwiej i przyjemniej znaleźć interesujące nas utwory, jeżeli są one ułożone w określonym porządku.

44.1. Algorytmy wyszukiujące – wyszukiwanie liniowe

We współczesnej informatyce algorytmy wyszukiujące otrzymują od nas informacje o tym, czego szukamy, a one – po przepracowaniu problemu – jako dane wyjściowe zwracają nam pewną liczbę odpowiedzi. Oczywiście liczba i jakość uzyskanych odpowiedzi będzie zależna od tego, czego szukamy. Większość algorytmów, które mają za zadanie rozwiązać postawiony przed nimi problem, to algorytmy wyszukiujące.

algorytmy naiwne

Najprostszymi algorytmami wyszukiującymi są algorytmy naiwne, które podczas przetwarzania danych stosują intuicyjną metodę wyszukiwania w przestrzeni poszukiwań, czyli w zbiorze wszystkich możliwych rozwiązań. Problemem jest jednak ich wydajność. Opracowano bardziej zaawansowane metody, które wykorzystują wiedzę na temat samej przestrzeni poszukiwań. Dzięki temu mamy algorytmy, które potrzebują o wiele mniej czasu do przetworzenia danych.

wyszukiwanie liniowe

Wyszukiwanie liniowe w informatyce nazywane jest również wyszukiwaniem sekwencyjnym. Polega ono na przeglądaniu kolejnych elementów w zbiorze (tablicy), czyli na porównywaniu żądanego klucza z kolejnymi kluczami z sekwencji danych.

Wyszukiwanie zakończy się powodzeniem, jeżeli klucz zostanie odnaleziony, albo niepowodzeniem, jeżeli algorytm przeszuka wszystkie klucze, ale właściwy nie zostanie odnaleziony.



Liczba koniecznych przeszukań tablicy będzie zależała od położenia szukanego elementu i będzie wynosić od 1 do n , przy czym n jest liczbą całkowitą elementów, które będą przeszukiwane. Złożoność algorytmu wynosi $O(n)$, czyli jest liniowa. Stąd wywodzi się nazwa metody wyszukiującej.

Często przeszukujemy zbiór, aby otrzymać wszystkie wystąpienia szukanego przez nas elementu. W takim przypadku algorytm na wejściu powinien otrzymywać dodatkową pozycję elementu, od której mamy rozpocząć wyszukiwanie. Przy każdym kolejnym przeszukaniu pozycję tę będziemy zwiększali zawsze o 1 w stosunku do ostatniej znalezionej. Dzięki takiemu posunięciu nowe wyszukiwanie rozpocznie się od ostatnio znalezionej pozycji. Poniższy algorytm prezentuje najprostsze wyszukiwanie liniowe, które zatrzymuje swoją pracę po wyszukaniu pierwszego wystąpienia.

Lista kroków:

Dane wejściowe:

n – liczba elementów

$T[]$ – tablica, którą będziemy przeszukiwać, składająca się z n -elementów do posortowania

w – poszukiwana wartość w zbiorze

Dane wyjściowe:

i – pozycja elementu o wartości w ; jeżeli element nie zostanie wyszukany, to i przyjmie wartość n ,

Krok 1 $i = 0$

Krok 2 jeżeli $i \geq n$, to idź do Krok 7

Krok 3 jeżeli $T[i] = w$, to idź Krok 6

Krok 4 $i = i + 1$

Krok 5 idź do Krok 2

Krok 6 wyświetl wynik

Krok 7 koniec



ZADANIA DO WYKONANIA

1. Narysuj schemat blokowy przedstawiający wyszukiwanie liniowe (sekwencyjne).
2. Napisz program w języku C++, który będzie realizował wyszukiwanie liniowe.
3. Napisz program w języku Java, który będzie realizował wyszukiwanie liniowe.
4. Napisz program w języku C++, który wypełnia 50-elementową tablicę T całkowitymi liczbami pseudolosowymi z zakresu od 0 do 9, a następnie losuje nową liczbę pseudolosową z tego samego zakresu i znajduje jej wszystkie wystąpienia z tablicy T . Na końcu wyświetla wylosowaną liczbę oraz zawartość tablicy T z zaznaczonymi jej wystąpieniami.
5. Napisz program w języku Java, który wypełnia 100-elementową tablicę T całkowitymi liczbami pseudolosowymi z zakresu od 0 do 5, a następnie losuje nową liczbę pseudolosową z tego samego zakresu i znajduje jej wszystkie wystąpienia z tablicy T . Na końcu wyświetla wylosowaną liczbę oraz zawartość tablicy T z zaznaczonymi jej wystąpieniami.

PODSUMOWANIE LEKCJI

- s. 276** • Algorytmy naiwne to najprostsze algorytmy wyszukiujące, które podczas przetwarzania danych stosują intuicyjną metodę wyszukiwania w przestrzeni poszukiwań, czyli w zbiorze wszystkich możliwych rozwiązań.
- s. 276** • Wyszukiwanie liniowe w informatyce nazywane jest również wyszukiwaniem sekwencyjnym. Polega ono na przeglądaniu kolejnych elementów w zbiorze (tablicy), czyli na porównywaniu żadanego klucza z kolejnymi kluczami z sekwencji danych.



45. Szukanie połówek, czyli wyszukiwanie elementów poprzez połowienie

NA TEJ LEKCJI:

- dowiesz się, czym jest wyszukiwanie binarne, zwane również wyszukiwaniem poprzez połowienie;
- poznasz, w jaki sposób stworzyć i zastosować algorytm realizujący wyszukiwanie połówkowe.

Wyobraźmy sobie, że w słowniku języka angielskiego będziemy musieli odnaleźć słowo *programming* (programowanie). Oczywiście wyszukując je, możemy zacząć od słów na literę A i po kolei wertować kolejne strony słownika, aż dotrzemy do litery P. Prawdopodobnie nie będziecie tego robić w ten sposób, tylko losowo otworzycie słownik i sprawdzicie, czy jeszcze nie dotarliście do litery P, czy może już ją przekroczyliście.

A teraz wyobraź sobie, że logujesz się do Instagrama. Twoim loginem jest nazwa użytkownika *ideafix* – serwis może zacząć wyszukiwanie tego użytkownika od litery A, ale bardziej sensownie będzie, gdy zacznie od środka. W tych i podobnych przypadkach do wyszukania interesujących nas elementów możemy użyć wyszukiwania binarnego, które przyspieszy tę czynność.

45.1. Wyszukiwanie binarne, czyli wyszukiwanie poprzez dzielenie na pół

Wyszukiwanie binarne jest jednym z bardziej wydajnych algorytmów wyszukiwania określony element w uporządkowanej liście elementów. Jego działanie polega na wielokrotnym dzieleniu na pół przeszukiwanej listy. Czynność ta będzie wykonywana tak długo, aż nie znajdzie się wyszukiwany element.

Założmy, że mamy tablicę n -elementową T i chcemy w niej odnaleźć element x . Kolejnymi elementami naszej tablicy będą: $t[1]$, $t[2]$, $t[3]$, ..., $t[n-1]$, $t[n]$.

Jeżeli tablica T będzie zawierała posortowane elementy, to możemy do jej przeszukania użyć wyszukiwania binarnego, czyli połówkowego. Algorytm ten nosi również nazwę wyszukiwania przez połowienie.

Przyjmijmy, że elementy tablicy T są posortowane rosnąco. Założmy, że początkowy indeks tablicy oznaczony jest literą p i przyjmuje wartość 1 ($p = 1$), a końcowy – literą k i przyjmuje wartość n ($k = n$). Dzielimy tablicę na pół elementem o indeksie równym $s = (p + k)/2$, przy czym zakładamy, że jest to dzielenie bez reszty, ponieważ chcemy uzyskać w wyniku liczbę całkowitą. Następnie sprawdzamy,

wyszukiwanie
binarne



czy element o indeksie s jest tym, którego szukamy. Jeżeli tak, to nasz algorytm kończy w tym miejscu swoją pracę, a jeżeli nie – musimy sprawdzić, czy element o indeksie s jest większy od szukanego. Jeżeli jest większy, to mamy pewność, że elementy $s + 1, s + 2, \dots, k$ również są większe – wynika to z tego, że tablica T jest posortowana rosnąco. Natomiast jeżeli element o indeksie s jest mniejszy od szukanego, to wiemy, że również elementy o indeksach $s - 1, s - 2, \dots, p$ będą mniejsze. Wynika to również z faktu, że tablica jest posortowana rosnąco. Ustawiamy więc indeks początku tablicy p na wartość $s + 1$ i powtarzamy jej dzielenie na pół w ten sam sposób. Czynności te będziemy powtarzali tak długo, aż znajdziemy szukany element bądź wskaźnik początku tablicy p będzie większy od wskaźnika końca tablicy k – będzie to wtedy oznaczać, że w naszej tablicy nie ma szukanego elementu.

Algorytm wyszukiwania binarnego – lista kroków

Wejście

p – indeks pierwszego elementu tablicy T – $p \in \mathbb{C}$

k – indeks ostatniego elementu tablicy T – $k \in \mathbb{C}$

T – tablica do wyszukiwania, posortowana rosnąco od p do k

x – wartość szukanego elementu

Wyjście

r = indeks elementu o wartości x lub

$r = -1$, jeżeli szukany element nie zostanie znaleziony

Zmienne pomocnicze

s – indeks środkowego elementu – $s \in \mathbb{C}$

Krok 1 $r = -1$

Krok 2 dopóki $p \leq k$ wykonuj kroki od Krok 2 do Krok 10

Krok 3 $s = ((p+k)/2)$

Krok 4 jeżeli $x \neq T[s]$, to idź do Krok 7

Krok 5 $p = s$

Krok 6 idź do Krok 11

Krok 7 jeżeli $x < T[s]$, to idź do Krok 10

Krok 8 $p = s + 1$

Krok 9 kolejny obieg pętli od Krok 2

Krok 10 $k = s - 1$

Krok 11 zakończ wynikiem r



ZADANIA DO WYKONANIA

1. Narysuj schemat blokowy przedstawiający wyszukiwanie połówkowe (binarne).
2. Napisz program w języku C++, który będzie realizował wyszukiwanie połówkowe (binarne) w zbiorze uporządkowanym rosnącym.
3. Napisz program w języku C++, który będzie realizował wyszukiwanie połówkowe (binarne) w zbiorze uporządkowanym rosnącym. Dodatkowo program powinien zliczać liczbę obiegów pętli w algorytmie i po zakończeniu wyświetlać tę liczbę.
4. Napisz program w języku Java, który będzie realizował wyszukiwanie połówkowe (binarne) w zbiorze uporządkowanym rosnącym.
5. Napisz program w języku Java, który będzie realizował wyszukiwanie połówkowe (binarne) w zbiorze uporządkowanym rosnącym. Dodatkowo program powinien zliczać liczbę obiegów pętli w algorytmie i po zakończeniu wyświetlać tę liczbę.
6. Napisz program w języku C++, który wypełnia 5-elementową tablicę T całkowitymi liczbami z zakresu od 0 do 9, a następnie sprawdź, czy występuje w niej wartość równa liczbie zadanej przez użytkownika z zakresu od 0 do 9.
7. Napisz program w języku Java, który wypełnia 8-elementową tablicę T całkowitymi liczbami pseudolosowymi z zakresu od 0 do 5, a następnie sprawdź, czy występuje w niej wartość równa liczbie zadanej przez użytkownika z zakresu od 0 do 4.

PODSUMOWANIE LEKCJI

- Wyszukiwanie binarne jest jednym z bardziej wydajnych algorytmów wyszukujących określony element w uporządkowanej liście elementów. Jego działanie polega na wielokrotnym dzieleniu na pół przeszukiwanej listy. Czynność ta będzie wykonywana tak długo, aż nie znajdzie się wyszukiwany element.

s. 279



46. Ustawianie kart, czyli sortowanie przez wstawianie

NA TEJ LEKCJI:

- przypomnisz sobie metody sortowania ciągu elementów;
- dowiesz się, czym jest sortowanie poprzez wstawianie.

Pewnie zdarzało się wam grać w karty – makao, tysiąca itp. W momencie, kiedy otrzymujemy zestaw kart, pierwszą czynnością jest często układanie ich od najmniej ważnej do najważniejszej. Robimy to zazwyczaj, wkładając karty między inne. Właśnie podczas tej czynności sortujemy je metodą poprzez wstawianie.

46.1. Sortowanie przez wstawianie

sortowanie
poprzez
wstawianie

Algorytm sortowania przez wstawianie jest jednym z najprostszych algorytmów sortujących. Jego działanie przypomina ustawianie kart w kolejności. Najpierw bierzemy pierwszą kartę, a następnie pobieramy kolejne, aż do wyczerpania talii. Każdą z pobieranych kart porównujemy z tymi, które już mamy, i staramy się przyporządkować jej miejsce przed kartą starszą (bądź młodszą, jeżeli będziemy je układać w kolejności malejącej). Gdy znajdziemy odpowiednie miejsce, wstawiamy w nie daną kartę, rozsuwając już ułożone karty – stąd nazwa *algorytm wyszukujący*. Jeżeli nasza karta będzie najstarszą, to umieszczamy ją na samym końcu.

Algorytm ten będzie efektywny dla niewielkiej liczby elementów sortowanych, a jego złożoność wyniesie $O(n^2)$.

Prześledźmy działanie algorytmu dla następującego ciągu liczb: 3, 6, 4, 1, 7, 2.

Sortowanie rozpoczniemy od drugiego elementu – 6 – i porównamy go z elementem poprzedzającym. W naszym przypadku liczbą poprzedzającą jest 3. Jeżeli napotkamy większą od naszej, to musimy ją przesunąć o jeden w prawo. Czynność tę będziemy powtarzać do momentu napotkania liczby mniejszej lub skończenia zasobu liczb, czyli gdy nie będzie spełniony już warunek $k \geq 0$.

Kolejnym krokiem będzie wstawienie naszej liczby w odpowiednie miejsce. W ten sposób otrzymamy podzbiór liczb uporządkowanych. Czynność powtarzamy z pozostałymi elementami naszego ciągu:

3, 6, 4, 1, 7, 2 – liczba 6 pozostaje na swoim miejscu (dwa pierwsze elementy mamy posortowane)

3, 4, 6, 1, 7, 2 – liczba 4 zostaje wstawiona między liczby 3 i 6 (trzy pierwsze elementy mamy posortowane)



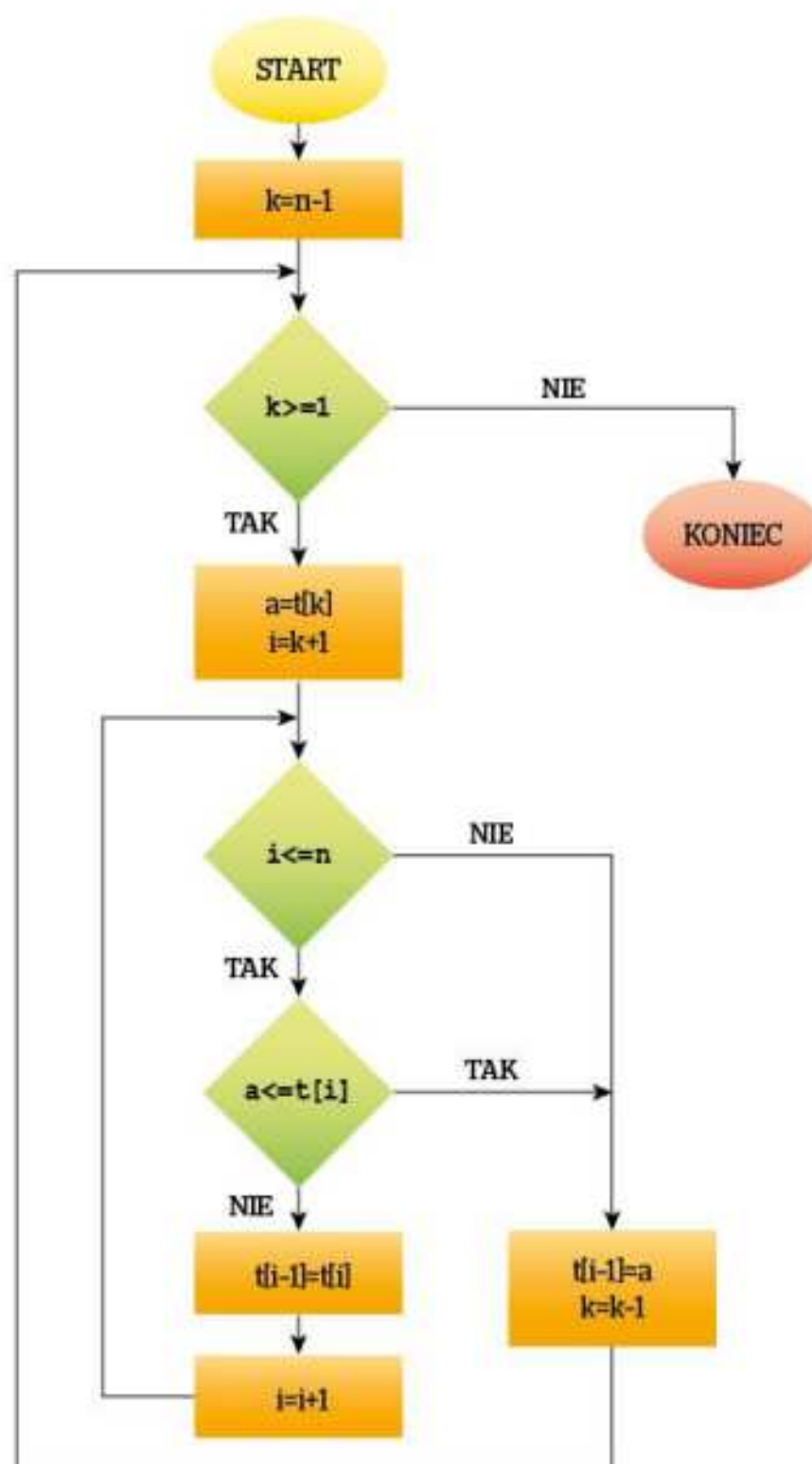
1, 3, 4, 6, 7, 2 – liczba 1 zostaje ustawiona na samym początku (cztery pierwsze elementy mamy posortowane)

1, 3, 4, 6, 7, 2 – liczba 7 pozostaje na swoim miejscu (pięć pierwszych elementów mamy posortowanych)

1, 2, 3, 4, 6, 7 – liczba 2 zostaje wstawiona między liczby 1 i 3 – otrzymujemy zbiór z wszystkimi uporządkowanymi elementami

Oczywiście algorytm ten nie należy do najszybszych, ale za to jest stabilny. Bardzo dobrze będzie się zachowywał, gdy będziemy stosowali go w zbiorze częściowo posortowanym.

Schemat blokowy



Rys. 46.1. Schemat blokowy sortowania przez wstawianie



Algorytm składa się z dwóch pętli. Pierwsza pobiera elementy t z tablicy. Na końcu zbioru będziemy tworzyli listę uporządkowaną. Pętla sortująca (wewnętrzna) wyszukuje miejsca na uporządkowanej liście dla elementu pobranego z tablicy. Jeżeli takie miejsce zostanie znalezione, to elementy listy będą w odpowiedni sposób rozsunięte, aby utworzyły miejsce dla nowego elementu wybranego przez pętlę sortującą. Po umiejscowieniu elementu czynność zostaje wykonana ponownie, a lista rozrasta się w sposób uporządkowany. Jeżeli na liście nie będzie elementu większego od tego, który został wybrany, to trafia on na koniec listy. Sortowanie zostanie ukończone, gdy pierwsza pętla wybierze wszystkie elementy.

ZADANIA DO WYKONANIA

1. Napisz program w języku C++, który ułoży elementy listy malejąco, wykorzystując algorytm sortowania poprzez wstawianie.
2. Napisz program w języku C++, który ułoży elementy listy rosnąco, wykorzystując algorytm sortowania poprzez wstawianie.
3. Napisz program w języku Java, który ułoży elementy listy malejąco, wykorzystując algorytm sortowania poprzez wstawianie.
4. Napisz program w języku Java, który ułoży elementy listy rosnąco, wykorzystując algorytm sortowania poprzez wstawianie.
5. Napisz program w języku C++, który ułoży elementy listy rosnąco, wykorzystując algorytm sortowania poprzez wstawianie.
6. Napisz program w języku C++, który ułoży elementy listy malejąco, wykorzystując algorytm sortowania poprzez wstawianie.
7. Napisz program w języku Java, który ułoży elementy listy rosnąco, wykorzystując algorytm sortowania poprzez wstawianie.
8. Napisz program w języku Java, który ułoży elementy listy malejąco, wykorzystując algorytm sortowania poprzez wstawianie.

PODSUMOWANIE LEKCJI

s. 282

- Sortowanie przez wstawianie jest to prosty algorytm sortowania polegający na wstawianiu kolejnych elementów ciągu we właściwe miejsca. Jest to algorytm realizujący metodę przyrostową.



47. Regula falsi, czyli przybliżone rozwiązywanie równań

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest określoność funkcji oraz ciągłość funkcji;
- dowiesz się czym jest *regula falsi*;
- poznasz algorytm umożliwiający obliczenie przybliżonej wartości pierwiastka metodą *regula falsi*.

Spotykamy się często z funkcjami, dla których trudno obliczyć pierwiastki, czyli miejsca przecięcia funkcji z osią OX . Z pomocą w takich sytuacjach przychodzą nam metody, które umożliwiają obliczenie przybliżonych wartości tych pierwiastków. Jedną z nich jest *regula falsi*, czyli metoda fałszywej prostej.

47.1. Regula falsi

Regula falsi w języku łacińskim oznacza fałszywą linię prostą. Nazwa bierze się stąd, że jeżeli będziemy badać funkcję w coraz węższych przedziałach, to zacznie ona przypominać wykres linii prostej. Algorytm *regula falsi* ma takie same warunki początkowe, jak algorytm wyszukiwania binarnego.

Mamy daną funkcję $f(x)$ oraz przedział $[a, b]$, w którym będziemy szukać pierwiastka naszej funkcji. Abyśmy mogli zastosować algorytm *regula falsi*, nasza funkcja musi spełniać następujące warunki:

1. Funkcja $f(x)$ jest funkcją określoną, czyli dla każdej wartości argumentu x z przedziału $[a, b]$ będziemy mogli policzyć wartość funkcji.
2. Funkcja $f(x)$ jest ciągła, czyli przebiega przez wszystkie pośrednie wartości.
3. Funkcja $f(x)$ na krańcach przedziału $[a, b]$ przyjmuje różne znaki. Funkcja $f(x)$ w przedziale $[a, b]$ będzie przyjmowała wszystkie wartości pośrednie między $f(a)$ i $f(b)$, ponieważ z założenia musi być funkcją ciągłą. Wartości te będą miały różne znaki – czyli będą leżały po różnych stronach osi OX . Musi więc być taki punkt x_0 w przedziale $[a, b]$, dla którego funkcja ta przyjmie wartość pośrednią:

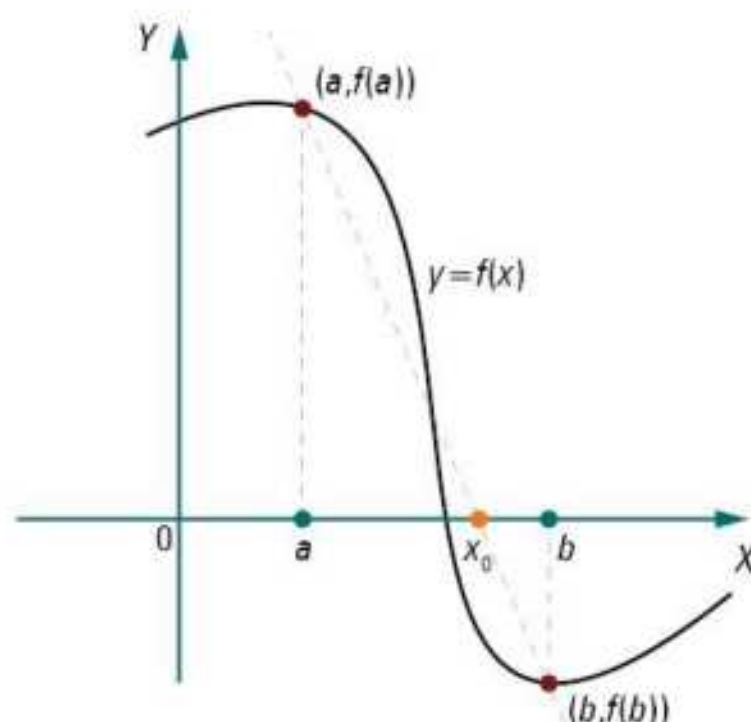
$$f(a) < f(x_0) = 0 < f(b) \text{ lub } f(a) > f(x_0) = 0 > f(b)$$

W przypadku, gdy funkcja $f(x)$ będzie spełniała powyższe warunki, w przedziale $[a, b]$ będzie istniał pierwiastek oraz będziemy mogli go wyszukać algorytmem *regula falsi*.

regula falsi

określoność
funkcji

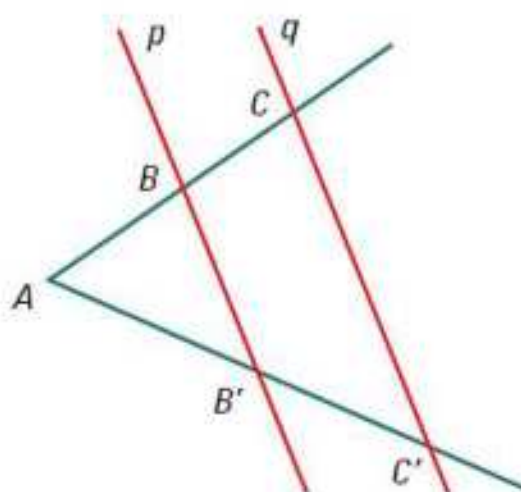
ciągłość funkcji



Rys. 47.1. Warunki spełniające wyszukanie pierwiastka metodą *regula falsi*

Ideą metody *regula falsi*, czyli fałszywej prostej, jest założenie, że funkcja w coraz mniejszych przedziałach, które przybliżają się do pierwiastka, zaczyna przypominać funkcję liniową. Skoro tak się dzieje, to przybliżenie pierwiastka będziemy mogli otrzymać, przeprowadzając prostą z punktów krańcowych przedziałów. Sieczna przetnie oś OX w punkcie x_0 , który będziemy mogli przyjąć jako przybliżenie pierwiastka.

Wzór dla przybliżonego pierwiastka x_0 można wyprowadzić z twierdzenia Talesa – jeżeli ramiona kąta przetniemy dwiema prostymi równoległymi, to długości odcinków wyznaczonych przez te proste na jednym ramieniu kąta będą proporcjonalne do długości odcinków wyznaczonych przez te proste na drugim ramieniu:



Rys. 47.2. Twierdzenie Talesa

$$\frac{|AB|}{|A'B'|} = \frac{|BC|}{|B'C'|} = \frac{|AC|}{|A'C'|}$$



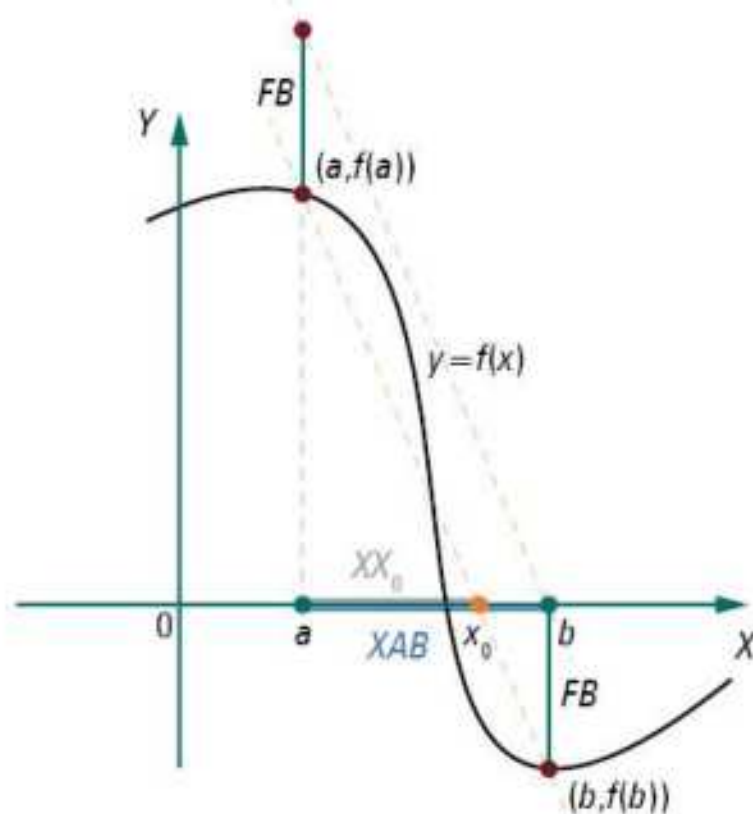
Jeżeli twierdzenie Talesa zastosujemy do naszego przypadku, to musimy postępować następująco:

1. Kąt zostanie utworzony przez następujące odcinki:

pionowo $FA = (a, 0) - (a, f(a))$ powiększony o odcinek $FB = (b, 0) - (b, -f(b))$

poziomo natomiast $XAB = (a, 0) - (b, 0)$.

Prostymi równoległymi będą cięciwy – łącząca krańcowe punkty przedziału oraz przesunięta pionowo w górę o długość odcinka FB .



Rys. 47.3. Twierdzenie Talesa zastosowane w badanej funkcji $y=f(x)$

Zgodnie z powyższym możemy zapisać:

$$\frac{FA - FB}{XAB} = \frac{FA}{XX_0}$$

Podstawmy do naszych danych odcinki:

$$FA = f(a)$$

$$FB = -f(b)$$

$$XAB = b - a$$

$$XX_0 = x_0 - a$$

Po podstawieniu do wzoru długości odcinków otrzymamy:

$$\frac{f(a) - f(b)}{b - a} = \frac{f(a)}{x_0 - a}$$



Obliczmy zatem x_0 :

$$(x_0 - a) \cdot \frac{f(a) - f(b)}{b - a} = f(a)$$

$$x_0 - a = f(a) \cdot \frac{b - a}{f(a) - f(b)}$$

$$x_0 = a + f(a) \cdot \frac{b - a}{f(a) - f(b)}$$

$$x_0 = a - f(a) \cdot \frac{b - a}{f(b) - f(a)}$$

47.2. Algorytm *regula falsi*

W algorytmie *regula falsi*, w przeciwieństwie do algorytmu wyszukiwania binarnego, punkt x_0 będzie wyznaczany w zależności od tego, jakie wartości funkcja ma na krańcach przedziału. Dzięki temu w każdej iteracji będziemy mogli otrzymać bardziej szczegółowe przybliżenie do rzeczywistej wartości pierwiastka.

Dane wejściowe:

$f(x)$ – funkcja, której pierwiastek będziemy obliczać; funkcja ta musi spełniać trzy warunki podane w poprzednim podrozdziale, abyśmy mogli zastosować metodę *regula falsi*

a, b – punkty krańcowe przedziału, w których będziemy szukać pierwiastka funkcji – $a, b \in R$

Dane wyjściowe:

x_0 – szukany pierwiastek funkcji $f(x)$

Zmienne pomocnicze:

fa, fb, f_0 – wartości funkcji odpowiednio w punktach a, b i x_0 – $fa, fb, f_0 \in R$

x_1 – przechowuje poprzednią wartość x_0

e_0 – określa dokładność porównania z zerem

e_x – określa dokładność wyznaczania pierwiastka x_0

Lista kroków

Krok 1 wczytaj a i b

Krok 2 $fa = f(a); fb = f(b);$

$$x_1 = a; x_0 = b$$

Krok 3 jeżeli $fa \cdot fb > 0$, to wypisz: "Funkcja nie spełnia założeń" i przejdź do Krok 10

Krok 4 dopóki $|x_1 - x_0| > e_x$, to wykonuj kroki od Krok 5 do Krok 8

Krok 5 $x_1 = x_0$

Krok 6 $x_0 = a - fa \cdot ((b - a)/(fb - fa)); f_0 = f(x_0)$

Krok 7 jeżeli $|f_0| < e_0$, to idź do Krok 9

Krok 8 jeżeli $fa \cdot f_0 < 0$, to $b = x_0; fb = f_0$, w przeciwnym wypadku $a = x_0; fa = f_0$

Krok 9 wypisz x_0

Krok 10 zakończ



ZADANIA DO WYKONANIA

1. Narysuj schemat blokowy, który wyszuka przybliżoną wartość pierwiastka funkcji $y = f(x)$ z zastosowaniem metody *regula falsi*.
2. Napisz program w języku C++, który będzie wyszukiwał przybliżoną wartość pierwiastka funkcji $y = f(x)$ z zastosowaniem metody *regula falsi*.
3. Napisz program w języku Java, który będzie wyszukiwał przybliżoną wartość pierwiastka funkcji $y = f(x)$ z zastosowaniem metody *regula falsi*.

PODSUMOWANIE LEKCJI

- *Regula falsi* to algorytm rozwiązywania równań nieliniowych jednej zmiennej.
- Określoność funkcji oznacza, że dla każdego argumentu x z przedziału $[a, b]$ istnieje wartość funkcji.
- Funkcja $f(x)$ jest ciągła, czyli przebiega przez wszystkie pośrednie wartości.

s. 285

s. 285

s. 285



48. Geometria obliczeniowa, czyli sprawdzanie przynależności punktu do wielokąta wypukłego

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest wielokąt;
- powtórzysz informacje o wielokątach wklęsłych i wypukłych;
- dowiesz się, w jaki sposób sprawdzić, czy punkt przynależy do wielokąta wypukłego.

W latach 70. XX wieku wyodrębniono geometrię obliczeniową jako samodzielny dział algorytmiki. Stało się tak dlatego, że uzyskiwane w jej ramach wyniki mają istotne znaczenie w wielu dziedzinach informatyki i inżynierii, takich jak robotyka, symulacje komputerowe, grafika komputerowa, projektowanie wspomagane komputerowo czy też bazy danych.

Najważniejszym zadaniem geometrii obliczeniowej jest wyszukiwanie pojęć i rozwiązywanie problemów przy tworzeniu algorytmów związanych z samą geometrią.

Geometria obliczeniowa jest działem algorytmiki zajmującym się algorytmami i strukturami danych, które pozwalają na efektywne wykonywanie działań na obiektach geometrycznych, takich jak odcinki, wielokąty, okręgi czy zbiory punktów.

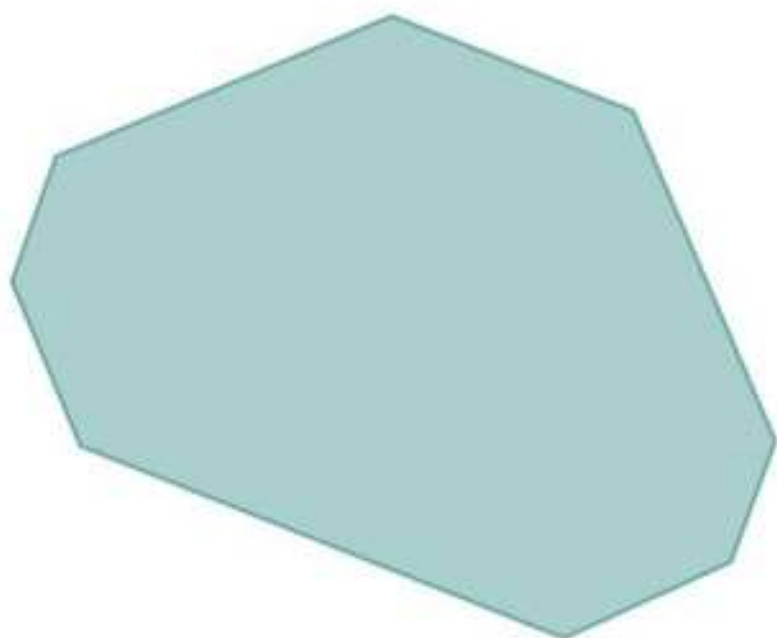
48.1. Sprawdzenie przynależności punktu do wielokąta wypukłego

wielokąt

Na początku przypomnijmy sobie, czym jest wielokąt. Wielokątem, bądź też wielobokiem, będziemy nazywać zbiór punktów płaszczyzny, który jest ograniczony przez łamaną, zazwyczaj zamkniętą, wraz z tą łamaną. Rozróżniamy dwa rodzaje wielokątów – wielokąty wklęsłe i wypukłe.

wielokąt wypukły

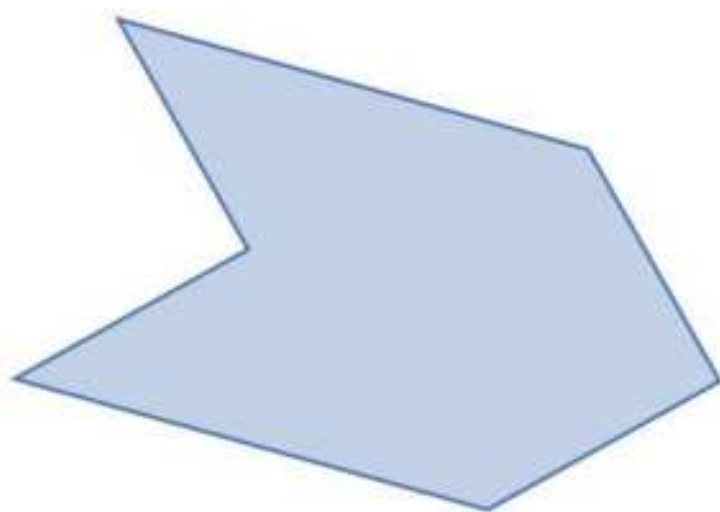
Wielokąt wypukły to taki, który jest figurą geometryczną wypukłą, czyli wtedy i tylko wtedy, gdy wszystkie jego kąty zewnętrzne są wypukłe lub wszystkie przekątne zawierają się w tym wielokącie.



Rys. 48.1. Wielokąt wypukły

Wielokątem wklęsłym jest wielokąt, którego co najmniej jedna przekątna nie zawiera się w nim, a także co najmniej jeden kąt wewnętrzny ma miarę większą od kąta półpełnego.

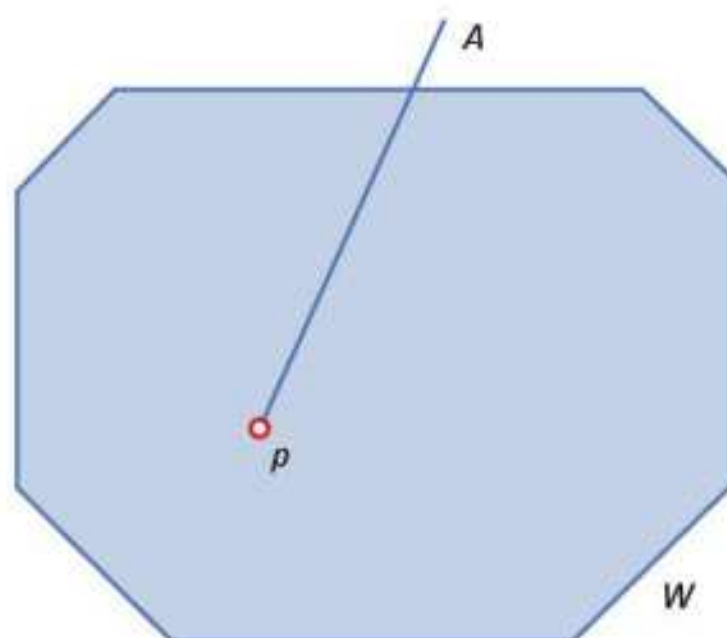
wielokąt wklęsły



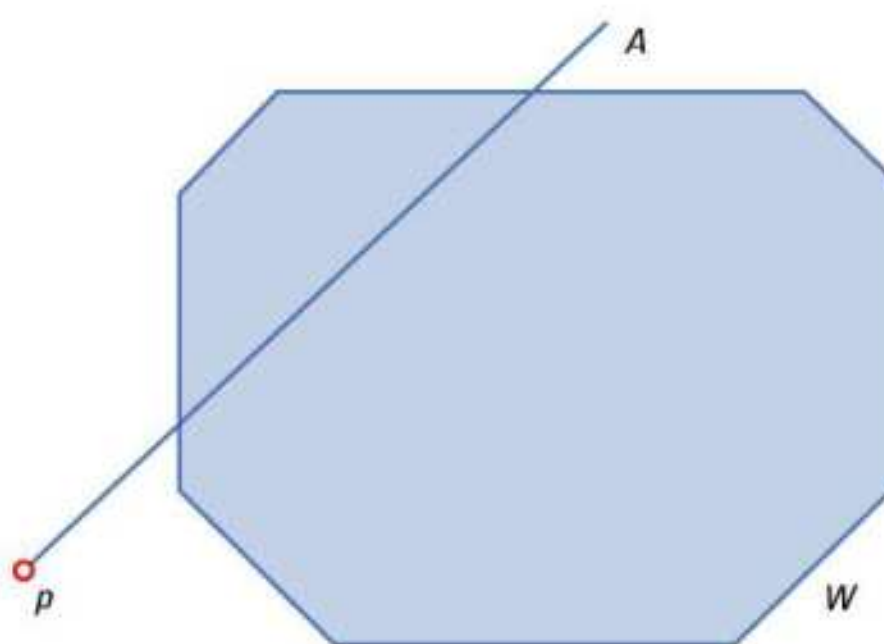
Rys. 48.2. Wielokąt wklęsły

Sprawdzenie, czy punkt należy do wielokąta, nie jest czynnością skomplikowaną. Istnieje bardzo prosty sposób na to, aby zbadać, czy punkt należy do prostokąta. Wystarczy sprawdzić, czy rzuty prostokątne tego punktu na osie OX i OY wpadają do rzutów prostokątnych boków tego prostokąta. Sprawa trochę się komplikuje, gdy chcemy sprawdzić, czy punkt należy do wielokąta. I tutaj na pomoc przychodzi geometria obliczeniowa. Musimy zastosować kryterium parzystości.

Mamy dany wielokąt W oraz punkt p . Z punktu p kreślimy półprostą A i sprawdzamy, ile razy przetnie ona boki wielokąta. Jeżeli półprosta A przetnie wielokąt nieparzystą liczbę razy, to będzie oznaczało, że punkt należy do wielokąta, w przeciwnym razie punkt p nie będzie należał do wielokąta W .



Rys. 48.3. Punkt należy do wielokąta – półprosta przecina wielokąt nieparzystą liczbę razy



Rys. 48.4. Punkt nie należy do wielokąta – półprosta przecina wielokąt parzystą liczbę razy



ZADANIA DO WYKONANIA

1. Napisz algorytm w postaci listy kroków, który będzie badał, czy punkt należy do wielokąta wypukłego.
2. Narysuj algorytm w formie schematu blokowego, który będzie badał, czy punkt należy do wielokąta wypukłego.
3. Opisz problem i napisz algorytm w postaci listy kroków w przypadku, gdy punkt znajduje się na jednym z boków wielokąta.
4. Napisz program w języku C++, który będzie badał, czy punkt należy do wielokąta wypukłego.
5. Napisz program w języku Java, który będzie badał, czy punkt należy do wielokąta wypukłego.

PODSUMOWANIE LEKCJI

- Wielokątem bądź wielobokiem nazywamy zbiór punktów płaszczyzny, który jest ograniczony przez łamaną, zazwyczaj zamkniętą, wraz z tą łamaną. s. 290
- Wielokąt wypukły to taki, który jest figurą geometryczną wypukłą, czyli wtedy i tylko wtedy, gdy wszystkie jego kąty zewnętrzne są wypukłe lub wszystkie przekątne zawierają się w tym wielokącie. s. 290
- Wielokątem wklęsłym jest wielokąt, którego co najmniej jedna przekątna nie zawiera się w nim, a także co najmniej jeden kąt wewnętrzny ma miarę większą od kąta półpełnego. s. 291



49. Algorytm wyszukiujący, czyli liniowe przeszukiwanie ciągu w poszukiwaniu żadanego elementu z wykorzystaniem wartownika

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest algorytm wyszukiujący i algorytm naiwny;
- dowiesz się, co to jest przestrzeń poszukiwań;
- poznasz sposób działania algorytmu wyszukiwania z wartownikiem.

Umiejętność wyszukiwania informacji jest najbardziej cenioną umiejętnością wśród przyszłych pracowników. To informacja jest dzisiaj najbardziej pożądanym i cenionym towarem na świecie.

Wyszukiwanie informacji w informatyce to podstawowy element algorytmiki. Wyszukiwanie i przeszukiwanie tablic czy ciągów jest istotnym aspektem programowania.

49.1. Algorytmy wyszukiujące

algorytm
wyszukujący

Algorytm wyszukiujący na wejściu dostaje problem, który musi rozwiązać, wraz z informacjami początkowymi (wejściowymi), a na wyjściu oddaje rozwiązanie. Zajmuje się on wyszukiwaniem i przeszukiwaniem danych, które mu dostarczyliśmy, aby na koniec (na wyjściu) podać rozwiązanie lub jego kilka wariantów.

przestrzeń
poszukiwań

Zbiór rozwiązań, który otrzymamy, to przestrzeń poszukiwań. Najprostszymi algorytmami wyszukiującymi są algorytmy naiwne. Bardziej złożone metody wyszukiwań, które działają nie tak intuicyjnie, jak algorytmy naiwne, wykorzystują wiedzę na temat samej przestrzeni wyszukiwań. Dzięki ich konstrukcji i działaniu możemy skrócić czas potrzebny do przetworzenia danych.

wyszukiwanie
liniowe

Wyszukiwanie liniowe to przeszukiwanie tablicy bądź zbioru element po elemencie. Musimy pamiętać, aby nie wychodzić poza zakres tablicy – szukamy elementu, który spełnia nasze warunki.

Wyszukiwanie liniowe zobrazujemy za pomocą listy kroków.

Dane wejściowe:

n – liczba naturalna, taka że $n > 0$ (liczba elementów tablicy T)



s – szukana liczba całkowita (element przeszukiwanej tablicy T)

$T[]$ – tablica n -elementowa zawierająca elementy od 0 do $n - 1$

Dane wyjściowe:

p – pozycja elementu o wartości s ; jeżeli element nie zostanie odnaleziony, to p przyjmie wartość n

Lista kroków:

Krok 1 $p = 0$

Krok 2 jeżeli $p \geq n$, to idź do Krok 6

Krok 3 jeżeli $T[p] = s$, to idź do Krok 6

Krok 4 $p = p + 1$

Krok 5 idź do Krok 2

Krok 6 zakończ

49.2. Wyszukiwanie liniowe z wartownikiem

Do algorytmów, które umożliwiają wyszukanie konkretnego elementu z tablicy, a stosują bardziej złożone metody wyszukiwania, zaliczamy algorytm wyszukiwania liniowego z wartownikiem.

Zadaniem tego algorytmu jest znalezienie elementu w tablicy bądź zbiorze. Tym razem jednak powiększymy tę tablicę (zbiór) o jeden element zwany wartownikiem, któremu nadamy wartość szukanego przez nas elementu. Wartownik znajdzie się na pozycji n w tej tablicy. Jeżeli algorytm go znajdzie, to zwróci nam wartość n jako brak poszukiwanego elementu. Natomiast jeżeli znaleziona pozycja będzie mniejsza od n , to wskaże na rzeczywisty element zbioru. Dodany przez nas element – wartownik – ma za zadanie niedopuszczenie do tego, aby algorytm wyszedł poza elementy przeszukiwanej tablicy.

Lista kroków przedstawiająca ten algorytm będzie wyglądała następująco:

Dane wejściowe:

n – liczba naturalna, gdzie $n > 0$ (liczba elementów tablicy T)

s – szukana liczba całkowita (element przeszukiwanej tablicy T)

$T[]$ – tablica $n + 1$, n -elementowa zawierająca elementy od 0 do $n - 1$

Dane wyjściowe:

p – pozycja elementu o wartości s ; jeżeli element nie zostanie przez nas odnaleziony, to p przyjmie wartość n

Lista kroków:

Krok 1 $T[n] = s$

Krok 2 $p = 0$

Krok 3 jeżeli $T[p] = s$, to idź do Krok 6

Krok 4 $p = p + 1$

Krok 5 idź do Krok 3

Krok 6 zakończ

wyszukiwanie
liniowe
z wartownikiem



ZADANIA DO WYKONANIA

1. Napisz algorytm (w postaci listy kroków), który będzie realizował wyszukiwanie liniowe liczby z tablicy n -elementowej, rozpoczynając wyszukiwanie od pozycji q , a nie od 0.
2. Napisz program w języku C++, którego zadaniem będzie sprawdzenie, czy w tablicy n -elementowej wszystkie jej elementy są większe od 7.
3. Napisz program w języku Java, którego zadaniem będzie wyszukanie w tablicy n -elementowej takich jej elementów, które są podzielne przez 3 bez reszty.
4. Napisz program w języku C++, który wyszuka w tablicy n -elementowej wszystkie elementy, które będą większe od 5, a mniejsze od 11.

PODSUMOWANIE LEKCJI

- s. 294 • Algorytm wyszukiwający to taki, którego zadaniem jest znalezienie w zbiorze lub tablicy szukanego elementu.
- s. 294 • Przestrzeń poszukiwań w algorytmice to wynik działania algorytmu wyszukiwającego.
- s. 294 • Wyszukiwanie liniowe to przeszukiwanie tablicy bądź zbioru element po elemencie.
- s. 295 • Zadaniem algorytmu wyszukiwającego z wartownikiem jest znalezienie w tablicy szukanego elementu. Do przeszukiwanej tablicy jest dodawana szukana wartość – tzw. wartownik. Jego zadaniem jest niedopuszczenie do tego, aby algorytm wyszedł poza elementy przeszukiwanej tablicy.



50. Podejście zachłanne, czyli problem plecakowy

NA TEJ LEKCJI:

- dowiesz się, czym jest problem plecakowy;
- poznasz metodę zachłanną rozwiązywania problemu plecakowego.

Osoby zajmujące się handlem, zwłaszcza ludzie sprzedający swoje produkty na targowiskach, często mają dylemat, jak się spakować, aby zabrać ze sobą jak najwięcej towaru. Z taką samą sytuacją spotykamy się również w algorytmice. Teraz się dowiemy, jak ten problem rozwiązać.

50.1. Problem plecakowy – metoda zachłanna

Wróćmy do sprzedawcy z targowiska. Udając się rano do pracy, ma on dylemat, jak się spakować, aby zabrać jak najwięcej towaru, a zarazem, aby to był towar o jak największej wartości – to jest właśnie ogólny problem plecakowy. Pakując się, bardzo dobrze wie, ile ma towaru oraz jaka jest jego wartość i waga.

ogólny problem
plecakowy

Stosując metodę zachłanną, pakujemy plecak w taki sposób, aby włożyć tam przedmioty o jak największej wartości. Niestety, może się zdarzyć, że do plecaka włożymy tylko jeden przedmiot. Zastanówmy się, czy nie warto wziąć ze sobą więcej lżejszych przedmiotów, które w sumie będą miały większą wartość niż ten jeden o najwyższej wartości. Musimy sprawdzić, co będzie nam się bardziej opłacało. Zastosujmy wzór – iloraz wartości przedmiotu i jego wagi. Oczywiście nie otrzymamy optymalnego rozwiązania, ale będziemy mieli zagwarantowane, że spakujemy towary, które przyniosą nam największy zysk.

metoda zachłanna

Przyjmijmy, że mamy cztery różne elementy, których wartość i waga się od siebie różnią.

Przedmiot	A	B	C	D
Wartość	3	2	1,5	1
Waga	2	1,5	3	5
Opłacalność	1,5	1,33	0,5	0,2

Z powyższej tabeli wynika, że powinniśmy zabrać jak najwięcej przedmiotów oznaczonych literą A, a jak najmniej tych oznaczonych literą D, ponieważ są ciężkie i niewiele warte.



Rozpocznijmy pakowanie. Przyjmijmy, że możemy do plecaka spakować towar o maksymalnej wadze 17: $W = 17$. Sprawdźmy zatem, ile sztuk najbardziej opłacalnego towaru jesteśmy w stanie zapakować. W naszym przypadku będzie to 8 sztuk. Symulując wkładanie przedmiotów, musimy pamiętać, aby zmniejszyć wartość W o liczbę włożonych sztuk oraz zwiększyć wartość Z spakowanych przedmiotów: $W = 17 - 16 = 1$, a $Z = 8 * 3 = 24$.

W plecaku nie zmieścimy więcej przedmiotów oznaczonych literą A , dlatego te same obliczenia powtórzymy dla kolejnych przedmiotów. Kolejnym, mniej opłacalnym przedmiotem będzie ten z liczbą B . Do plecaka wejdzie do plecaka dokładnie jedna sztuka. Nie włożymy więcej, ponieważ nasze W wyniesie 0 (przedmiot B waży 1,5). Wartość naszego plecaka wzrośnie o 2, czyli $Z = 29$. Takim oto sposobem spakowaliśmy optymalnie nasz plecak.

Przeanalizujmy inny przykład w tzw. metodzie nieoptymalnej. Mamy dwa przedmioty A i B , których wartość i waga są przedstawione w poniższej tabeli.

Przedmiot	A	B
Wartość	4	3
Waga	2	1,5
Opłacalność	2	2

Opłacalność każdego z przedmiotów jest identyczna, natomiast różna jest ich wartość. Jeżeli przyjmimy, że nasze $W = 9$, to do plecaka zapakujemy 4 przedmioty A i ani jednego przedmiotu B – przez co wartość plecaka $Z = 16$. Natomiast gdybyśmy wzięli pod uwagę przedmiot B , to zapakowalibyśmy 6 sztuk, a wartość naszego plecaka $Z = 18$.

50.2. Program realizujący metodę zachłanną

Niech funkcja `spakujPlecak()` otrzyma następujące argumenty:

Dane wejściowe:

`l_waga` – lista zawierająca wagi n -elementów

`l_wartosc` – lista zawierająca wartości n -elementów

`n` – liczba przedmiotów do spakowania

`W` – maksymalna waga plecaka

```
double spakujPlecak(double* l_waga, double* l_wartosc,
int n, double W) {
    sortuj(l_waga, l_wartosc, n);
    double W = 0;
    int s = 0
    for (int i = 0; i < n; i++){
        s = W/l_waga[i];
        W -= s * l_waga[i];
    }
```



```

        Z += s * l_wartosc[i];
    }
    return Z;
}

```

Ponieważ danych nie damy rady posortować względem jednej listy, musimy zastosować niestandardowe sortowanie, które uwzględni wartości z obu list:

```

void sortowanie(double* l_waga, double* l_wartosc, int n){
    double q;
    for(int i = 0; i < n - 1; i++){
        for(int j = 0; j < n - 1; j++){
            if(l_wartosc[j]/l_waga[j] <=
l_wartosc[j+1]/l_waga[j+1]){
                q = l_waga[j];
                l_waga[j] = l_waga[j+1];
                l_waga[j+1] = q;
                q = l_wartosc[j];
                l_wartosc[j] = l_wartosc[j+1];
                l_wartosc[j+1] = q;
            }
        }
    }
}

```

ZADANIA DO WYKONANIA

1. Napisz listę kroków, która będzie realizowała algorytm pakowania plecaka metodą zachłanną.
2. Napisz program w języku C++, który będzie realizował algorytm pakowania plecaka metodą zachłanną.
3. Napisz program w języku Java, który będzie realizował algorytm pakowania plecaka metodą zachłanną.
4. Napisz program w języku C++, który będzie realizował algorytm pakowania plecaka metodą zachłanną oraz będzie wypisywał, ile (i jakich) przedmiotów zostało umieszczonych w plecaku.

PODSUMOWANIE LEKCJI

- Ogólny problem plecakowy polega na tym, by tak spakować przedmioty, aby ich wartość była największa.
- Metoda zachłanna to taki sposób pakowania plecaka, aby włożyć do niego przedmioty o jak największej wartości.

s. 297

s. 297

51. Z miasta A do miasta B, czyli szukanie najkrótszej drogi metodą zachłanną

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest algorytm zachłanny;
- dowiesz się, czym jest problem najkrótszej drogi i jak go rozwiązać;
- poznasz algorytm Dijkstry.

Planując podróż czy trasę do konkretnego miejsca, wybieramy przeważnie najkrótszą drogę – oczywiście ma to sens, gdy poruszamy się pieszo. Kierowcy wiedzą bardzo dobrze, że najkrócej nie zawsze oznacza najszybciej. W tym rozdziale zajmiemy się wyszukiwaniem właśnie takiej najkrótszej drogi z punktu A do punktu B. Pomoże w tym algorytm wyszukiwania najkrótszej drogi.

51.1. Algorytm zachłanny

algorytm
zachłanny

Algorytm zachłanny to taki, który w celu wyznaczenia rozwiązania w każdym kroku będzie dokonywał najbardziej prawdopodobnego wyboru rozwiązania częściowego, czyli tzw. zachłannego. Algorytm zachłanny nie będzie oceniał, czy w kolejnych krokach ma sens wykonywanie działań; dokona on wyboru, który w danej chwili jest najlepszy, i będzie kontynuował dalsze rozwiązywanie problemu wynikającego z podjętej decyzji.

51.2. Problem najkrótszej ścieżki

problem
najkrótszej ścieżki

Problem najkrótszej ścieżki polega na znajdowaniu w grafie najkrótszego połączenia pomiędzy wierzchołkami.

Aby znaleźć najkrótszą ścieżkę między dwoma wierzchołkami grafu, musimy znaleźć najkrótsze ścieżki między wierzchołkiem wyjściowym a wszystkimi innymi wierzchołkami. Problem ten możemy opisać jako znalezienie najkrótszej drogi między miastem A i miastem B.

Szukając najkrótszej drogi, pozostałe wierzchołki grafu będziemy traktować jak skrzyżowania dróg – krawędzi, a wagi krawędzi będą odwzorowywać długość naszego odcinka drogowego.

Do znalezienia najkrótszej ścieżki między dwoma wierzchołkami posłużymy się algorytmem Dijkstry.



Algorytm ten został opracowany przez Holendra Edsgera Dijkstrę (czyt. edshera dejkstrę) i służy do znajdowania najkrótszej ścieżki z pojedynczego wierzchołka w grafie o nieujemnych wagach krawędzi, czyli nieujemnych długościach odcinków. Pozwala odnaleźć w grafie wszystkie ścieżki pomiędzy wybranym wierzchołkiem a pozostałymi, wyliczając przy tym wartość przejścia każdej ścieżki. Możemy go łatwo zmodyfikować tak, aby szukał wyłącznie najkrótszej ścieżki od jednego wierzchołka do innego przez nas wskazanego. Działanie tego algorytmu powinno być przerwane w momencie znalezienia najkrótszej drogi między wybranymi wierzchołkami.

Algorytm Dijkstry jest przykładem algorytmu zachłannego.

51.3. Algorytm Dijkstry

W każdej iteracji wybierany jest ten spośród nieodwiedzonych wierzchołków, do którego możemy dotrzeć najkrótszą drogą. Po wyznaczeniu ścieżki do konkretnego wierzchołka nie zostanie ona zmodyfikowana w trakcie dalszego wykonywania algorytmu.

Dla każdego wierzchołka są wyznaczone dwie wartości: droga dotarcia do wierzchołka oraz poprzedni wierzchołek na ścieżce. Na początku działania naszego algorytmu dla wierzchołka źródłowego droga dotarcia wynosi 0, ponieważ już jesteśmy w tym miejscu, natomiast dla każdego innego droga ta równa się nieskończoności – ponieważ nie wiemy, jak mamy się do nich dostać.

Działanie algorytmu:

1. Wszystkie wierzchołki na początku znajdują się w zbiorze P .
2. Dopóki zbiór P nie jest pusty, pobieramy z niego wierzchołek o najkrótszej drodze dotarcia. Oznaczamy go jako x i usuwamy ze zbioru P .
3. Dla każdej krawędzi k wychodzącej z wierzchołka x wykonaj następujące czynności:
 - oznacz wierzchołek znajdujący się na końcu krawędzi k jako v
 - jeżeli droga dotarcia do wierzchołka v z wierzchołka x poprzez krawędź k jest mniejsza od aktualnej drogi dotarcia do wierzchołka v , to:
 - przypisz drodze dotarcia do wierzchołka v drogę dotarcia do wierzchołka x powiększoną o wagę krawędzi k
 - ustaw wierzchołek x jako poprzednik wierzchołka v .



ZADANIA DO WYKONANIA

1. Napisz algorytm w postaci listy kroków, który będzie realizował algorytm Dijkstry.
2. Napisz program w języku C++, który będzie realizował algorytm Dijkstry.
3. Napisz program w języku Java, który będzie realizował algorytm Dijkstry.

PODSUMOWANIE LEKCJI

- s. 300 • Algorytm zachłanny to taki, który w celu wyznaczenia rozwiązania w każdym kroku będzie dokonywał najbardziej prawdopodobnego wyboru rozwiązania cząstkowego, czyli tzw. zachłannego.
- s. 300 • Problem najkrótszej ścieżki polega na znajdowaniu w grafie najkrótszego połączenia pomiędzy wierzchołkami.
- s. 301 • Algorytm Dijkstry pozwala odnaleźć w grafie wszystkie ścieżki pomiędzy wybranym wierzchołkiem a pozostałymi, wyliczając przy tym wartość przejścia każdej ścieżki.



52. Pakowanie plecaka, czyli programowanie dynamiczne

NA TEJ LEKCJI:

- dowiesz się, czym jest programowanie dynamiczne;
- poznasz sposób rozwiązywania problemu pakowania plecaka metodą dynamiczną.

Aby dojść do poprawnego rozwiązania, zawsze możemy spróbować wykonać zadanie kilkoma różnymi metodami. W poprzednich rozdziałach poznaliśmy metody zachłanne rozwiązywania problemów, a w tym poznamy metodę dynamicznego programowania.

52.1. Programowanie dynamiczne

Wiesz już, czym jest programowanie dynamiczne. W tym rozdziale przypomnimy ci i poszerzymy ten temat. Programowanie dynamiczne jest jedną z technik projektowania algorytmów, którą stosuje się do rozwiązywania zagadnień optymalizacyjnych, czyli wyznaczania najlepszego rozwiązania z punktu widzenia określonego kryterium.

Programowanie dynamiczne to alternatywa dla algorytmów zachłannych. Jego twórcą jest amerykański matematyk Richard Bellman.

Programowanie dynamiczne opiera się na podziale rozwiązywanego problemu na kilka części. W odróżnieniu od metody dziel i rządź, części te nie są rozłączne. Musi je cechować również własność optymalnej podstruktury – dany problem ma optymalne rozwiązanie, jeżeli jest funkcją optymalnych rozwiązań problemów.

Wyniki rozwiązania problemów w programowaniu dynamicznym zapisywane są w tabeli, dzięki czemu, jeżeli natrafimy ponownie na ten sam problem, nie będziemy musieli już go ponownie rozwiązywać.

Wykorzystując programowanie dynamiczne, możemy zastosować jedną z dwóch metod: metodę zstępującą z zapamiętywaniem lub metodę wstępującą.

Metoda zstępująca z zapamiętywaniem polega na rekurencyjnym wywoływaniu funkcji z zapamiętywaniem wyników. Metoda ta jest bardzo podobna do metody dziel i rządź, z tą różnicą, że jeżeli rozwiązanie danego problemu już mamy w tabeli wyników, to je z niej odczytujemy.

Metoda wstępująca polega na rozwiązywaniu wszystkich problemów w taki sposób, że zaczynamy od tych najmniejszych. W momencie, kiedy przejdziemy do rozwiązywania problemu całkowitego, będziemy mieli dostępne rozwiązania pośrednie. W tym podejściu nie będziemy zużywać pamięci na rekurencyjne wywoływanie funkcji. Niedoskonałością tej metody jest fakt, że możemy niektóre problemy rozwiązać nadmiarowo.

programowanie
dynamiczne

metoda zstępująca
z zapamiętywaniem

metoda
wstępująca



52.2. Programowanie dynamiczne – problem plecakowy

Rozwiązanie problemu plecakowego metodą programowania dynamicznego jest równie skuteczne jak metodą zachłanną.

Przyjmijmy, że mamy N przedmiotów, które będziemy musieli spakować do plecaka o pojemności wag równej K . Każdy z naszych przedmiotów waży w_i i ma wartość c_i .

Niech $Q(i)$ będzie największą wartością, przy założeniu wagi mniejszej bądź równej i , a $Q(K)$ niech będzie rozwiązaniem plecaka.

$Q(i)$ jest zdefiniowane rekurencyjnie:

- $Q(0) = 0$,
- $Q(i) = \max\{c_j + Q(i - w_j) : w_j \leq i\}$

Rozwiązanie dla pustego plecaka jest równe zero. Obliczenie wyników kolejno dla $Q(0)$, $Q(1)$ aż do $Q(W)$ pozwoli nam obliczyć wynik.

Pseudokod (rozwiązanie znajduje się w komórce $Q[W]$, tablica $Q[0...W]$ przechowuje wyniki, wagi znajdują się w tablicy $w[1... n]$, a wartości w tablicy $c[1... n]$):

```
for i=0 to W do
  Q[i] = 0
for i = 0 to W do
  for j = 1 to n do
    if ( w[j] <= i ) then
      Q[i] = max(Q[i], Q[i - w[j]] + c[j])
```

ZADANIA DO WYKONANIA

1. Napisz algorytm w postaci listy kroków, który będzie rozwiązywał problem plecakowy metodą programowania dynamicznego.
2. Napisz program w języku C++, który będzie rozwiązywał problem plecakowy metodą programowania dynamicznego.
3. Napisz program w języku Java, który będzie rozwiązywał problem plecakowy metodą programowania dynamicznego.

PODSUMOWANIE LEKCJI

- s. 303 • Programowanie dynamiczne opiera się na podziale rozwiązywanego problemu na kilka części. W odróżnieniu od metody dziel i rządź, części te nie są rozłączne. Musi je cechować również własność optymalnej podstruktury – dany problem ma optymalne rozwiązanie, jeżeli jest funkcją optymalnych rozwiązań problemów.
- s. 303 • Metoda zstępująca z zapamiętywaniem polega na rekurencyjnym wywoływaniu funkcji z zapamiętywaniem wyników.
- s. 303 • Metoda wstępująca polega na rozwiązywaniu wszystkich problemów w taki sposób, że zaczynamy od tych najmniejszych. W momencie, kiedy przejdziemy do rozwiązywania problemu całkowitego, będziemy mieli dostępne rozwiązania pośrednie.



53. Metoda haszowania, czyli wyszukiwanie wzorca w tekście

NA TEJ LEKCJI:

- dowiesz się, czym jest metoda haszowania;
- poznasz sposoby radzenia sobie z kolizjami, które występują przy stosowaniu metody haszowania.

Każdy z nas ma w telefonie listę znajomych umieszczoną w kontaktach czy w komunikatorach. Gdy chcemy szybko znaleźć numer telefonu, mail czy zdjęcie, które znajomy umieścił na Instagramie, wystarczy, że wpiszemy kilka pierwszych liter nazwiska czy nicka znajomego, a pokażą nam się automatycznie podpowiedzi z nazwiskami lub nickami. Zastanawialiście się, jak to jest możliwe? Prawdopodobnie do wyszukiwania została zastosowana metoda haszowania albo wyszukiwanie wzorca w tekście.

53.1. Metoda haszowania

Jeżeli przyjrzymy się klasycznym metodom wyszukiwania, odkryjemy, że polegają one zazwyczaj na porównywaniu klucza, którego szukamy, z elementami przeszukiwanej tablicy. W haszowaniu natomiast próbuje się uzyskać bezpośrednie odniesienie do konkretnych elementów tablicy za pomocą operacji arytmetycznych, które przekształcają klucz w odpowiedni adres tablicy. W związku z tym haszowanie polega na odnalezieniu takiej funkcji h , która na podstawie danej opisanej przez klucz k wskaże nam indeks i w tablicy T , pod którym dana będzie przechowywana. Opiszemy to wzorem $T(i) = h(k)$.

Niewątpliwą zaletą tej metody jest maksymalne uproszczenie w przeszukiwaniu tablicy, ponieważ znając funkcję h , automatycznie możemy określić indeks komórki tablicy, w której znajduje się dana. Wadą natomiast będzie znalezienie odpowiedniej funkcji h , która ułatwi proces wyszukiwania.

Wyobraźmy sobie, że w programie mamy przechować słowa. Jeżeli użyjemy do tego celu tablicy, to zmierzmy się z problemem, w jaki sposób zamienić poszczególne słowa na liczby będące indeksami w tej tablicy. Gdybyśmy chcieli ponumerować wszystkie słowa, które występują w słowniku języka polskiego, to otrzymalibyśmy tablicę ogromnych rozmiarów. Z pomocą przyjdzie nam haszowanie. Odwzorujemy nasze słowa za pomocą funkcji haszującej, która zamieni zbiór w niewielką tablicę: $h: S\{0, \dots, n-1\}$.

Oczywiście musimy mieć świadomość, że napotkamy na kolizje podczas tworzenia takiej funkcji h .

metoda
haszowania



53.2. Tablica haszująca

Tablica haszująca, której rozmiar n składa się z list jednokierunkowych, przechowuje obiekty (słowa). Funkcja ta dla obiektów ze zbioru S będzie generowała kody – hasze – będące liczbami naturalnymi ze zbioru $[0, n - 1]$.

Tablica haszująca udostępni nam następujące operacje:

```
void insert(x) {
    wstaw obiekt x na początek listy T[h(x)];
}
void delete(x) {
    usuń obiekt x z listy T[h(x)];
}
bool find(x) {
    if (obiekt x jest na liście T[h(x)])
        return true;
    else
        return false;
}
```

Kolizje danych nie będą powodowały utraty tych danych, ponieważ z każdym haszem związane jest nie pole tablicy, ale lista.

53.3. Wyszukiwanie wzorca w tekście

Haszowanie ma zastosowanie w wielu algorytmach tekstowych. Jednym z nich będzie z pewnością algorytm wyszukiwania wzorca w tekście, który oprzemy na haszowaniu.

Założmy, że mamy słowa złożone tylko z liter alfabetu łacińskiego, a kolejnym literom alfabetu przypiszemy liczby naturalne, poczynając od 0:

$a = 0, b = 1, \dots, z = 25$

Do oznaczania haszy słów zastosujemy następujący zapis:

$$h(a_0 a_1 \dots a_n) = (a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n) \bmod p$$

gdzie p będzie dużą pierwszą literą, a x dowolną ustaloną liczbą całkowitą większą od rozmiaru alfabetu, czyli od 25, ale mniejszą od p .

Wartość tej funkcji haszującej bardzo łatwo obliczymy dla kolejnych, coraz dłuższych sufiksów słowa, stosując zależność:

$$h(a_i a_{i+1} \dots a_n) = (a_i + x \cdot h(a_{i+1} \dots a_n)) \bmod p$$

Aby szukać wzorców w tekście, przyjmijmy następujące oznaczenia:

1. m -literowy wzorec to $W = w_1 w_2 \dots w_m$
2. n -literowy tekst to $T = t_1 t_2 \dots t_n$
3. hasz podstawa $t_i \dots t_{i+m-1}$ (długość m) oznaczmy jako h_i



Musimy obliczyć hasze dla podstów m -literowych i porównać je z haszem wzorca. Jeżeli będziemy mieli obliczony hasz h_{i+1} dla podstowa $t_{i+1} \dots t_{i+m}$, to możemy znaleźć hasz podstowa zaczynający się od pozycji t_i za pomocą poniższego wzoru:

$$h_i = (t_i + x \cdot h_{i+1} - t_{i+m} x^n) \bmod p$$

Na początku wartości potęg x modulo p powinniśmy zapisać w formie tablicy.

Gdy będziemy mieli już obliczone wszystkie hasze h_i , przechodzimy do porównania ich z haszem wzorca $h(W)$.

Jeżeli do danego i będzie zachodziła nierówność: $h_i \neq h(W)$, to z pewnością na pozycji i nie będzie występował wzorec W .

Natomiast jeżeli $h_i = h(W)$, to na pozycji i prawdopodobnie występuje wzorec W .

ZADANIA DO WYKONANIA

1. Napisz w języku C++ prostą funkcję haszującą, przypisując ciągom znaków liczby całkowite z przedziału od 0 do 1000. Zaimplementuj tablicę haszującą, która wykorzysta twoją funkcję.
2. Napisz algorytm w postaci listy kroków wyszukujący dwuwymiarowy wzorec w tekście, który będzie oparty na haszowaniu słów.
3. Opracuj program w języku C++ wyszukujący dwuwymiarowy wzorec w tekście, który będzie oparty na haszowaniu słów.
4. Opracuj program w języku Java wyszukujący dwuwymiarowy wzorec w tekście, który będzie oparty na haszowaniu słów.

PODSUMOWANIE LEKCJI

- Haszowanie polega na odnalezieniu takiej funkcji h , która na podstawie danej opisanej przez klucz k wskaże nam indeks i w tablicy T , pod którym dana będzie przechowywana. Opiszemy to wzorem $T(i) = h(k)$.

s. 305



54. Metoda Monte Carlo, czyli symulacja ruchów Browna

NA TEJ LEKCJI:

- dowiesz się, czym jest metoda Monte Carlo i jak za jej pomocą obliczyć pole figury wpisanej w kwadrat;
- poznasz ruchy Browna.

Losowość zarówno w matematyce, jak i w informatyce jest istotnym elementem w rozwiązywaniu problemów.

Metoda Monte Carlo to zagadnienie, nad którym pracował jednocześnie zespół węgierskiego matematyka Johna von Neumanna (czyt. dzona fon nojmena) oraz polski matematyk Stanisław Ulam. Swoją nazwę ta metoda zawdzięcza nazwie miasta w księstwie Monako słynącego z gier hazardowych, które cechują się losowym charakterem. Główną ideą metody jest rozwiązywanie pewnych problemów matematycznych za pomocą losowych wyborów.

54.1. Metoda Monte Carlo (czyt. monte karlo)

metoda Monte Carlo

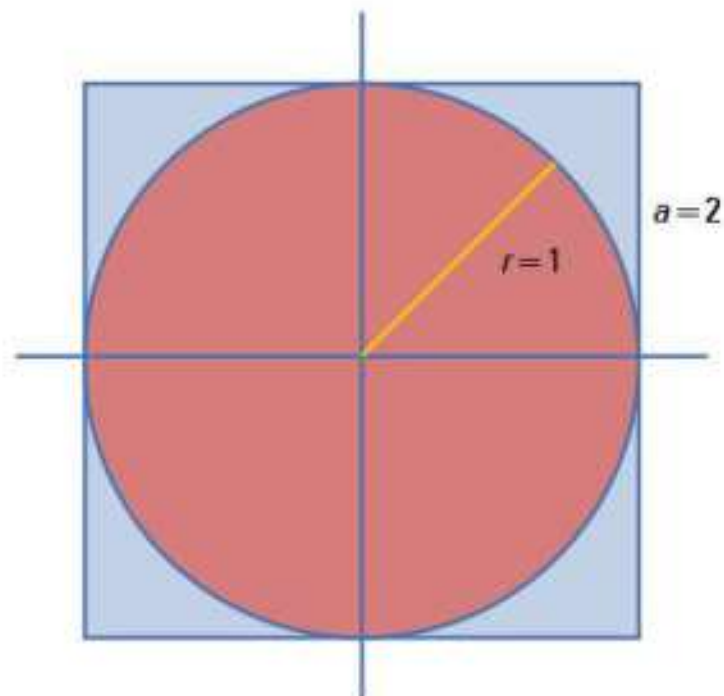
Metoda Monte Carlo jest metodą modelowania matematycznego procesów, które są zbyt złożone, aby było można przewidzieć ich wynik za pomocą podejścia czysto analitycznego, np. obliczania całek, łańcuchów procesów statystycznych itp.

Istotną rolę w tej metodzie odgrywa losowanie wielkości, które charakteryzują ten proces. Podczas losowania powinna zostać zachowana zasada, że jest ono dokonywane zgodnie z rozkładem, który musi być znany.

Aby zobrazować ogólny sposób działania metody Monte Carlo, posłużymy się przykładem tarczy, do której rzucamy rzutkami – tzw. gry w darta. Mamy więc koło wewnątrz kwadratu. Po oddaniu dość dużej liczby rzutów do tarczy i policzeniu rzutek, które trafiły w koło, możemy szacunkowo obliczyć pole jego powierzchni.

Innym przykładem wykorzystania metody Monte Carlo do obliczeń będzie próba znalezienia przybliżonej wartości liczby π . Aby tego dokonać, wykorzystamy koło wpisane w kwadrat. Musimy dokładnie znać pole powierzchni tego koła.

Przyjmijmy, że kwadrat ma długości boków $a = 2$, a koło – taką samą średnicę. Z tak przyjętych zapisów będzie również wynikało, że promień koła $r = 1$.



Rys. 54.1. Koło wpisane w kwadrat

Wzór na pole koła:

$$P = \pi \cdot r^2$$

Gdy podstawimy pod promień r liczbę 1, pole koła będzie równe liczbie π . Im więcej trafień będziemy mieli umiejscowionych w kole, tym z większą dokładnością będziemy mogli oszacować jego pole powierzchni. Obliczymy je, mnożąc pole powierzchni kwadratu przez liczbę punktów wewnątrz koła i dzieląc otrzymany wynik przez liczbę wszystkich trafień zarówno w koło, jak i w kwadrat.

```
#include <iostream>
#include <random>

using std::cout;
using std::cin;
using std::endl;

int main()
{
    std::mt19937 gen{std::random_device{}()};
    std::uniform_real_distribution<double> losuj{-1., 1.};

    int pkw = 0;
    int pko = 0;

    cout << "Podaj liczbę losowanych punktów: ";
    cin >> pkw;

    double x, y;
    for(int i{0}; i < pkw; ++i)
    {
        x = losuj(gen);
        y = losuj(gen);
        if(x*x + y*y <= 1)
```



```

        {
            ++pko;
        }
    }

    cout << "Liczba punktów w kole = " << pko << endl;
    cout << "Liczba punktów w kwadracie = " << pkw << endl;
    double _PI_ = 4. * pko/pkw;
    cout << "Szukana liczba pi = " << _PI_ << endl;
}

```

54.2. Symulacja ruchów Browna

ruchy Browna

Szkocki biolog – Robert Brown (czyt. braŋn) – w 1827 roku obserwując pod mikroskopem pyłki kwiatowe, dostrzegł, że są one w nieustannym, chaotycznym ruchu. Zjawisko to jest charakterystyczne dla wszystkich mikroskopijnych cząstek zawieszonych bez względu na ich rodzaj. Cząstki poruszają się bez ustanku, a prędkość ich wzrasta tym bardziej, im cząsteczki są mniejsze, a temperatura jest wyższa. Ruch cząsteczek nazwano ruchami Browna na cześć naukowca, który je odkrył.

Matematyczny opis ruchów Browna został podany niezależnie przez Alberta Einsteina i Mariana Smoluchowskiego. Obaj naukowcy zauważyli, że chaotyczny ruch cząsteczek jest wywołany przez nieustające bombardowanie ich przez cząsteczki wody.

Do symulowania obserwowanych w przyrodzie ruchów Browna wykorzystuje się metodę Monte Carlo.

Symulacja polega na losowym wyznaczeniu kilkuset losowych punktów o współrzędnych x i y . Aby wykreślić ruchy Browna, konieczne jest przyjęcie następujących założeń:

- cząsteczka na początku będzie znajdowała się w środku układu współrzędnych $(0, 0)$;
- cząsteczka podczas każdego ruchu będzie się przemieszczała o wektor o stałej długości q ;
- kierunek ruchu będzie wyznaczał losowy kąt φ ;
- kolejne współrzędne cząsteczki będą określone wzorami:

$$\varphi \in (0, 2\pi)$$

$$X_n = X_{n-1} + (r \cdot \cos(\varphi))$$

$$Y_n = Y_{n-1} + (r \cdot \sin(\varphi))$$

Wartości, które możemy obliczyć dzięki tej metodzie, są tablicami współrzędnych X i Y . Liczba ich elementów będzie zależna od liczby kroków symulacji.



ZADANIA DO WYKONANIA

1. Napisz algorytm (w postaci listy kroków), dzięki któremu obliczysz przybliżoną wartość pola powierzchni koła wpisanego w kwadrat, stosując metodę Monte Carlo.
2. Napisz program w języku Java, dzięki któremu obliczysz przybliżoną wartość pola powierzchni koła wpisanego w kwadrat, stosując metodę Monte Carlo.
3. Wykorzystując metodę Monte Carlo, napisz algorytm, który przeprowadzi symulację ruchów Browna.
4. Wykorzystując metodę Monte Carlo, napisz program w języku C+, który przeprowadzi symulację ruchów Browna.

PODSUMOWANIE LEKCJI

- Metoda Monte Carlo jest metodą modelowania matematycznego procesów, które są zbyt złożone, aby było można przewidzieć ich wynik za pomocą podejścia czysto analitycznego, np. obliczania całek, łańcuchów procesów statystycznych itp. s. 308
- Ruchy Browna to chaotyczne ruchy cząstek w płynie (cieczy lub gazie), wywołane zderzeniami zawiesziny z cząsteczkami płynu. Matematyczny opis ruchów Browna został podany niezależnie przez Alberta Einsteina i Mariana Smoluchowskiego. Obaj naukowcy zauważyli, że chaotyczny ruch cząstek jest wywołany przez nieustające bombardowanie ich przez cząsteczki wody. s. 310



PODSUMOWANIE PRZED SPRAWDZIANEM

30. Jak to zapisać, czyli zamiana systemów liczbowych z ósemkowego na szesnastkowy

- s. 214 • System ósemkowy jest pozycyjnym systemem liczbowym składającym się z cyfr od 0 do 7. W informatyce wykorzystywany jest przez system Linux w poleceniu `chmod`, które ustawia prawa dostępu do pliku.
- s. 215 • System szesnastkowy jest pozycyjnym systemem liczbowym, który składa się z cyfr od 0 do 9 oraz liter od A do F, i w którym A przyjmuje w systemie dziesiętnym wartość 10, a F wartość 15.
- s. 216 • Przeliczenia między dowolnymi systemami możemy dokonać w dwóch podstawowych krokach. Zamieniamy liczbę z dowolnego systemu na system dziesiętny, a następnie z systemu dziesiętnego na kolejny system.

31. Mniej lub więcej ósemek, czyli dodawanie i odejmowanie w systemie ósemkowym

- s. 219 • Dodawanie w systemie ósemkowym wykonujemy identycznie jak w systemie binarnym czy dziesiętnym. W przypadku nadmiaru dokonujemy przesunięcia liczby 1.
- s. 220 • Odejmowanie w systemie dziesiętnym jest identyczną czynnością, jak w systemach binarnym czy dziesiętnym. W przypadku niedoboru pożyczamy liczbę 1.

32. Iloczyn i iloraz oktalny, czyli mnożenie i dzielenie w systemie ósemkowym

- s. 224 • Mnożenie w systemie ósemkowym jest identyczne jak w systemie dziesiętnym, z tą różnicą, że mamy do dyspozycji tylko osiem cyfr.
- s. 225 • Dzielenie w systemie ósemkowym jest czynnością identyczną jak w systemie dziesiętnym. Podczas wykonywania dzielenia musimy pamiętać o kolejności działań i konwersji liczb między systemami.

33. Nie tylko dwójkowy i dziesiętny, czyli suma i różnica w systemie innym niż dziesiętny

- s. 230 • System pozycyjny liczb jest takim systemem zapisu liczb, w którym każda pozycja zapisu liczby ma swoją określoną i niezmienną wagę liczbową.
- s. 230 • Dodawanie w pozycyjnym systemie liczb wykonujemy identycznie jak w systemie dziesiętnym. W przypadku nadmiaru dokonujemy przesunięcia do kolumny następnej.
- s. 231 • Odejmowanie w pozycyjnym systemie liczb wykonujemy podobnie jak w systemie dziesiętnym. Musimy jednak pamiętać, że gdy cyfra odjemnej jest mniejsza niż odpowiednia cyfra odjemnika, bierzemy jednostkę z poprzedniego miejsca.

34. Iloczyn i iloraz w systemie innym niż dziesiętny, czyli wielokrotności w systemach pozycyjnych

- s. 233 • Mnożenie w liczbowych systemach pozycyjnych, tak samo jak w przypadku dodawania czy odejmowania, wykonuje się podobnie jak w systemie dziesiętnym. Musimy tylko pamiętać o tym, ile dany system zawiera liczb.



- Dzielenie, tak samo jak pozostałe działania arytmetyczne, wykonujemy w sposób bardzo zbliżony do dzielenia wykonywanego w systemie dziesiętnym. Tutaj również musimy pamiętać, w jakim systemie liczbowym tę czynność wykonujemy, i brać pod uwagę dostępny zakres cyfr w danym systemie.

s. 235

35. Wyciąganie elementów z listy, czyli znajdowanie najdłuższego spójnego podciągu niemalejącego

- Ciąg to przyporządkowanie wszystkim liczbom naturalnym z przedziału $[1, n]$ lub wszystkim dodatnim liczbom naturalnym elementów pewnego ustalonego zbioru. W pierwszym przypadku będzie to ciąg skończony, a w drugim – nieskończony.
- Podciąg powstaje wtedy, gdy wybierzemy pewną liczbę (skończoną lub nieskończoną) elementów ciągu źródłowego, zachowując ich pierwotną kolejność występowania.
- Spójność podciągu zachodzi wtedy, gdy elementy podciągu występują kolejno po sobie.
- Z podciągiem rosnącym mamy do czynienia wtedy, gdy jego kolejny element jest większy od bezpośrednio go poprzedzającego.
- Podciąg niemalejący będzie występował w ciągu, gdy jego kolejne elementy będą większe niż element poprzedzający bądź jemu równe.

s. 237

s. 237

s. 237

s. 237

s. 239

36. Elementy występujące kolejno po sobie, czyli znajdowanie najdłuższego spójnego podciągu o największej sumie

- Spójny podciąg o maksymalnej sumie to taki fragment ciągu, którego suma jest największa.
- Sumy prefiksowe to sumy wszystkich wyrazów ciągu od jego początku, np. dla ciągu 1, 2, 3 sumy prefiksowe to 1, 3, 6. Sumy prefiksowe obliczymy w czasie liniowym (iterując do tablicy), jeśli dodamy nowy element do poprzedniej sumy prefiksowej.

s. 241

s. 242

37. Ta sama kolejność, czyli szukanie najdłuższego wspólnego podciągu

- Programowanie dynamiczne jest metodą projektowania algorytmów w taki sposób, aby problemy były rozwiązywane w sposób optymalny. Stanowi ono alternatywę do rozwiązywania zagadnień za pomocą algorytmów zachłannych.
- Najdłuższy wspólny podciąg (NWP) jest to najdłuższy podciąg znaków, które występują w tej samej kolejności w dwóch porównywanych ze sobą łańcuchach. Elementy podciągów nie muszą występować obok siebie.

s. 245

s. 245



38. Z której strony, czyli położenia punktów względem prostej

- Układ współrzędnych (na płaszczyźnie dwuwymiarowej) to dwie prostopadłe do siebie osie liczbowe – oś odciętych X i oś rzędnych Y .
- Położenie punktu w układzie współrzędnych określamy za pomocą współrzędnych – $P(x, y)$. Zawsze pierwsza współrzędna to odcięta, a druga to rzędna.

s. 250

s. 252



39. Algorytmy badające własności geometryczne, czyli przynależności punktu do odcinka

- s. 253 • Odcinek jest najkrótszą linią, która łączy dwa punkty.
- s. 254 • Nachylenie jest stosunkiem różnicy poziomów odcinka i różnicy jego przesunięcia.
- s. 257 • Współliniowość punktów występuje wtedy, gdy punkty A , B i C leżą na jednej prostej.
- s. 258 • Macierzą nazywamy układ liczb lub symboli, które są ułożone w wiersze i kolumny.

40. Skrzyżowanie dróg, czyli badanie przecinania się odcinków

- s. 260 • Odcinki się przecinają, gdy mają jeden wspólny punkt i nie zawierają się w jednej prostej.
- s. 262 • O „czystym przecięciu” mówimy wtedy, gdy odcinki się przecinają i końce jednego odcinka nie należą do drugiego.
- s. 262 • O „przecięciu brzegowym” mówimy wtedy, gdy jeden z końców odcinka leży na prostej zawierającej drugi odcinek.

41. Wewnątrz czy na zewnątrz, czyli badanie przynależności punktu do trójkąta

- s. 264 • Trójkąt to najmniejsza figura wypukła i domknięta, zawierająca pewne trzy ustalone i niewspółliniowe punkty płaszczyzny (otoczka wypukła wspomnianych trzech punktów). Odcinki tworzące łamaną nazywamy bokami, a punkty wspólne sąsiednich boków – wierzchołkami trójkąta.
- s. 266 • We współrzędnych barycentrycznych położenie punktu opisywane jest w odniesieniu do współrzędnych (x, y) wierzchołków.

42. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą prostokątów

- s. 268 • Całka oznaczona, nazywana również całką Riemanna, pozwala obliczyć pole powierzchni między wykresem funkcji $f(x)$ w pewnym przedziale $[a, b]$ a osią odciętych.
- s. 268 • Całka nieoznaczona jest rozumiana jako pojęcie odwrotne do pochodnej funkcji.
- s. 269 • Całkowanie numeryczne w przybliżony sposób umożliwia obliczenie wartości pola figury, która powstała z zamknięcia obszaru między wykresem funkcji $y = f(x)$.
- s. 269 • Metoda prostokątów jest prostym i skutecznym sposobem wyznaczenia przybliżonej wartości całki oznaczonej poprzez podzielenie obliczanego pola na mniejsze obszary, zazwyczaj prostokąty.

43. Wyznaczanie pola ograniczonego wykresem funkcji, czyli całkowanie numeryczne metodą trapezów

- s. 273 • Obie metody dla małej liczby przedziałów są nieefektywne. Wówczas o tym, która jest lepsza, decyduje przypadek lub – określając to lepiej – rodzaj funkcji i szczęście w doborze punktów podziału. Przyjmuje się, że metoda trapezowa jest bardziej precyzyjna. W obu metodach przedziały muszą być tak małe, aby wartości tworzyły wykres w przybliżeniu liniowy, inaczej błąd może być duży.



44. Znajdowanie elementów w zbiorze, czyli wyszukiwanie liniowe

- Algorytmy naiwne to najprostsze algorytmy wyszukiujące, które podczas przetwarzania danych stosują intuicyjną metodę wyszukiwania w przestrzeni poszukiwań, czyli w zbiorze wszystkich możliwych rozwiązań. s. 276
- Wyszukiwanie liniowe w informatyce nazywane jest również wyszukiwaniem sekwencyjnym. Polega ono na przeglądaniu kolejnych elementów w zbiorze (tablicy), czyli na porównywaniu żadanego klucza z kolejnymi kluczami z sekwencji danych. s. 276

45. Szukanie połówek, czyli wyszukiwanie elementów poprzez połowienie

- Wyszukiwanie binarne jest jednym z bardziej wydajnych algorytmów wyszukiujących określony element w uporządkowanej liście elementów. Jego działanie polega na wielokrotnym dzieleniu na pół przeszukiwanej listy. Czynność ta będzie wykonywana tak długo, aż nie znajdzie się wyszukiwany element. s. 279

46. Ustawianie kart, czyli sortowanie przez wstawianie

- Sortowanie przez wstawianie jest to prosty algorytm sortowania polegający na wstawianiu kolejnych elementów ciągu we właściwe miejsca. Jest to algorytm realizujący metodę przyrostową. s. 282

47. Regula falsi, czyli przybliżone rozwiązywanie równań

- *Regula falsi* to algorytm rozwiązywania równań nieliniowych jednej zmiennej. s. 285
- Określoność funkcji oznacza, że dla każdego argumentu x z przedziału $[a, b]$ istnieje wartość funkcji. s. 285
- Funkcja $f(x)$ jest ciągła, czyli przebiega przez wszystkie pośrednie wartości. s. 285



48. Geometria obliczeniowa, czyli sprawdzanie przynależności punktu do wielokąta wypukłego

- Wielokątem bądź wielobokiem nazywamy zbiór punktów płaszczyzny, który jest ograniczony przez łamaną, zazwyczaj zamkniętą, wraz z tą łamaną. s. 290
- Wielokąt wypukły to taki, który jest figurą geometryczną wypukłą, czyli wtedy i tylko wtedy, gdy wszystkie jego kąty zewnętrzne są wypukłe lub wszystkie przekątne zawierają się w tym wielokącie. s. 290
- Wielokątem wklęsłym jest wielokąt, którego co najmniej jedna przekątna nie zawiera się w nim, a także co najmniej jeden kąt wewnętrzny ma miarę większą od kąta półpełnego. s. 291

49. Algorytm wyszukiujący, czyli liniowe przeszukiwanie ciągu w poszukiwaniu żadanego elementu z wykorzystaniem wartownika

- Algorytm wyszukiujący to taki, którego zadaniem jest znalezienie w zbiorze lub tablicy szukanego elementu. s. 294
- Przestrzeń poszukiwań w algorytmice to wynik działania algorytmu wyszukiującego. s. 294
- Wyszukiwanie liniowe to przeszukiwanie tablicy bądź zbioru element po elemencie. s. 294
- Zadaniem algorytmu wyszukiującego z wartownikiem jest znalezienie w tablicy szukanego elementu. Do przeszukiwanej tablicy jest dodawana szukana wartość – tzw. wartownik. Jego zadaniem jest niedopuszczenie do tego, aby algorytm wyszedł poza elementy przeszukiwanej tablicy. s. 295



50. Podejście zachłanne, czyli problem plecakowy



- s. 297 • Ogólny problem plecakowy polega na tym, by tak spakować przedmioty, aby ich wartość była największa.
- s. 297 • Metoda zachłanna to taki sposób pakowania plecaka, aby włożyć do niego przedmioty o jak największej wartości.

51. Z miasta A do miasta B, czyli szukanie najkrótszej drogi metodą zachłanną

- s. 300 • Algorytm zachłanny to taki, który w celu wyznaczenia rozwiązania w każdym kroku będzie dokonywał najbardziej prawdopodobnego wyboru rozwiązania częściowego, czyli tzw. zachłannego.
- s. 300 • Problem najkrótszej ścieżki polega na znajdowaniu w grafie najkrótszego połączenia pomiędzy wierzchołkami.
- s. 301 • Algorytm Dijkstry pozwala odnaleźć w grafie wszystkie ścieżki pomiędzy wybranym wierzchołkiem a pozostałymi, wyliczając przy tym wartość przejścia każdej ścieżki.

52. Pakowanie plecaka, czyli programowanie dynamiczne



- s. 303 • Programowanie dynamiczne opiera się na podziale rozwiązywanego problemu na kilka części. W odróżnieniu od metody dziel i rządź, części te nie są rozłączne. Musi je cechować również własność optymalnej podstruktury – dany problem ma optymalne rozwiązanie, jeżeli jest funkcją optymalnych rozwiązań problemów.
- s. 303 • Metoda zstępująca z zapamiętywaniem polega na rekurencyjnym wywoływaniu funkcji z zapamiętywaniem wyników.
- s. 303 • Metoda wstępująca polega na rozwiązywaniu wszystkich problemów w taki sposób, że zaczynamy od tych najmniejszych. W momencie, kiedy przejdziemy do rozwiązywania problemu całkowitego, będziemy mieli dostępne rozwiązania pośrednie.

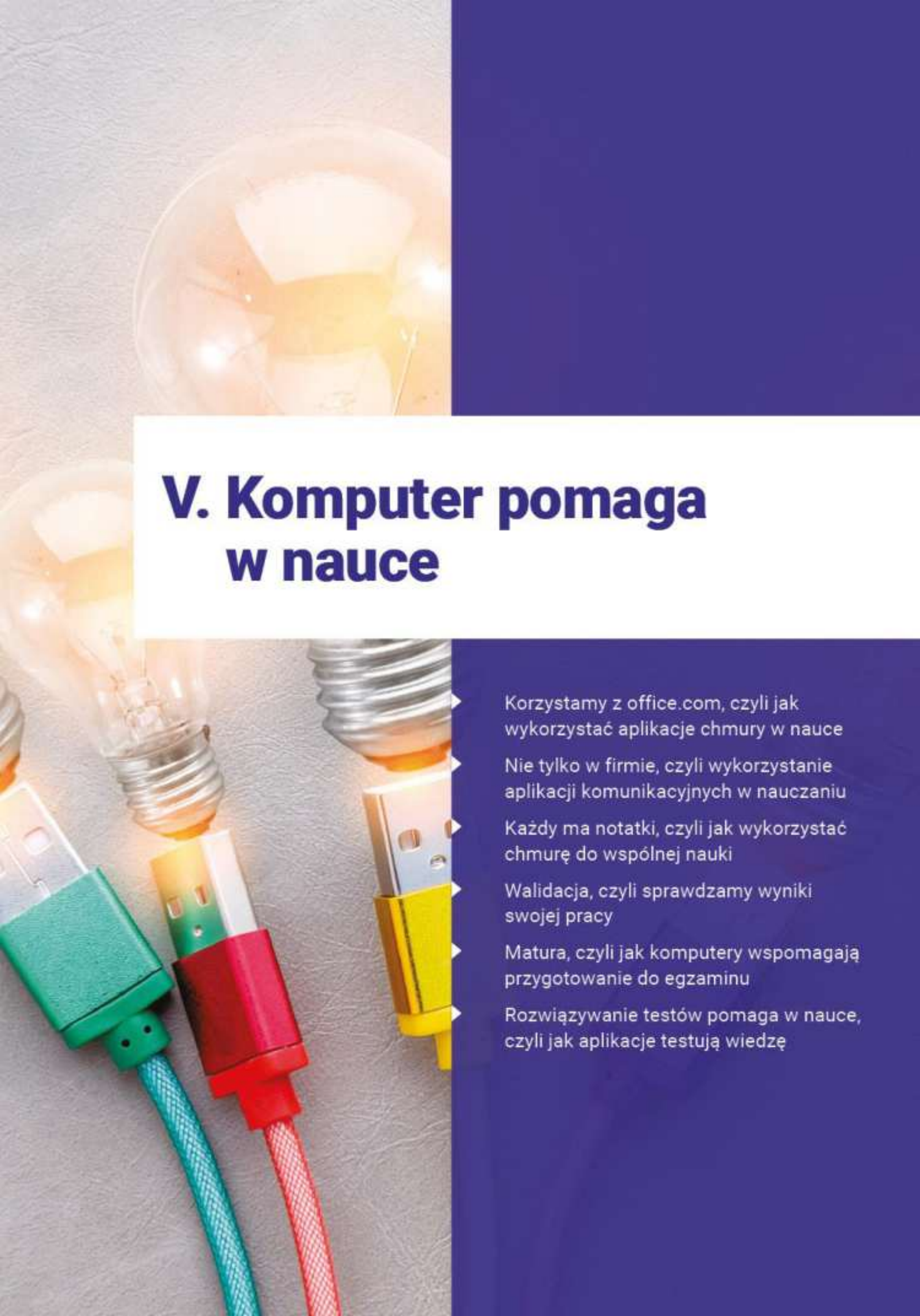
53. Metoda haszowania, czyli wyszukiwanie wzorca w tekście



- s. 305 • Haszowanie polega na odnalezieniu takiej funkcji h , która na podstawie danej opisanej przez klucz k wskaże nam indeks i w tablicy T , pod którym dana będzie przechowywana. Opiszemy to wzorem $T(i) = h(k)$.

54. Metoda Monte Carlo, czyli symulacja ruchów Browna

- s. 308 • Metoda Monte Carlo jest metodą modelowania matematycznego procesów, które są zbyt złożone, aby było można przewidzieć ich wynik za pomocą podejścia czysto analitycznego, np. obliczania całek, łańcuchów procesów statystycznych itp.
- s. 310 • Ruchy Browna to chaotyczne ruchy cząstek w płynie (cieczy lub gazie), wywołane zderzeniami zawieszonych z cząsteczkami płynu. Matematyczny opis ruchów Browna został podany niezależnie przez Alberta Einsteina i Mariana Smoluchowskiego. Obaj naukowcy zauważyli, że chaotyczny ruch cząstek jest wywołany przez nieustające bombardowanie ich przez cząsteczki wody.



V. Komputer pomaga w nauce

Korzystamy z office.com, czyli jak wykorzystać aplikacje chmury w nauce

Nie tylko w firmie, czyli wykorzystanie aplikacji komunikacyjnych w nauczaniu

Każdy ma notatki, czyli jak wykorzystać chmurę do wspólnej nauki

Walidacja, czyli sprawdzamy wyniki swojej pracy

Matura, czyli jak komputery wspomagają przygotowanie do egzaminu

Rozwiązywanie testów pomaga w nauce, czyli jak aplikacje testują wiedzę



55. Korzystamy z office.com, czyli jak wykorzystać aplikacje chmury w nauce

NA TEJ LEKCJI:

- poznasz przeznaczenie niektórych aplikacji chmury office.com;
- dowiesz się, jak je wykorzystać w organizacji czasu, nauce i opracowaniu materiałów edukacyjnych;
- przećwiczysz pracę z aplikacją chmury office.com przeznaczoną do tworzenia prezentacji internetowych.

Office.com jest powszechnie stosowaną chmurą informatyczną zawierającą znane oprogramowanie – między innymi Word, Excel i PowerPoint – w postaci aplikacji internetowych. Ale o tym wiesz i zapewne z nich korzystasz. Po zalogowaniu łatwo zauważyć, że tych aplikacji jest o wiele więcej. Niektóre z nich mogą pomóc w nauce, a także w tworzeniu materiałów elektronicznych, np. w ramach prac domowych. Takie umiejętności mogą bardzo się przydać w późniejszej pracy zawodowej, bo wiele firm także korzysta z tej chmury.

55.1. Zarządzamy czasem, czyli inteligentna chmura pomaga w planowaniu zajęć

Istnieje wiele programów do zarządzania czasem, jednak integracja takiej aplikacji z danymi i programami chmury informatycznej pozwala na wykorzystanie pełni możliwości tego typu oprogramowania. Po pierwsze, można korzystać z wcześniej zapisanych danych, np. w bazie z kontaktami, a po drugie, informować współpracowników o swoim rozkładzie dnia. Obie cechy są bardzo przydatne w pracy zawodowej, jednak skutecznie pomagają organizować czas także innym użytkownikom, np. uczniom.

To Do – aplikacja do zarządzania listą zadań

W chmurze office.com opisaną funkcję pełni aplikacja **To Do**. Dzięki niej nie tylko zaplanujesz swój dzień, lecz także nie zapomnisz o przygotowaniu się do sprawdzianu. Możesz z niej korzystać w chmurze office.com lub zainstalować ją na komputerze. Aplikacja w pełni współpracuje z jej odpowiednikiem w urządzeniach mobilnych z systemem Android i iPhone OS.

Korzystanie z To Do poznasz na przykładzie planowania zajęć określonego dnia (rys. 55.1.).



Po uruchomieniu aplikacji To Do wyświetli się menu. Dzięki dobrze dobranym nazwom opcji z łatwością wybierzesz właściwą.

Mój dzień
niedziela, 18 października

+ Dodaj zadanie

Mój dzień
niedziela, 18 października

Opracować prezentację na lekcję języka polskiego

DODAJ

Dla każdego wpisanego wydarzenia możesz ustawić dodatkowe parametry, np. **Przypomnij**. Dzięki temu aplikacja w twoim telefonie przypomni ci o nadchodzącym terminie.

To Do

Oznacz jako wykonane

Mój dzień
niedziela, 18 października

+ Dodaj zadanie

Opracować prezentację na lekcję języka polskiego

+ Dodaj krok

+ Dodano do widoku „Mój dzień”

Przypomnij

+ Dodaj termin wykonania

Powtórz

Wybierz kategorię

+ Dodaj plik

Utworzone Dnia

Przypomnij

Przypomnienie

Później dziś 15:00

Jutro pon, 9:00

Następny tydzień pon, 9:00

Wybierz datę i godzinę

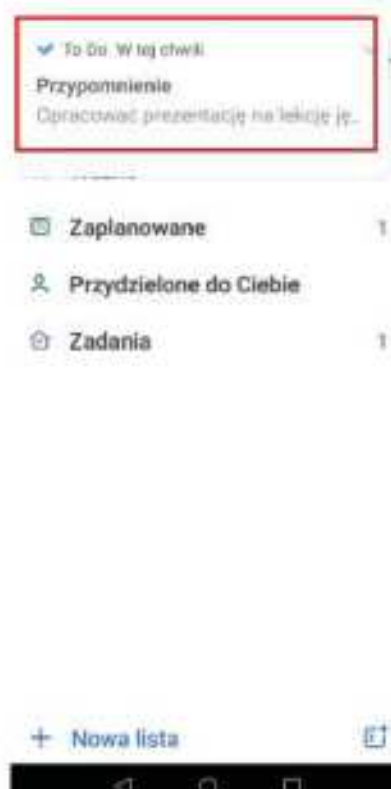
Rys. 55.1. Przykład zapisania wydarzenia w planie dnia w aplikacji To Do uruchamianej z chmury office.com



aplikacja To Do
na różne
platformy

Rys. 55.2. Aplikacja To Do zainstalowana w komputerze PC i smartfonie

opcja Przypomnienie



Jeśli nadejdzie czas przypomnienia, aplikacja w komputerze (jeśli jest aktywna) lub w smartfonie wyświetli stosowny komunikat (rys. 55.3.). Będzie tak bez względu na to, na jakim urządzeniu zostało ono ustanowione. Wszystkie zmiany w aplikacji są zsynchronizowane i w każdym urządzeniu z aplikacją (zalogowaną na to samo konto) zostają zmieniane automatycznie.

Pozostałymi opcjami aplikacji To Do możesz posługiwać się intuicyjnie. Możliwe jest także synchronizowanie (udostępnianie) wydarzeń z innymi użytkownikami aplikacji. Ponadto współpracuje ona z innymi programami, np. obsługującymi pocztę. Dzięki temu można ustawić oflagowaną informację z poczty jako zadanie.

Rys. 55.3. Przypomnienie w aplikacji smartfona

55.2. Sway, czyli prezentacje edukacyjne w sieci

Coraz częściej zachodzi potrzeba opisanie pewnego tematu, wydarzenia itp. na stronie internetowej. Wiesz już, że do tego są potrzebne hosting i znajomość sposobów tworzenia strony. Środowiska CMS są zazwyczaj programami uniwersalnymi, a więc oferują wiele opcji, wśród których tylko nieliczne będą nam potrzebne. Jak więc zbudować prezentację, która będzie funkcjonować w sieci, a jednocześnie nie będzie wymagać od nas wielu godzin pracy? Z pomocą przychodzi nam aplikacja Sway, dostępna w Windows 10 i chmurze office.com. Aplikacja zainstalowana w komputerze działa w połączeniu z kontem Microsoft. Może to być konto utworzone podczas instalacji systemu lub dowolne konto office.com. Tworzenie prezentacji jest intuicyjne.



obiekty na
licencji CC
w Office

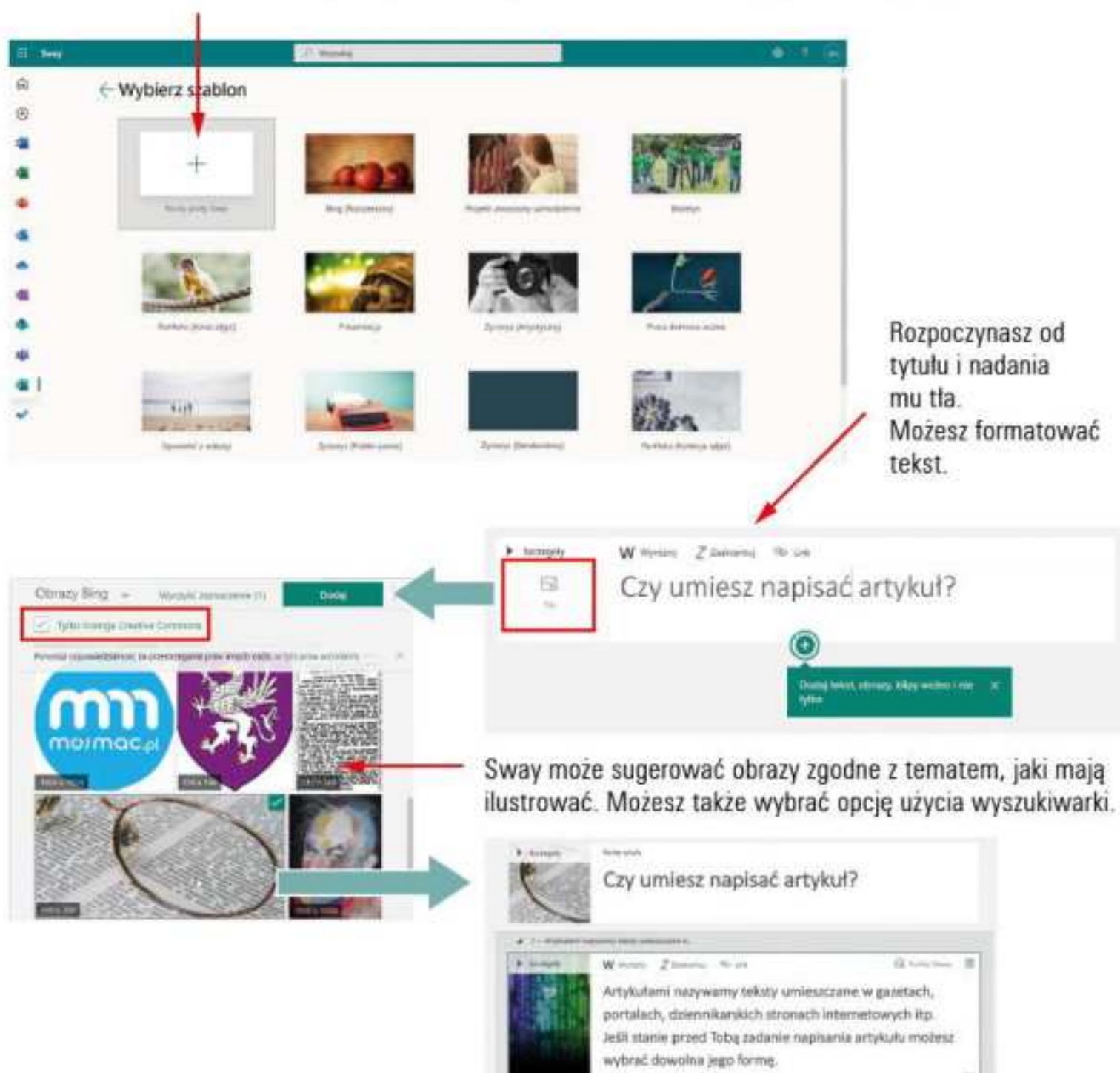
Dla ułatwienia Microsoft oferuje bogaty zestaw obiektów, np. zdjęć, na licencji CC. Takie podejście znacznie zwiększa atrakcyjność prezentacji, przyspiesza ich tworzenie i zapewnia prawa do obiektów. Dobrze przygotowane prezentacje służą upowszechnianiu wiedzy, opracowywaniu trudnych tematów, wykonywaniu zadań domowych, projektów itp. (rys. 55.4.).

integracja aplikacji
stacjonarnych
i chmur

Uwaga!

Podobnie jak w przypadku To Do zmiany i nowo tworzone prezentacje będą dostępne na każdym z urządzeń, na którym w aplikacji Sway zalogowano się na to samo konto. Są one bowiem zawsze przechowywane w chmurze.

Po uruchomieniu aplikacji możesz wybrać szablon lub tworzyć prezentację od podstaw.



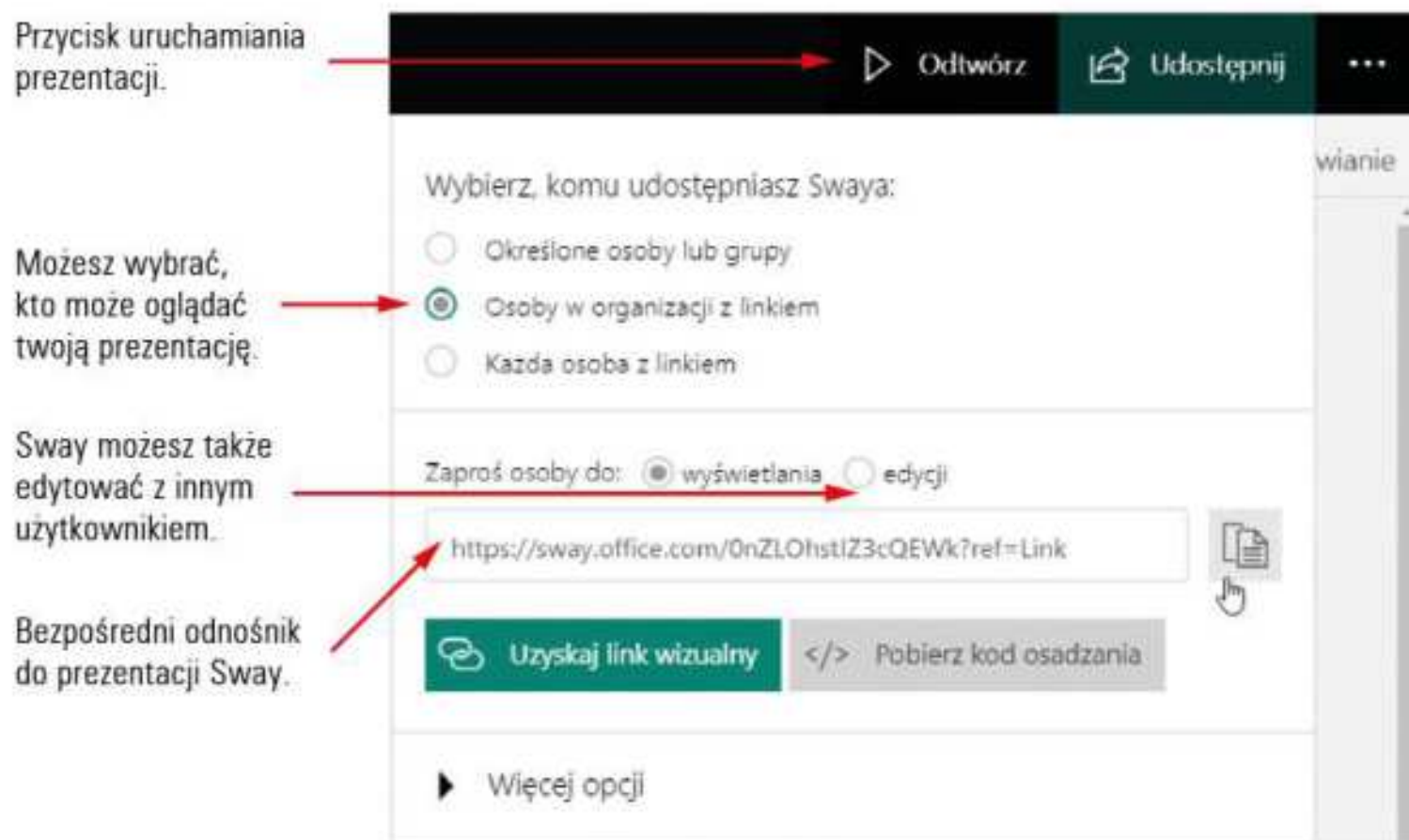
Rys. 55.4. Przykładowy proces tworzenia prezentacji Sway

**sterowanie aplikacją
Sway i jej
udostępnianie**

Sterowanie wyświetlaniem slajdów odbywa się za pomocą klawiszy ze strzałkami lub myszki. Najczęściej nie widać wyraźnych granic pomiędzy nimi. Sprawiają wrażenie jakby były ułożone na jednej płaszczyźnie.

Przesuwanie ekranu może odbywać się w pionie lub w poziomie. Możliwe jest też zaprogramowanie zbliżeń. Oczywiście slajdy mogą zawierać odnośniki i elementy multimedialne. Jak zwykle podczas pracy z edytorem prezentacji końcowy efekt zależy jedynie od wyobraźni i umiejętności autora. Warto wykonać kilka prób dla różnych wersji i wybrać najlepszą.

Udostępnianie prezentacji odbywa się za pośrednictwem odnośnika. Prezentacja może zostać także zaszyta w kodzie strony (rys. 55.5.). Odpowiedni fragment HTML można pobrać bezpośrednio ze Sway. Nie oznacza to jednak zintegrowania prezentacji ze stroną. Będzie ona pobierana z twojego konta i wyświetlana w miejscu wskazanym przez kod strony. Podobnie działa wyświetlanie reklam na stronach internetowych.



Rys. 55.5. Przykładowe okno udostępniania prezentacji Sway

Aplikacje w office.com są różnorodne i zapewne znajdziesz jeszcze inne, które mogą służyć organizacji dnia nauk, przygotowaniu do egzaminów lub do wspomagania kształcenia. Warto się im przyjrzeć i poznać ich cechy.



ZADANIA DO WYKONANIA

1. Zaloguj się do chmury office.com. Korzystając ze stron pomocy Microsoft, opisz krótko przeznaczenie wszystkich aplikacji.
2. Zainstaluj w swoim telefonie i komputerze aplikację To Do i sprawdź, czy zmiany poczynione w jednej z nich (smartfonie, PC i chmurze) zostaną odnotowane w pozostałych. Opisz zalety i wady takiego rozwiązania.
3. Wykonaj prezentację Sway na temat aplikacji To Do. Przedstaw przeznaczenie i zastosowanie aplikacji w organizacji czasu nauki ucznia. Podaj przykłady.
4. Przejrzyj udostępnione publicznie prezentacje Sway. Oceń je i opisz wybraną, według Ciebie najciekawszą. Przedstaw jej zalety i decydujące o nich cechy.

PODSUMOWANIE LEKCJI

- **To Do** jest aplikacją działającą w połączeniu z kontem Microsoft. s. 318
- Można ją instalować **w urządzeniach przenośnych i komputerach. Aplikacje synchronizują się ze sobą.** Zmiana w jednej instalacji tego samego użytkownika będzie widoczna na pozostałych. s. 320
- Opcje w aplikacji To Do pomagają w organizacji pracy lub nauki. **Przypomnienie** powoduje wyświetlanie komunikatu o czekającym nas zadaniu na wszystkich urządzeniach z aplikacją danego użytkownika. s. 320
- W Sway, podobnie jak w innych aplikacjach office.com, można używać oferowanych przez chmurę obiektów **upowszechnionych na licencji CC.** Wyszukiwane są kontekstowo, co przyspiesza ten proces. s. 321
- **Sway**, podobnie jak To Do, jest aplikacją, którą **można zainstalować stacjonarnie i w smartfonach.** s. 321
- Prezentacje Sway można udostępniać za pośrednictwem **odnośnika** i wklejać w treść strony dzięki udostępnionemu **kodowi HTML.** s. 322



56. Nie tylko w firmie, czyli wykorzystanie aplikacji komunikacyjnych w nauczaniu

NA TEJ LEKCJI:

- dowiesz się, do czego służą usługi spotkań wideo;
- zrozumiesz, czym jest e-praca i jak można ją zorganizować w firmie;
- poznasz przykład wykorzystania aplikacji komunikacyjnych do prowadzenia zdalnych zajęć w szkole i na uczelniach.

Do niedawna większość prac projektowych, biurowych itp. należało wykonywać w siedzibie firmy. Ważne było, aby pracownicy mieli dostęp do danych i materiałów oraz współpracowali z pozostałymi członkami zespołu. Wraz z rozwojem technik informatycznych stało się jasne, że taką pracę można zorganizować inaczej, wykorzystując w tym celu aplikacje i narzędzia chmur informatycznych. Nie jest już konieczne zebranie całego zespołu w jednym miejscu ani chodzenie do archiwum, by skorzystać z danych. Skoro jest to możliwe w przedsiębiorstwie, to dlaczego nie wykorzystać tego typu aplikacji w edukacji?

56.1. Firma w chmurze, czyli jak zmienia się spojrzenie na organizację pracy

Wyobraź sobie firmę informatyczną, która nie ma biur i pracowni. Czy to jest możliwe? W klasycznie zorganizowanej firmie, np. informatycznej, pracownicy i przełożeni pracują nad projektami w jej siedzibie. Tam programiści, graficy i pozostali pracownicy otrzymują zadania i rozliczają się z pracy. Odbywają także spotkania i konsultacje, tam też testują wspólnie wypracowane rozwiązania (rys. 56.1.).

Zupełnie inaczej może wyglądać organizacja pracy w firmie, w której część zadań wykonuje się w chmurze informatycznej (rys. 56.2.).



Rys. 56.1. Schemat przykładowej organizacji pracy firmy z wykorzystaniem chmury informatycznej

Przeanalizuj cechy firmy, w której większość pracowników wykorzystuje chmurę i pracuje w domu.

wykorzystanie
chmury w firmie



Rys. 56.2. Niektóre cechy przedsiębiorstwa działającego w chmurze

Nie można jednoznacznie stwierdzić, który z modeli jest lepszy. Wszystko zależy od charakteru działalności. Dobrym rozwiązaniem będzie także łączenie obu sposobów organizacji pracy. Dzięki temu można zatrudnić w „stacjonarnej” firmie specjalistów mieszkających daleko od jej siedziby.



zdalna praca
a wykluczenie

Oczywiście chmurami informatycznymi posługują się także pracownicy firmy pracujący w jej siedzibie. To jeden ze sposobów usprawnienia pracy i obiegu dokumentów. Wtedy jednak nie nazywamy tego e-pracą. **E-praca** to praca wykonywana poza siedzibą firmy, np. w miejscu zamieszkania, z wykorzystaniem sieci informatycznej i chmury z jej oprogramowaniem. Zastosowanie chmur informa-



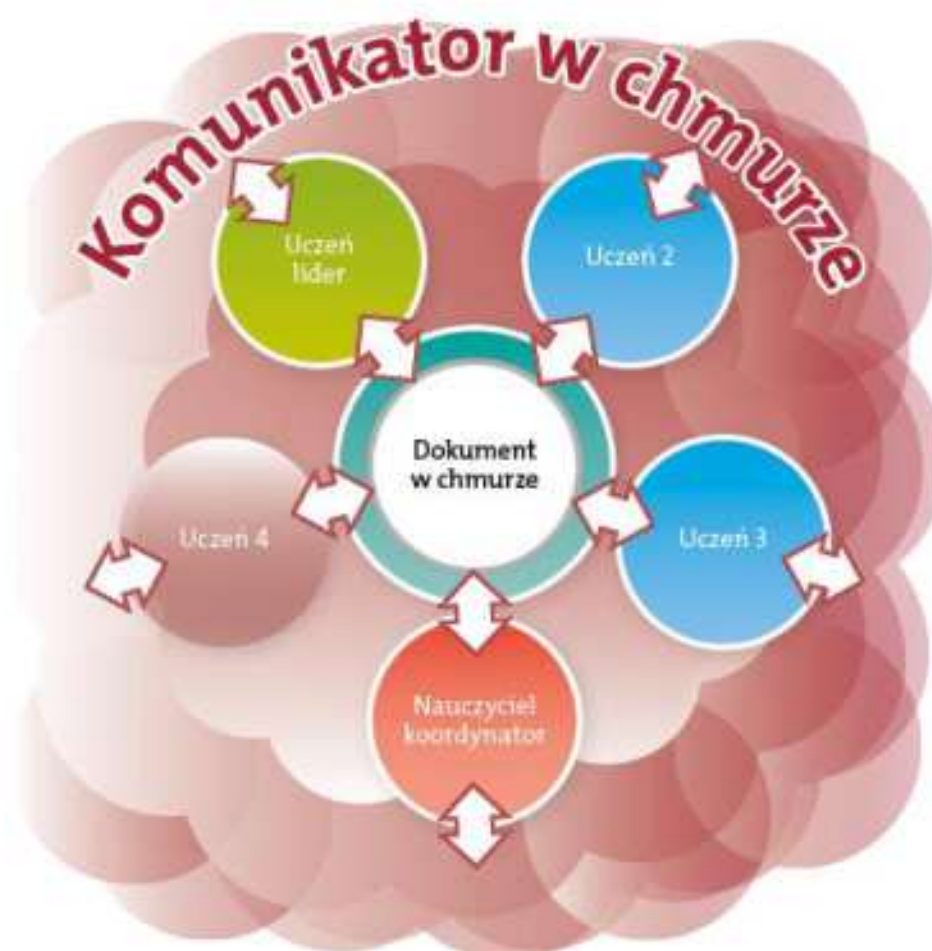
tacyjnych, w tym systemów do wideokonferencji, zapobiega także **wykluczeniu zawodowemu**, np. osób z niepełnosprawnością lub wychowujących dzieci. W obu przypadkach o wiele wygodniej jest pracować w domu (rys. 56.3.).

Rys. 56.3. Przykład aktywności zawodowej z wykorzystaniem chmury i aplikacji do wideokonferencji

56.2. Szkoła w chmurze, czyli jak zorganizować naukę zdalną

W czasie normalnego funkcjonowania szkoły aplikacje chmury mogą zostać wykorzystane między innymi do zarządzania pracą uczniów.

Załóżmy, że jesteś liderem zespołu pracującego nad projektem jednego z przedmiotów szkolnych. Wykorzystanie chmury do pracy zespołowej może mieć różne formy i tylko od twojej pomysłowości zależy, jak to zorganizujesz. Jeden z modeli takiej współpracy przedstawiono na mapie myśli (rys. 56.4.).



Rys. 56.4. Mapa myśli przedstawiająca jeden z możliwych modeli wykorzystania chmury do pracy zespołowej



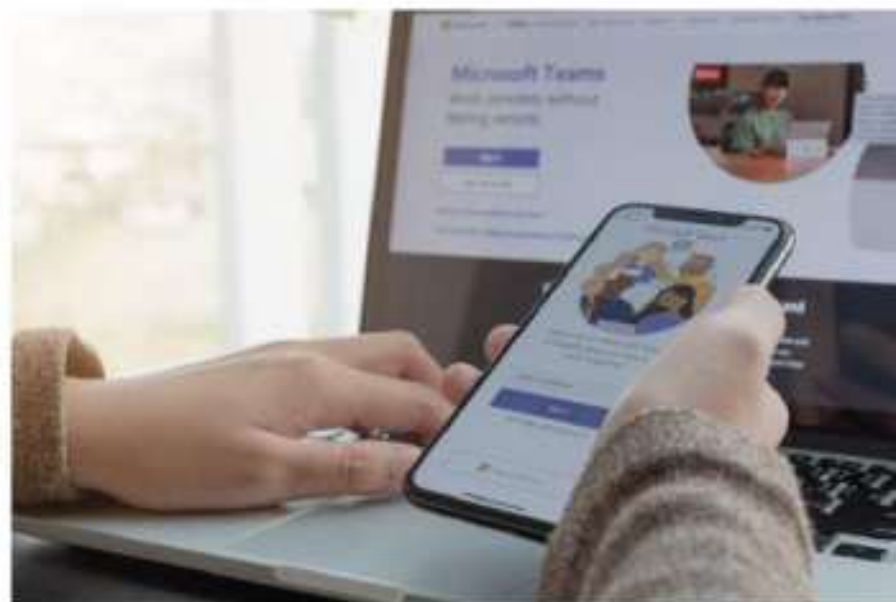
Przeanalizuj rysunek 56.4. Zastanów się, jak zorganizować strukturę chmury pasującą do takiego modelu. Zapisz czynności, jakie trzeba wykonać, by przygotować chmurę do realizacji projektu.

Przykładowe czynności przygotowujące chmurę do realizacji projektu:

1. Lider zespołu tworzy puste lub sformatowane dokumenty niezbędne do realizacji projektu. W uzasadnionych przypadkach może to zrobić nauczyciel – koordynator projektu.
2. Wszystkie dokumenty projektu zostają udostępnione odpowiednim członkom zespołu.
3. W czasie pracy nad dokumentami uczniowie konfiguruje i uruchamiają komunikator (w OneDrive będzie to Skype, ale może być także inny spoza chmury), który pozwoli na pełną współpracę w czasie edycji dokumentów i dyskusje na temat jego treści.
4. Możliwa jest kontrola i pomoc nauczyciela, który także ma dostęp do dokumentów i komunikatora.

Zdecydowałeś/zdecydowałaś już o wyborze chmury. Umiesz założyć konto i przygotować chmurę do pracy. Czas z niej skorzystać.

Co możemy zrobić w przypadku czasowego zamknięcia szkół, np. z powodu kłęski żywiołowej czy epidemii? Jak umożliwić naukę uczniowi, który przez dłuższy czas nie może przychodzić do szkoły? Można wykorzystać te same narzędzia chmury i w podobny sposób zarządzać nauką. Tu jednak zwykły komunikator i poczta mogą stwarzać wiele problemów. Wynikają one bardziej ze sposobu zarządzania niż z ułomności samych aplikacji. W wymienionych przypadkach kluczową rolę będzie odgrywał bezpośredni kontakt z uczniami lub słuchaczami studiów (rys. 56.4.). Typowe komunikatory, które służą do prowadzenia rozmów przez niewielką liczbę osób, zazwyczaj nie mają narzędzi do zarządzania spotkaniem, a jest to przecież niezbędne do uporządkowania procesu nauczania. Istnieją chmury, które udostępniają wymagane narzędzia. Przykładem może być **office365.com** i aplikacja **Teams**. To nie tylko pojedynczy program komunikacyjny. Aplikacja ta jest powiązana z innymi aplikacjami chmury i może być uruchamiana w wielu środowiskach, np. telefonach, tabletach itp. (rys. 56.5.), co jest ważną zaletą z punktu widzenia prowadzenia zdalnych zajęć.



Rys. 56.5. Przykład uruchamiania aplikacji Teams na laptopie i w telefonie komórkowym



Zalety Teams:

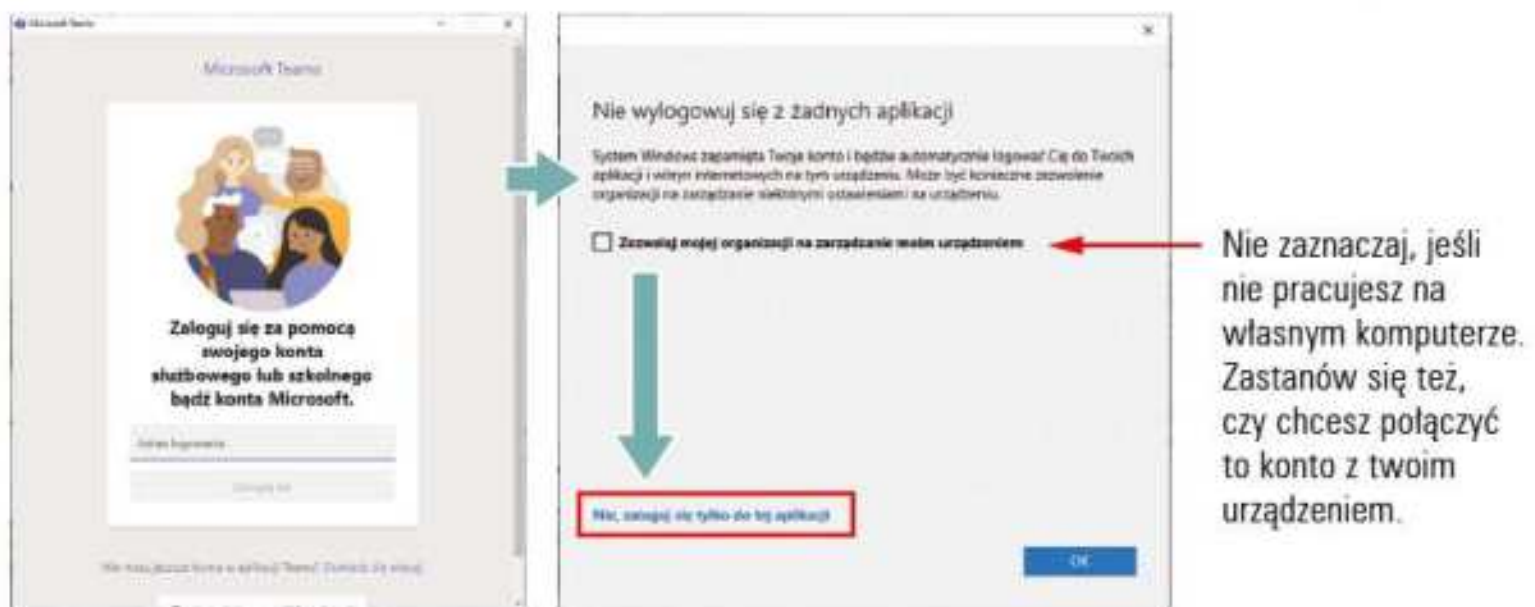
1. Integracja z chmurą office.com.
2. Aplikacja bezpłatna dla użytkowników szkolnych kont w chmurze office.com.
3. Możliwość grupowania uczestników w dowolnie zebranych zespołach, w których można tworzyć niezależnie skonfigurowane kanały dla danych przedmiotów.
4. Łatwość planowania spotkań zespołów, np. za pomocą zaznaczania godzin i dat w kalendarzu.
5. Automatyczne powiadomienia o zaplanowanych spotkaniach dla całego zespołu.
6. Możliwość nadawania i odbierania wideo jednocześnie przez wszystkich uczestników spotkania.
7. Zarządzanie uczestnikami przez organizatora (np. włączanie i wyłączanie ich kamer oraz zmienianie ich statusu).
8. Udostępnianie przez uczestników dowolnych ekranów, w tym także całego pulpitu, pozostałym osobom.
9. Możliwość udostępniania prezentacji i innych dokumentów zapisanych w chmurze office.com.
10. Możliwość uczestniczenia w spotkaniu za pośrednictwem aplikacji, w tym także w urządzeniach przenośnych, lub za pośrednictwem przeglądarki internetowej (obie formy wymagają zalogowania do konta MS Office).
11. Możliwość zaproszenia do spotkania za pośrednictwem odnośnika.

56.3. Przykład lekcji, czyli korzystamy z Teams

Microsoft Teams jest usługą internetową działająca w chmurze Office. Współpracuje z innymi aplikacjami chmury.

Załóżmy, że lekcja dotyczy informatyki i tematu „Wykorzystanie chmur do wspólnej nauki w zespole”. Nauczyciel prowadzi ją dla uczniów z jednej klasy. Dzięki cechom Teams ma również możliwość dołączenia do zespołu uczniów z innych klas, co jest bardzo ważne w przypadku zajęć w grupach łączonych, np. z przedmiotu nieobowiązkowego. Samo zorganizowanie spotkania to jeszcze nie lekcja. Należy przygotować odpowiednie materiały ilustrujące problem oraz zadania dla uczniów. W czasie klasycznej lekcji prowadzonej w sali lekcyjnej nauczyciel do ich prezentacji wykorzystuje rzutnik lub duży monitor. W trakcie lekcji zdalnej może je prezentować bezpośrednio na ekranach uczestników spotkania.

Prześledźmy przykładowy proces przygotowania takiej lekcji od logowania do aplikacji (rys. 56.6.) aż po umówienie spotkania (rys. 56.9.). W bardzo podobny sposób **przygotowuje się spotkania zespołów** w przedsiębiorstwach. Te umiejętności mogą ci się więc przydać także w przyszłej pracy zawodowej.



Rys. 56.6. Uruchamianie aplikacji Teams

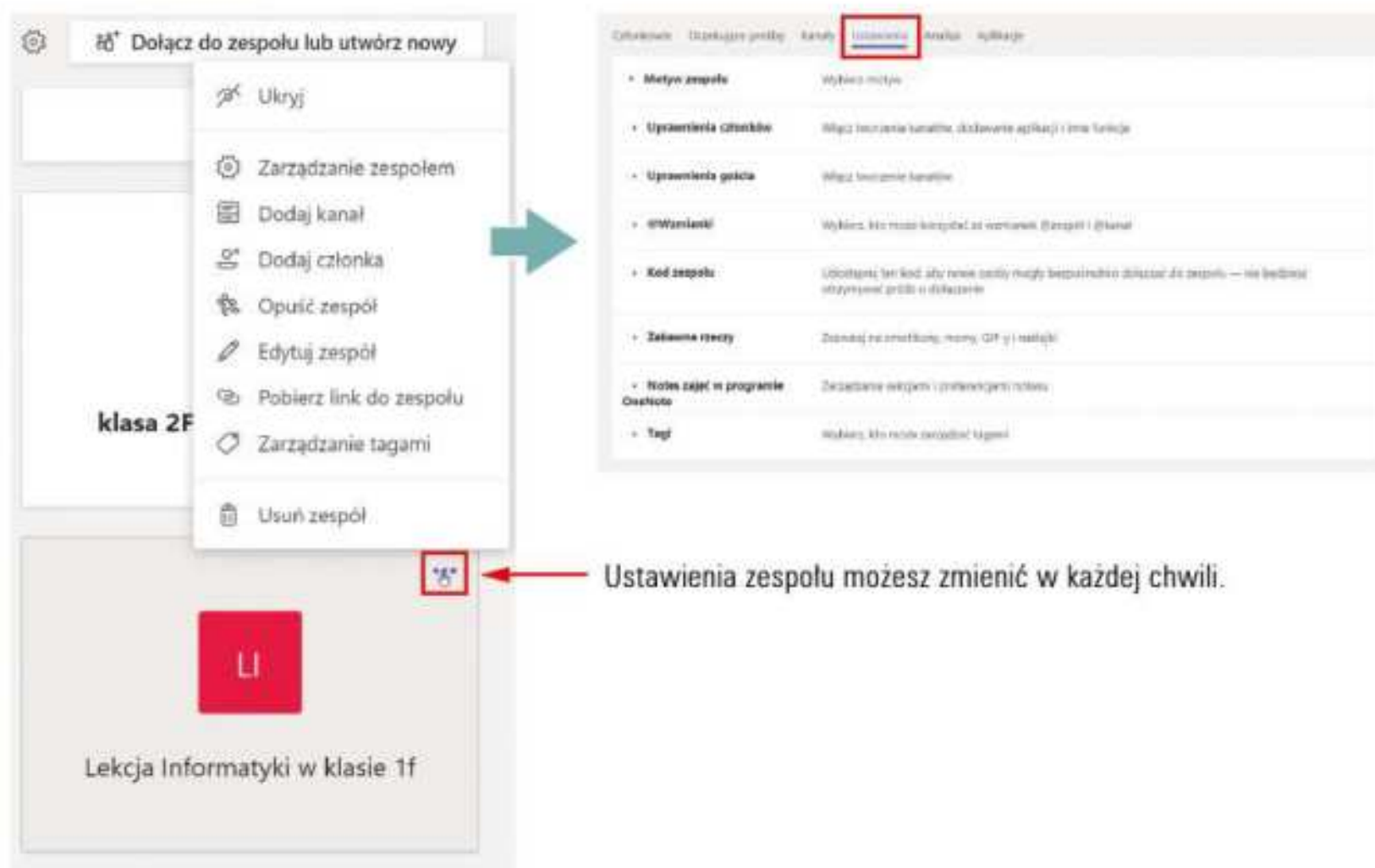


Rys. 56.7. Tworzenie zespołu biorącego udział w spotkaniach



zarządzanie zespołem

Oczywiście **zespołem można zarządzać** (rys. 56.8.).



Rys. 56.8. Opcje zarządzania zespołem w Teams

Czas na **zaproszenie** utworzonego zespołu na spotkanie (lekcję). W tym celu można użyć kalendarza (rys. 56.9.).





tworzenie spotkania

Po utworzeniu spotkania jego uczestnicy otrzymają zaproszenie. Możliwe jest także wysłanie zaproszenia z odnośnikiem za pośrednictwem poczty elektronicznej.

Prowadzący spotkanie powinien przygotować **odpowiednie materiały**. Mogą to być pliki, odnośnik lub dokumenty utworzone w chmurze Office. Może je dołączyć do danego spotkania za pośrednictwem Teams (rys. 56.10.).

Lekcja informatyki w klasie 1f — Zapraszamy!
Wybierz miejsce rozpoczęcia

Przełącz materiały z zajęć
Znajdź pomoc i szkolenia

Wybierz zespół z opcji **Zespoły** i wciśnij przycisk **Przełącz materiały** do zajęć.

Możesz tworzyć dokument, skopiować link lub przestać plik.

Powstały plik pokaże się w folderze przeznaczonym dla danego zespołu.

Rys. 56.10. Przykład przygotowania materiałów na spotkanie w zespole



Materiały na spotkania można przekazywać na kilka sposobów, np. wykorzystując OneDrive lub czat Teams.

W trakcie spotkania można także **udostępniać uczestnikom dowolne okno otwarte w systemie** lub prezentację z chmury.

Jeśli wykorzystasz znaną ci już aplikację Forms, utworzysz testy sprawdzające wiedzę.

Uwaga!

Jeśli administrator chmury Office, np. szkolnej, tak zdecyduje, to może nadać wiele uprawnień użytkownikom, między innymi prawo do tworzenia spotkań itp.

ZADANIA DO WYKONANIA

1. Opisz w dokumencie swój pomysł na przeprowadzenie w Teams lekcji na wybrany temat z innego niż informatyka przedmiotu szkolnego. Zaplanuj, jakie materiały będą przydatne w prowadzeniu zajęć i jakie treści powinny być przekazane.
2. Sprawdź w internecie, jak przebiega proces rejestracji darmowej wersji Teams przeznaczonej dla szkół i uczelni. Określ, jakie warunki należy spełnić, by proces przebiegł pomyślnie.
3. Wyszukaj w internecie alternatywne aplikacje lub chmury, które mogłyby zastąpić Teams. Opisz ich właściwości i porównaj z Teams.
- 4*. Zaprojektuj własną firmę informatyczną pod kątem miejsc pracy pracowników, umiejscowienia serwerów i siedziby. Opracuj system współpracy pomiędzy pracownikami pracującymi nad danym projektem. Przygotuj się do dyskusji na temat swojego projektu.

PODSUMOWANIE LEKCJI

- Działalność firmy można **przenieść w całości lub w części do chmury**. Wtedy pracownicy mogą pracować w domu. s. 325
- Zastosowanie chmury i pracy zdalnej pozwala także na usunięcie barier (np. długi czas dojazdu, brak windy w miejscu pracy) i **wykluczenia zawodowego** osób niepełnosprawnych. s. 326
- Spotkania w Teams można organizować dla wskazanych **użytkowników** lub członków tworzonych **zespołów**. s. 329
- **Zespołami można zarządzać** za pomocą zmian uprawnień i dodawania lub usuwania członków. s. 330
- Do utworzenia nowego spotkania może posłużyć **kalendarz**. s. 332
- Przed spotkaniami **można przygotować i udostępnić** uczestnikom materiały w postaci plików, odnośników lub dokumentów. s. 333



57. Każdy ma notatki, czyli jak wykorzystać chmurę do wspólnej nauki

NA TEJ LEKCJI:

- dowiesz się, które aplikacje chmur mogą być pomocne we wspólnej nauce;
- poznasz przykład organizacji pracy zespołu przygotowującego się do egzaminu.

Wiele projektów i prac domowych wykonujesz w zespole. Aby takie przedsięwzięcia się udały, zespół musi być dobrze zorganizowany i wyposażony w odpowiednie narzędzia informatyczne. Dlatego nie obędzie się bez chmury informatycznej. Jej aplikacje i możliwość gromadzenia zasobów pozwolą znacznie ograniczyć problemy natury logistycznej.

57.1. Wspólna nauka do egzaminu, czyli o organizacji pracy zespołu

Trudno mówić o współczesnej informatyce i zastosowaniu jej narzędzi w pracy bez zagadnień związanych z chmurami informatycznymi. Spotykasz się z nimi od bardzo dawna, zwłaszcza jeśli korzystasz np. z telefonu z systemem Android lub iOS, które są zintegrowane z chmurą. Twoje urządzenie może do niej wysyłać swoje zasoby, między innymi zdjęcia, a także przechowywać rozmaite dane i informacje. Zakładając darmowe skrzynki pocztowe, jak w gmail.com, zyskujesz dostęp do sporej pojemności dysku, aplikacji biurowych, usług streamin-
gowych (np. YouTube) i wielu innych. Używając szkolnego, komercyjnego lub darmowego konta office.com, masz dostęp do wielu aplikacji, w tym popularnych edytorów Word i PowerPoint. Nie jest dla ciebie problemem korzystanie z tych narzędzi i usług. Należy jednak zadać sobie pytanie, jak najlepiej je wykorzystać w nauce, codziennej pracy i rozrywce.

Rozpatrzmy przypadek, w którym grupa uczniów chce zoptymalizować czas i wysiłek podczas przygotowań do egzaminu lub pracy klasowej. W tym celu postanawia wykorzystać chmurę informatyczną.

Uwaga!

Jeśli nie pamiętasz, jakimi cechami charakteryzuje się chmura informatyczna oraz jakie narzędzia i zasoby w niej znajdziesz, wróć do rozdziału 1 podręcznika dla klasy 2 lub znajdź stosowne informacje w sieci.



Do dyspozycji mamy chmurę o **określonych cechach** (rys. 57.1.).

cechy chmury
przydatne w nauce



Rys. 57.1. Cechy chmury przydatne do organizacji nauki w zespole

Określmy zakres wykorzystania cech chmury w naszym przykładzie (tab. 57.1.).

przykładowy
zakres
wykorzystania
chmury w nauce

Lp.	Cecha	Przykład wykorzystania
1.	Różne uprawnienia dla członków zespołu	Pozwoli na ustalenie hierarchii użytkowników, np. umożliwi liderowi zespołu rozdzielanie zadań. Dzięki temu obszerny zakres materiału do nauki można podzielić tak, by każdy zespół mógł zająć się konkretnymi zagadnieniami i później przedstawić je pozostałym grupom.
2.	Wspólna platforma programowa	Pozwoli uniknąć niezgodności formatów plików tworzonych przez zespół. Każdy korzysta z tych samych wersji programów i edytorów.
3.	Integracja wielu usług	Wszystkie programy i pliki znajdują się w jednym miejscu, współdzielonym przez członków zespołu. Każda osoba korzysta z tych samych, zintegrowanych ze sobą aplikacji komunikacyjnych, pocztowych, organizatorów, kalendarzy itp.
4.	Programy uruchamiane w chmurze	Wszystkie programy są uruchamiane w przeglądarce lub za pomocą specjalnej aplikacji. Dzięki temu każdy może pracować na komputerach z różnymi systemami operacyjnymi, np. Linux, Windows, macOS, a także urządzeniami mobilnymi.



Lp.	Cecha	Przykład wykorzystania
5.	Współdzielenie dokumentów	Kilka osób może jednocześnie edytować ten sam dokument, np. tekstowy. Każdy widzi zmiany wprowadzane przez współautorów. Dzięki temu łatwiej można opracowywać dokumenty i samouczki. Wzmocnieniem tej cechy jest jednocześnie z edycją korzystanie z komunikatorów chmury.
6.	Komunikatory wideo	Wspomagają wspólne redagowanie dokumentów, a także pozwalają na odbywanie spotkań i narad oraz dyskusji i referowania tematu.
7.	Wspólna przestrzeń dyskowa	Może się ona składać z dysku przypisanego do jednego konta lub powstać w wyniku udostępniania folderów i plików poszczególnych członków zespołu. Dzięki temu wszyscy będą mieli dostęp do aktualnie gromadzonych materiałów pomocnych w nauce.

Tab. 57.1. Przykładowy zakres wykorzystania cech chmury informatycznej w pracy zespołu uczącego się do egzaminu

Istnieje wiele możliwych rozwiązań logistycznych organizacji pracy naszego zespołu. Przykładem może być podział zespołu na grupy, z których każda zajmie się dokładnym zebraniem informacji i przygotowaniem materiałów z innego zakresu danego tematu. Wszystkie jednak będą wykorzystywać wspólną przestrzeń dyskową, komunikatory i edytory (rys. 57.2.). Po opracowaniu wszystkich zagadnień osoby odpowiedzialne za dany dział zespołu przedstawiają je pozostałym członkom. Mogą się przy tym opierać na notatkach i materiałach zgromadzonych w chmurze. Komunikatory ułatwią także dyskusję oraz zadawanie pytań i udzielanie odpowiedzi. Zgromadzone na dysku zasoby mogą służyć indywidualnej nauce i powtarzaniu materiału w końcowej fazie przygotowań do egzaminu.

**przykład
organizacji pracy
w chmurze**



Rys. 57.2. Przykład organizacji pracy zespołu wykorzystującego chmurę do przygotowania się do egzaminu



miejsce
korzystania
z chmur

Dzięki właściwościom chmury **nie ma znaczenia, z jakiego miejsca i przy użyciu jakiego sprzętu się z nią łączymy**. Przecież chmura wymaga jedynie sprawnej przeglądarki i połączenia internetowego. Daje to możliwość pracy na komputerze w świetlicy czy w bibliotece tak samo skutecznie jak na swoim sprzęcie. Jeśli skorzystacie z urządzeń mobilnych, możecie uczyć się w każdym miejscu, np. w parku, na łące, w tramwaju, autobusie itp.

57.2. Koleżeńska pomoc, czyli wykorzystujemy chmurę do wymiany

pomoc
koleżeńska

Wyobraźcie sobie sytuację, że wasza koleżanka lub wasz kolega z pewnych przyczyn nie może uczestniczyć w zajęciach. Nauczyciele starają się pomóc, np. organizując transmisję z niektórych lekcji. Nie zawsze jednak pozwalają na to warunki. Czy można do pomocy wykorzystać chmurę? Oczywiście. Zakres zależy od pomysłowości.

Przykład 1.

Skanowanie notatek z zeszytu i udostępnienie ich w chmurze pomoże nie tylko nieobecnej koleżance/nieobecnemu koledze, lecz także całej klasie. Z takich zasobów powstanie archiwum notatek. Jeśli notatki z tej samej lekcji wyśle kilka osób, to informacja na dany temat będzie pełniejsza. Aby uniknąć chaosu i ułatwić dostęp do właściwych plików, należy zadbać o odpowiednią strukturę folderów na dysku chmury.

Przykład 2.

Wykorzystanie komunikatorów umożliwi rozmowy z uczniem nieobecnym na lekcjach i wyjaśnienie niezrozumiałych zagadnień z notatek, np. sposobu rozwiązania zadania z matematyki, fizyki itp.

Przykład 3.

Jeśli z chmury będzie korzystał również nauczyciel, może on między innymi weryfikować notatki lub osobiście objaśniać trudne zagadnienia.

Wykorzystanie chmury w opisanych przypadkach pomoże wam także w przyszłej pracy zawodowej, organizacji pracy zespołu w firmie, nauce na studiach, pracy naukowej itp.



ZADANIA DO WYKONANIA

1. Porównaj znane ci chmury, np. Google i Office.com, pod kątem wykorzystania w projekcie opisanym w rozdziale 57.1. Wymień cechy obu chmur, które są podobne lub takie same. Określ przydatność tych chmur do opisanego projektu.
2. Narysuj schemat (mapę myśli) wykorzystania chmury do zespołowego wykonania projektu – strony internetowej o patronie szkoły. Wzoruj się na rysunku 57.2.
3. Sprawdź, jakimi komunikatorami dysponują znane ci chmury. Wymień ich zalety i wady z punktu widzenia projektu z rozdziału 57.1. Określ, który z nich będzie najlepiej się sprawdzać w tym projekcie. Uzasadnij swoją tezę.
- 4*. W jaki sposób, nieopisany w rozdziale, chmury mogą pomagać w nauce i przygotowaniu się do sprawdzianów i egzaminów? Opisz jeden z takich sposobów.

PODSUMOWANIE LEKCJI

- s. 335 • Chmury mają cechy, które można wykorzystać w organizowaniu pracy zespołu, np. przygotowującego się do egzaminu.
- s. 335 • Zakres **wykorzystania tych cech** zależy od użytkowników. Dobrze zaplanowana praca z chmurą da dobre wyniki.
- s. 337 • Dzięki właściwościom chmury korzystanie z oprogramowania i zasobów **nie zależy od rodzaju komputera osobistego lub smartfona**.
- s. 337 • Za pośrednictwem chmury można także zorganizować **zdalne nauczanie lub pomoc koleżeńską** osobom nieobecnym na lekcjach.



58. Walidacja, czyli sprawdzamy wyniki swojej pracy

NA TEJ LEKCJI:

- dowiesz się, czym jest walidacja i jak pomaga w nauce;
- poznasz sieciowe usługi walidacyjne.

Twoja strona internetowa działa. Czy to oznacza, że jest bezpieczna i pozbawiona błędów? Okazuje się, że nawet takie strony internetowe mogą zawierać poważne błędy i niedoskonałości. Jak sprawdzić poprawność kodu programu lub strony? Należy go poddać procesowi walidacji. Ma to nie tylko praktyczny wymiar, lecz także duże znaczenie edukacyjne dla początkujących programistów i webmasterów.

58.1. Testuj, czyli pierwsza faza sprawdzania poprawności

Program komputerowy, w tym także kod strony internetowej, powinno się testować już w fazie jego powstawania. Każda nowa funkcja czy nowy moduł, zanim zostanie dodany do programu, powinien być przetestowany przede wszystkim przez ich autora. Dokładne sprawdzenie polega na testowaniu z użyciem wszystkich możliwych danych wejściowych, zaistniałych działań użytkownika itp. W trakcie nauki programowania pozwala to na uniknięcie sytuacji, w której rozbudowany program lub strona działa nieprawidłowo, a znalezienie błędu wymaga sprawdzenia wielu linii kodu. Twórcy oprogramowania wiedzą, że im wcześniej wykryje się błąd, tym niższe są koszty jego odnalezienia i usuwania.

Zasadę tę niektórzy nazywają **1-10-100**. Jeśli błąd wykryje programista, wystarczy, że dokona poprawy. Jeśli nie sprawdzi dokładnie poprawności działania i program zostanie przesłany do testowania, wykrycie błędu spowoduje przesłanie raportu i programu z powrotem do programisty. Ten musi znaleźć błąd, poprawić go i ponownie wysłać do testowania. Wszystko to zwiększa znacznie czas pracy nad programem i podwyższa koszty jego wytworzenia. Oczywiście błędy nie muszą być związane z samym kodem. Zdarza się, że na etapie projektowania algorytmów lub budowy strony internetowej nie uwzględnia się któregoś z założeń uzgodnionych z zamawiającym program. Mówimy wtedy, że program lub strona nie spełnia zapisów specyfikacji. Błędy mogą także zgłaszać użytkownicy (rys. 58.1.).

zasada 1-10-100



Rys. 58.1. Poprawa błędów wykryta na różnych poziomach testowania spoczywa na programiście



58.2. Nie jesteś sam, czyli walidacja z internetu

Mimo starań programisty może się tak zdarzyć, że nie zostaną wykryte wszystkie błędy. Na szczęście istnieją algorytmy, które mogą testować kod. Dotyczy to głównie kodu stron internetowych. Oczywiście nie wykryją one błędów związanych ze specyfikacją programu.

walidacja

Walidacja, czyli inaczej kontrola jakości kodu, w przypadku stron internetowych opiera się na standardach ustanowionych przez organizację **World Wide Web Consortium** (<https://www.w3.org/>).

konsorcjum
World Wide Web

„**Konsorcjum World Wide Web (W3C)** to międzynarodowa społeczność, w której organizacje członkowskie, personel zatrudniony w pełnym wymiarze czasu pracy i opinia publiczna wspólnie pracują nad opracowaniem standardów internetowych. Misją W3C, której przewodzą wynalazca i dyrektor sieci Tim Berners-Lee (wym. berners li) oraz dyrektor generalny Jeffrey Jaffe (czyt. dżefri dżafi), jest doprowadzenie sieci do pełnego wykorzystania jej potencjału” (<https://www.w3.org/Consortium/>).

walidacja w sieci

W3C nie tylko ustala standardy, lecz także daje **narzędzie do weryfikacji kodu strony**. Można je znaleźć na stronie <https://validator.w3.org/> (rys. 58.2.).

W3C oferuje także inne narzędzia i materiały edukacyjne dla webmasterów.

Możesz sprawdzać poprawność strony umieszczonej pod wskazanym adresem, między innymi już działającej w sieci.

Twój kod możesz przesłać także bezpośrednio do systemu walidacji.

Rys. 58.2. Narzędzia dla webmasterów na stronie konsorcjum W3C

Jeśli w kodzie wystąpią nieprawidłowości lub błędy, dostaniesz ich pełną listę. Błędy będą oznaczone jako **Error**, a ostrzeżenia – jako **Warning** (rys. 58.3.).



Rys. 58.3. Przykładowe komunikaty usługi walidacyjnej ze strony konsorcjum W3C

Jeśli potrzebujesz pomocy w weryfikacji kodu, tworzeniu programów i stron internetowych, wykorzystuj tylko wiarygodne źródła informacji i sprawdzone narzędzia. Zdobyte tak informacje będą wiarygodne.

ZADANIA DO WYKONANIA

1. Dokonaj walidacji kodu własnej strony internetowej, np. utworzonej na zajęciach z informatyki. Zinterpretuj wskazane błędy.
2. Wykorzystaj swoją znajomość języka angielskiego lub użyj tłumacza stron i sprawdź, czym jest Web Platform Tests ze strony <https://www.w3.org/>. Opisz krótko, na czym polega kompatybilność kodu strony z daną przeglądarką.
3. Użyj odnośnika W3Cx (Training) i sprawdź, w jakich bezpłatnych szkoleniach możesz wziąć udział. Wymień kilka z nich i napisz, czego dotyczą.
4. Wybierz jedną ze swoich ulubionych stron i za pośrednictwem narzędzia do walidacji (W3C) sprawdź poprawność jej kodu. Na podstawie raportu przekonasz się, że nawet zawodowi programiści popełniają błędy.

PODSUMOWANIE LEKCJI

- Nieoficjalnie obowiązująca zasada **1-10-100** mówi o tym, jak rosną koszty wyprodukowania programu lub strony w stosunku do etapu jego powstawania. s. 339
- **Walidacja** to kontrola jakości kodu oparta na standardach. W przypadku stron internetowych są to standardy ustanowione przez organizację **World Wide Web Consortium** (<https://www.w3.org/>). s. 340
- **W3C udostępnia narzędzie do walidacji** kodu strony internetowej zarówno umieszczonego w sieci, jak i zapisanego w pliku. s. 340
- Narzędzie do walidacji oznacza miejsca z **błędami (Error)** i miejsca, które **wymagają uwagi (Warning)**. s. 340



59. Matura, czyli jak komputery wspomagają przygotowanie do egzaminu

NA TEJ LEKCJI:

- poznasz przykłady aplikacji pomagających w przygotowaniu do matury;
- dowiesz się, jak wykorzystać aplikację GeoGebra w procesie przygotowania do matury z matematyki.

Wiesz już, jak wykorzystać aplikacje z chmury office.com do pracy w zespole i wspólnej nauki. Przygotowanie do matury i innych egzaminów wymaga także samodzielnej pracy. Nic nie stoi jednak na przeszkodzie, by oprócz notatek i podręczników wykorzystać także aplikacje komputerowe. Są one doskonałym uzupełnieniem tradycyjnych źródeł wiedzy i często pozwalają na eksperymentowanie, np. z prawami matematyki. Przykładem może być geogebra.org. To darmowa aplikacja, w której można przeprowadzać interaktywne eksperymenty, między innymi obserwować wpływ parametrów funkcji na kształt i położenie jej wykresu. W sieci znajdziesz jeszcze wiele mniej lub bardziej zaawansowanych aplikacji. Wystarczy poszukać.

59.1. GeoGebra, czyli silne narzędzie do symulacji matematycznych

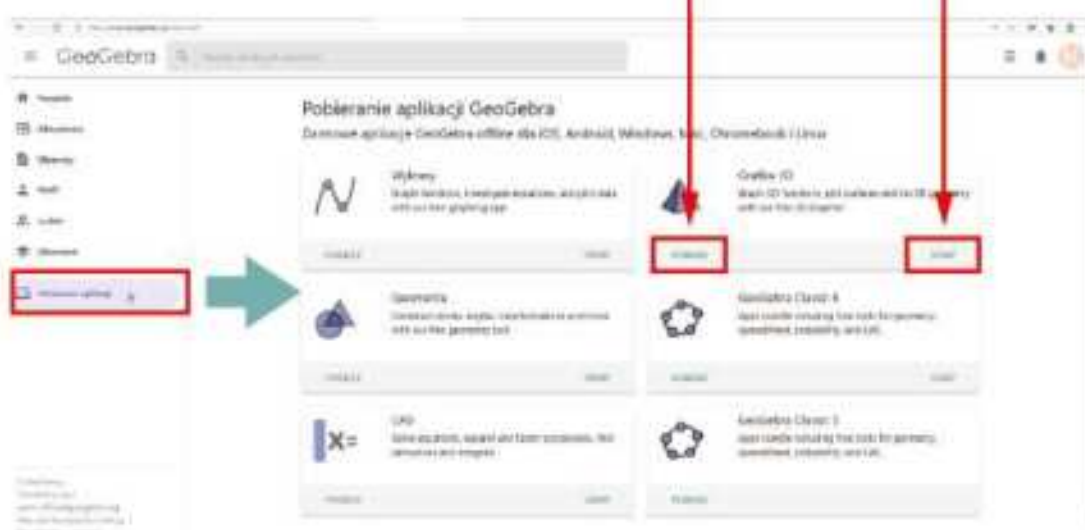
GeoGebra

GeoGebra jest chmurą informatyczną zawierającą kilka aplikacji matematycznych (rys. 59.1.). Każdą z nich można zainstalować lokalnie na komputerze osobistym.

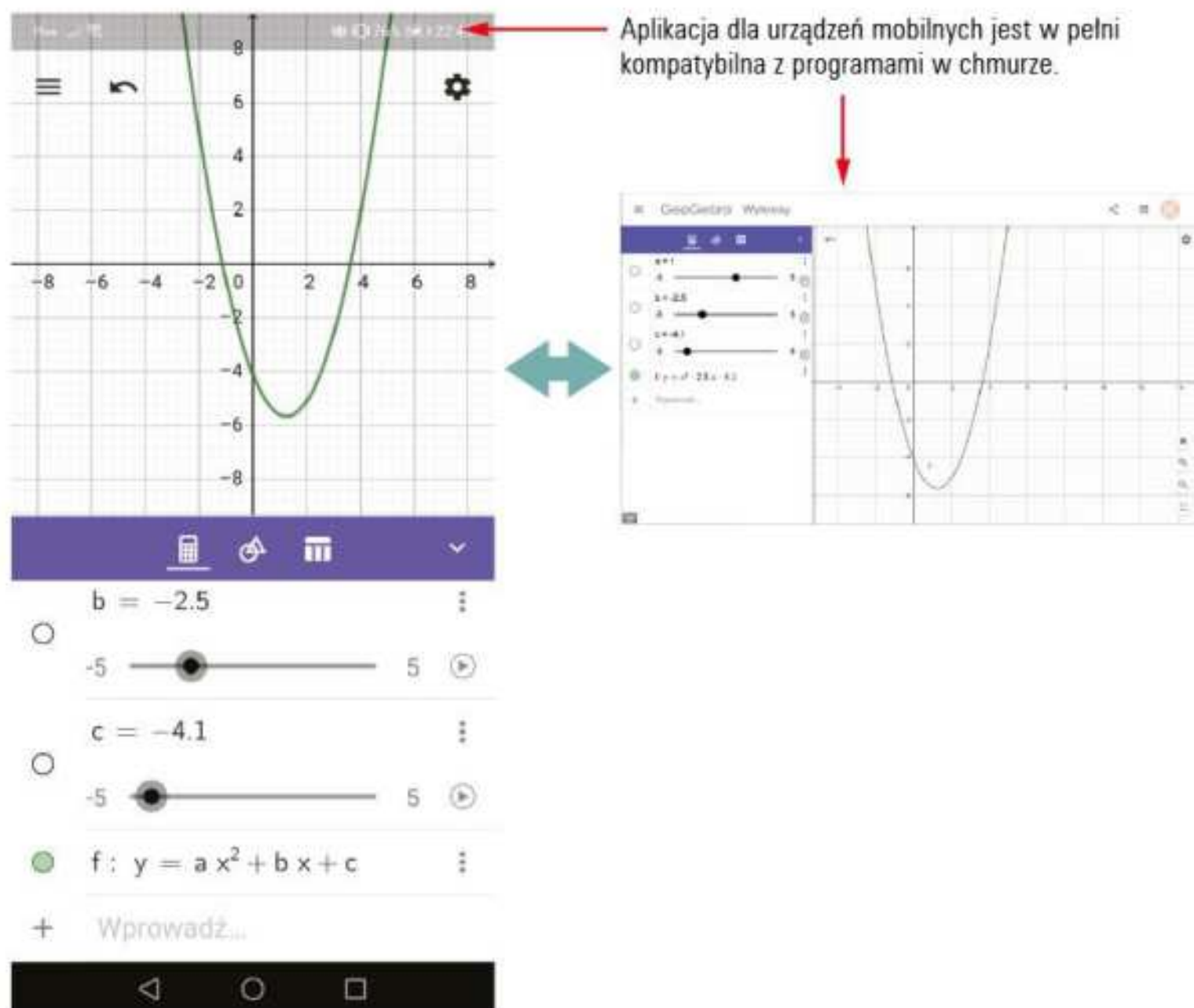
Oczywiście istnieje także aplikacja dla urządzeń mobilnych (rys. 59.2.). Jeśli założysz konto, twoje projekty i doświadczenia zapiszą się w chmurze i staną się dostępne za każdym razem, gdy zalogujesz się do GeoGebry.

GeoGebra dla urządzeń mobilnych

Po wybraniu opcji **Pobieranie aplikacji** możesz pobrać wybrany program lub uruchomić go bezpośrednio w oknie chmury.



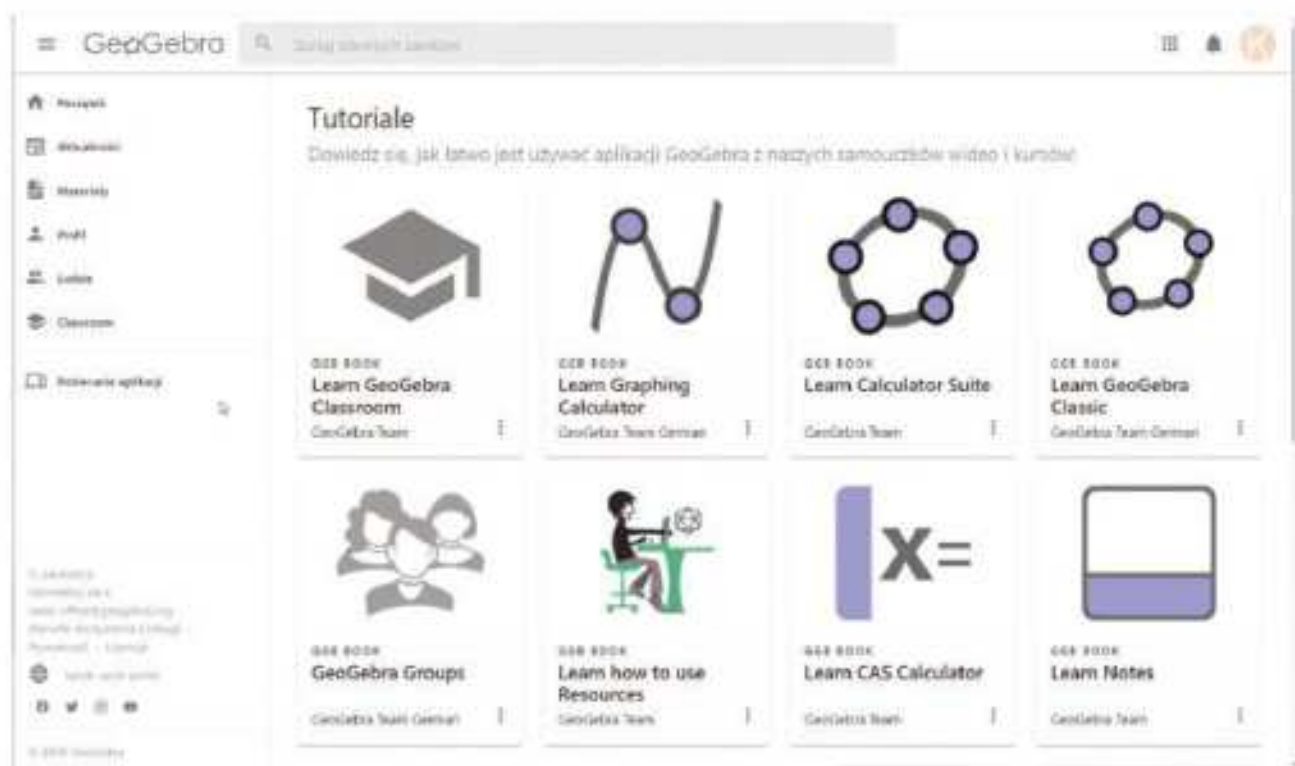
Rys. 59.1. Aplikacje geogebra.org gotowe do pobrania lub uruchomienia w chmurze



Rys. 59.2. Przykładowy wykres funkcji na ekranie aplikacji smartfona i w chmurze

Ze wszystkimi opcjami [geogebra.org](https://www.geogebra.org) możesz się zapoznać, korzystając z **samo-uczków** dostępnych w chmurze. Niektóre z nich są interaktywne i pozwalają na testowanie opisywanych czynności i weryfikację poprawności wykonania ćwiczenia (rys. 59.3.). Działy zawierają podział na sekcje podstawowe i zaawansowane. Są napisane w języku angielskim. Jeśli nie radzisz sobie z tłumaczeniem zwrotów matematycznych, użyj automatycznego tłumacza z sieci. Oczywiście z łatwością znajdziesz także tutoriale w internecie.

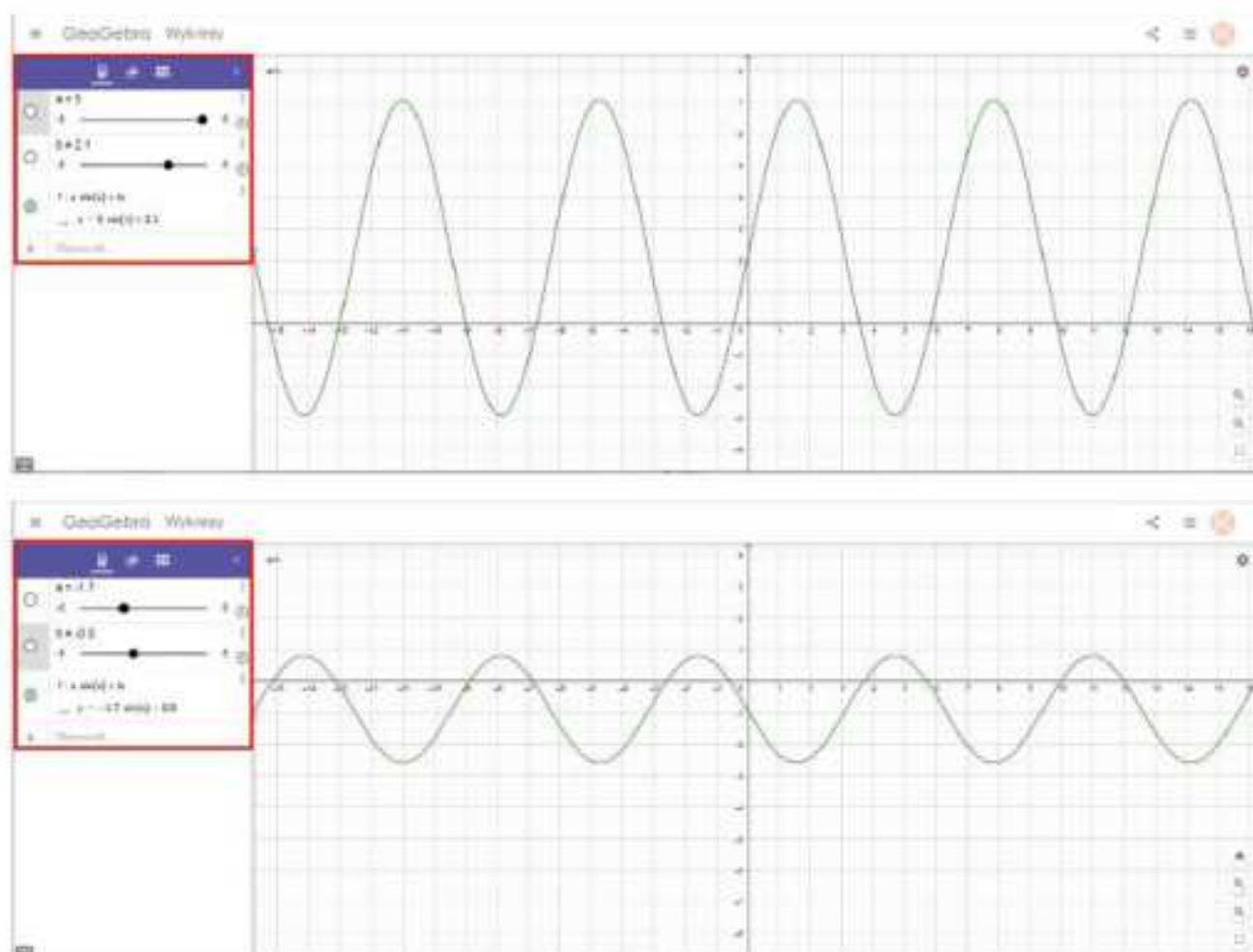
interaktywne
samouczki
GeoGebry



Rys. 59.3. Przykłady tematów samouczków z chmury geogebra.org

symulacje
w GeoGebra

Wielką zaletą GeoGebry jest **możliwość symulowania**, np. zachowania się funkcji i ich wykresów w zależności od zmian parametrów. Nie jest to możliwe przy użyciu tablicy lub zeszytu (rys. 59.4.).



Rys. 59.4. Przykład badania kształtu i położenia wykresu funkcji w zależności od parametrów



GeoGebra jest bardzo rozbudowanym narzędziem. Warto zapoznać się z jej możliwościami i wykorzystać je do ćwiczeń ułatwiających zrozumienie wielu trudnych zagadnień niezbędnych do osiągnięcia wysokiego wyniku na maturze.

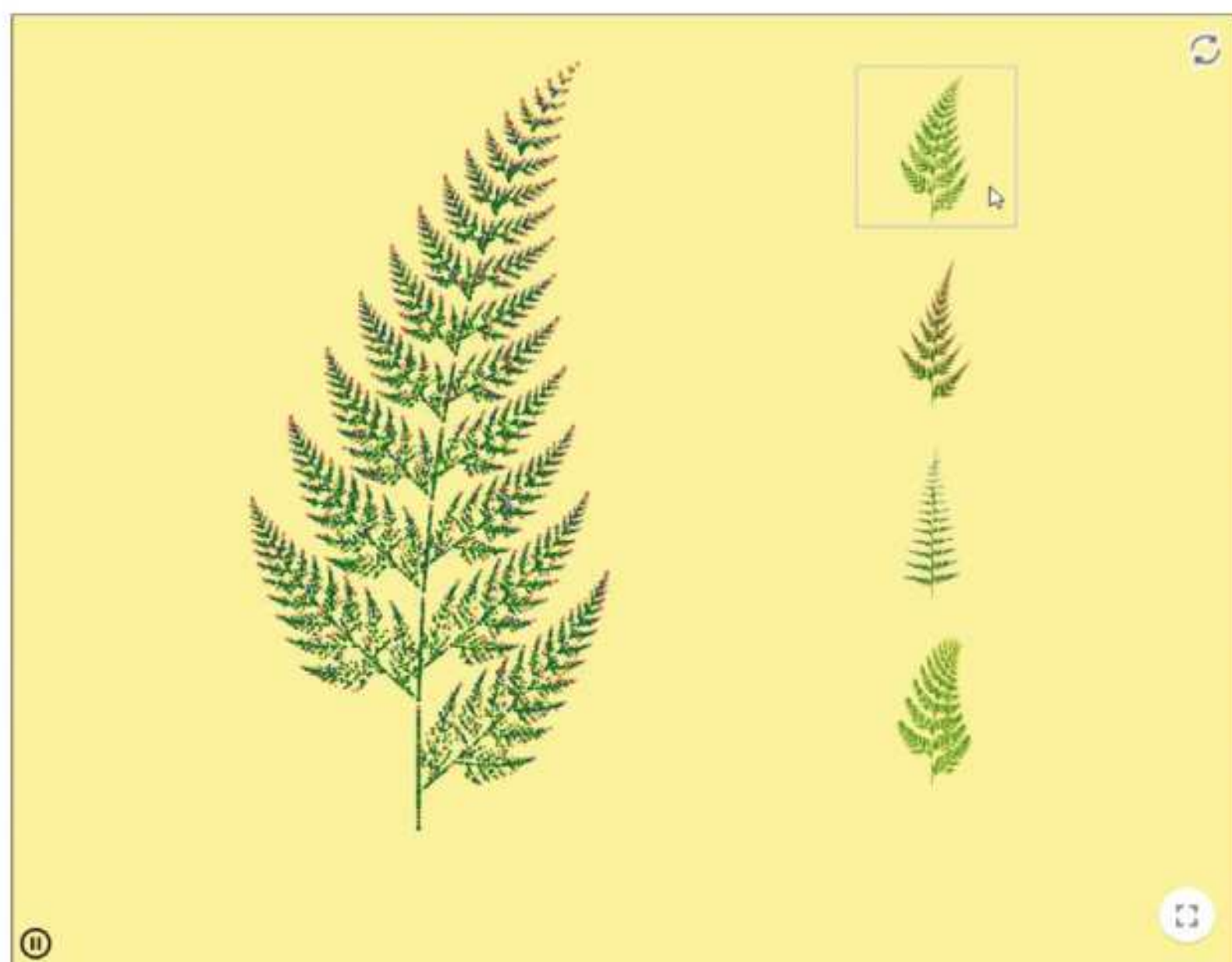
Wokół chmury zgromadziło się wielu entuzjastów, którzy **udostępniają** swoje **projekty pozostałym użytkownikom**. Możesz z nich skorzystać. Nie zawsze opłaca się wyważać otwarte drzwi. Twoje pomysły także mogą pomóc innym użytkownikom w nauce, jeśli tylko je rzetelnie opracujesz. Wśród przykładów udostępnionych przez innych użytkowników GeoGebry znajdziesz również ciekawe projekty, np. tworzenie fraktali (rys. 59.5.).

udostępnianie
projektów
GeoGebry

Paproć Barnsleya

Autor: Agata Matuszczak

Naciśnij małą miniaturkę aby zaczęła tworzyć się nowa paproć.



Rys. 59.5. Przykład fraktala paproć Barnsleya (wym. barnslija) opracowanego przez jedną z użytkowniczek chmury geogebra.org



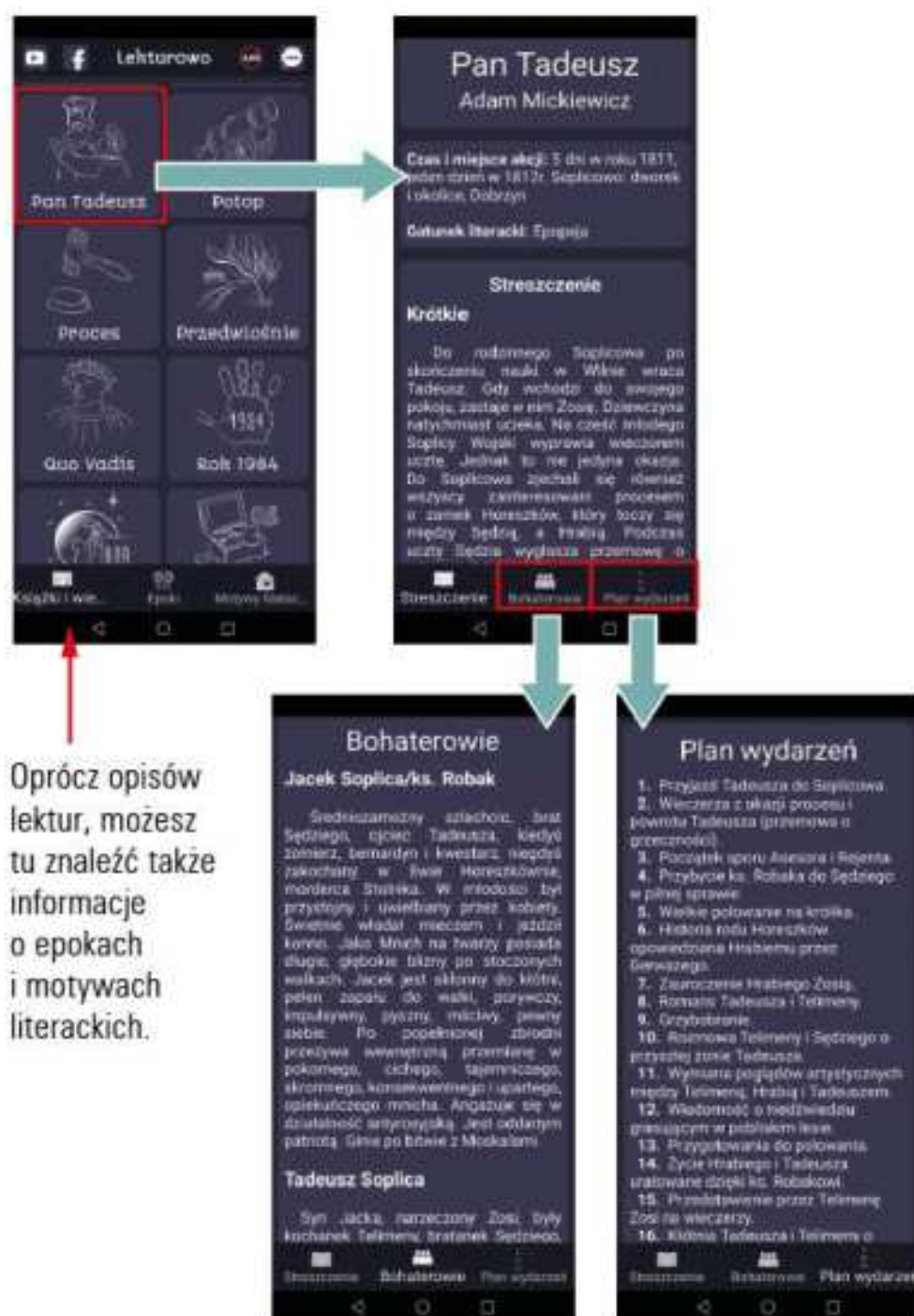
59.2. Tablice i inne aplikacje mobilne, czyli podręczna biblioteczka maturzysty

Smartfon jest narzędziem, które zawsze mamy pod ręką. Można go więc wykorzystać do nauki niezależnie od miejsca, w którym się znajdujemy: w tramwaju, autobusie, pociągu itp. Potrzebne są oczywiście odpowiednie aplikacje. Z łatwością znajdziesz je w sklepie Google. Niestety, nie wszystkie zawierają rzetelne informacje. Masz już spore doświadczenie w ocenianiu źródeł i z pewnością potrafisz wybrać te właściwe.

aplikacje
dla różnych
przedmiotów

Oczywiście aplikacje edukacyjne wspomagają naukę **różnych przedmiotów**.

Przykładem może być aplikacja **Lekturowo** zawierająca streszczenia, charakterystyki postaci i plany wydarzeń wielu lektur szkolnych (rys. 59.6.). Informacje te nie zastąpią czytania lektury, ale z pewnością pomogą w powtórkach.



Rys. 59.6. Przykład aplikacji pomagającej w powtórkę materiału z języka polskiego

Tablice Maturalne 2017

Tablice maturalne CKE 2017

MATEMATYKA

FIZYKA Z CHEMIĄ

O AUTORACH

ZASADY AZOTOWE I KWASY ORG...

ZASADY AZOTOWE I KWASY ORGANICZNE

KOD GENETYCZNY

POTENCJAŁ WODY W KOMÓRCE ROŚLINNEJ I RÓWNANIE HARDY'EGO-WEINBERGA

AMINOKWASY BIAŁKOWE

ROZPUSSZCZALNOŚĆ SOLI I WODOROTLENKÓW

STALE DYSOCJACJI

SZEREG ELEKTROCHEMICZNY

UKŁAD OKRESOWY PIERWIASTKÓW

KINEMATYKA

ORGANIA I FALE

DYNAMIKA

UKŁAD OKRESOWY PIERWIASTK...

• Zasady azotowe

pirymidowe

NC1=NC(=O)NC=C1 cytozyna (C)

CC1=CNC(=O)NC1=O tymina (T)

O=C1NC=CC(=O)N1 uracyl (U)

purynowe

NC1=NC2=C(N1)N=CN=C2 adenina (A)

NC1=NC2=C(N=CN2)C(=O)N1 guanina (G)

• kwasy organiczne

OC(CC(=O)O)C(=O)O kwas mlekowy

OC1=CC(=O)NC(=O)C1 kwas piroglutynowy

OC(CC(=O)O)C(=O)O kwas mlekowy

OC(CC(=O)O)C(=O)O kwas piroglutynowy

Wartości atomowe

H	1,01
C	12,01
N	14,01
O	16,00

Wartości atomowe pierwiastków

	1	2	3	4	5	6	7	8	9	10	11	12
1	22 Ti	23 V	24 Cr	25 Mn	26 Fe	27 Co	28 Ni	29 Cu	30 Zn	31 Ga	32 Ge	33 As
2	34 Se	35 Br	36 Kr	37 Rb	38 Sr	39 Y	40 Zr	41 Nb	42 Mo	43 Tc	44 Ru	45 Rh
3	46 Pd	47 Ag	48 Cd	49 In	50 Sn	51 Sb	52 Te	53 I	54 Xe	55 Cs	56 Ba	57 La
4	58 Ce	59 Pr	60 Nd	61 Pm	62 Sm	63 Eu	64 Gd	65 Tb	66 Dy	67 Ho	68 Er	69 Tm
5	70 Yb	71 Lu	72 Hf	73 Ta	74 W	75 Re	76 Os	77 Ir	78 Pt	79 Au	80 Hg	81 Tl
6	82 Pb	83 Bi	84 Po	85 At	86 Rn	87 Fr	88 Ra	89 Ac	90 Th	91 Pa	92 U	93 Np
7	94 Pu	95 Am	96 Cm	97 Bk	98 Cf	99 Es	100 Fm	101 Md	102 No	103 Lr	104 Rf	105 Db
8	106 Sg	107 Bh	108 Hs	109 Mt	110 Ds	111 Rg	112 Cn	113 Nh	114 Fl	115 Mc	116 Lv	117 Ts
9	118 Og	119	120	121	122	123	124	125	126	127	128	129

Wartości atomowe pierwiastków i ich symboli chemicznych podane są z dokładnością do dwóch miejsc po przecinku.

347



Korzystając z darmowych aplikacji, należy pamiętać, że niektóre wyświetlają reklamy. Zazwyczaj po zakupie aplikacji reklamy już się nie pojawiają.

ZADANIA DO WYKONANIA

1. Znajdź w internecie aplikacje dla komputerów osobistych wspomagające przygotowanie do matury z informatyki. Oceń ich zawartość merytoryczną i inne cechy. Wybierz kilka najciekawszych i godnych polecenia. Napisz ich krótkie recenzje. Udostępnij je wraz z odnośnikami koleżankom i kolegom z klasy.
2. Znajdź w sklepie Play lub App Store aplikacje wspomagające przygotowanie do matury z informatyki. Oceń ich zawartość merytoryczną i inne cechy. Wybierz kilka najciekawszych i godnych polecenia. Udostępnij ich nazwy koleżankom i kolegom z klasy, np. za pośrednictwem mediów społecznościowych.
3. Opracuj prezentację na temat „Jak ocenić jakość i wiarygodność aplikacji edukacyjnej”. Skonsultuj jej treść z koleżankami i kolegami z klasy. Uzupełnij ją, kierując się ich uwagami. Opublikuj prezentację w sieci, np. na portalu społecznościowym.
4. Wykorzystaj chmurę geogebra.org do prezentacji wpływu parametrów funkcji kwadratowej na kształt jej wykresu w układzie współrzędnych. Zarejestruj w pliku wideo zachowanie się wykresu tej funkcji podczas zmian parametrów.
- 5*. Korzystając z geogebra.org, wykonaj animację będącą dowodem twierdzenia Pitagorasa. Opisz krótko zastosowaną metodę.

PODSUMOWANIE LEKCJI

- s. 342 • Geogebra.org jest **darmową chmurą** zawierającą wiele narzędzi informatycznych wspomagających naukę matematyki.
- s. 342 • Możesz korzystać z GeoGebry w wersji **on-line lub aplikacji** zainstalowanej w PC i urządzeniu mobilnym.
- s. 343 • Dzięki **interaktywnym samouczkom** możesz zgłębić tajniki aplikacji chmury geogebra.org.
- s. 344 • Możliwość **obserwowania zmian wartości obliczeń lub kształtu funkcji** w zależności od wartości parametrów jest dużą zaletą programu i ułatwia zrozumienie danego tematu.
- s. 345 • **Osoby** pracujące w chmurze geogebra.org **udostępniają swoje projekty** pozostałym użytkownikom.
- s. 346 • W sieci są dostępne aplikacje ułatwiające przygotowanie się do egzaminów. Szczególnie przydatne są **programy dla urządzeń mobilnych**, ponieważ pozwalają wykorzystać np. czas podróży do powtórek i nauki.



60. Rozwiązywanie testów pomaga w nauce, czyli jak aplikacje testują wiedzę

NA TEJ LEKCJI:

- dowiesz się, w jaki sposób dotrzeć do testów maturalnych z poprzednich lat;
- poznasz przykłady aplikacji i stron internetowych z interaktywnymi testami maturalnymi.

Wiesz już, jak szukać stron i aplikacji pomagających w dobrym przygotowaniu się do prac klasowych i matury. Zanim przystąpisz do egzaminu, warto byłoby sprawdzić poziom wiedzy i umiejętność rozwiązywania zadań. W tym mogą pomóc strony i aplikacje. To jednak nie wystarczy. Do odniesienia sukcesu potrzebna ci będzie także wiedza o tym, jak twoje prace będą sprawdzane. Szczególnie dotyczy to zadań otwartych. Warto wiedzieć, według jakiego klucza sprawdzane są prace egzaminacyjne.

60.1. Dostęp do oryginałów, czyli odwiedzamy Centralną Komisję Egzaminacyjną

Na wielu stronach internetowych są publikowane zadania z rozmaitych przeprowadzonych już egzaminów. Jednak najlepiej szukać ich u źródła, czyli na stronach organizatora. Oficjalne, państwowe egzaminy zawodowe i maturalne organizuje Centralna Komisja Egzaminacyjna.

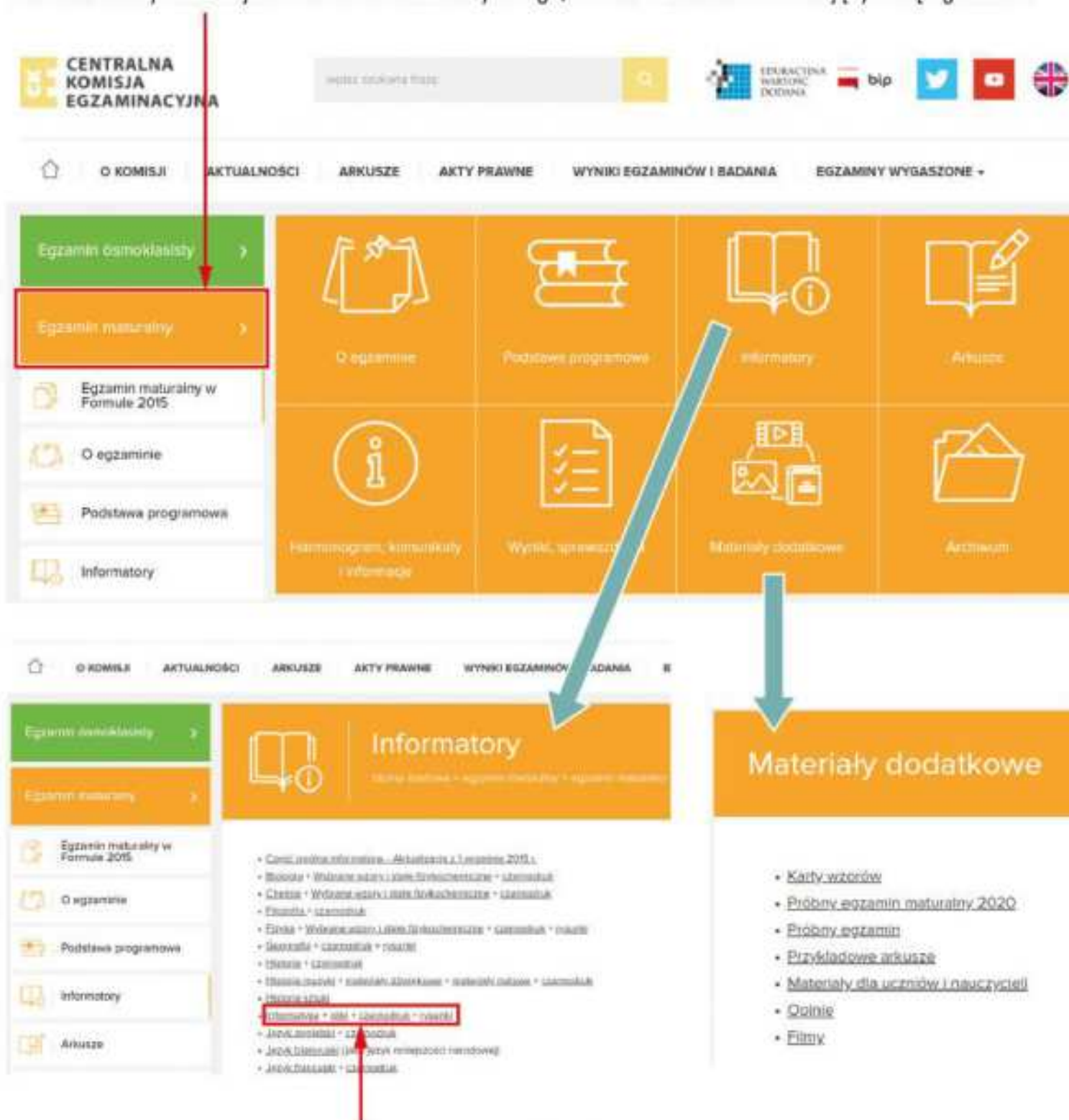
zadania CKE

To do niej należy opracowanie testów i zadań, druk arkuszy egzaminacyjnych, ich dystrybucja oraz odbiór prac. **Okręgowa Komisja Egzaminacyjna** podlegająca CKE odpowiada między innymi za przeprowadzenie egzaminów, sprawdzenie prac i ogłoszenie wyników. Każdy zdający otrzymuje osobiste hasło dostępu do konta, na którym może poznać liczbę zdobytych punktów. Ze względu na RODO nie wywiesza się już w szkołach list z punktacją.

Zajrzyjmy na stronę CKE (rys. 60.1.).



W menu strony CKE znajdziesz odnośniki do wszystkiego, co warto wiedzieć o czekającym cię egzaminie.



Informatyka: kompletny informator w formacie PDF, pliki: pliki do przykładowych zadań, czarnodruk: treść przykładowych zadań, rysunki: ilustracje do przykładowych zadań.

Rys. 60.1. Przykład ścieżki dostępu do wybranych informacji o egzaminie maturalnym

Zwróć uwagę, że na stronie CKE publikuje się wszystkie informacje dotyczące egzaminów, a także podstawy programowe. Dzięki temu powstają zadania, przykładowe arkusze egzaminacyjne itp. Dla zdających ważne są także komunikaty o harmonogramie i przebiegu egzaminu. Co najważniejsze, informacje te są zawsze aktualne.

zasady oceniania
na stronie CKE

Szczególną uwagę należy zwrócić na wytyczne dla egzaminatorów. Każdy sprawdzający pracę otrzymuje **zasady oceniania**, według których powinien sprawdzać i punktować poszczególne odpowiedzi (rys. 60.2.). Ich znajomość ułatwia rozwiązywanie zadań, szczególnie tych o charakterze otwartym.



Rys. 60.2. Przykład schematu punktowania danego egzaminu maturalnego z informatyki

Zbiór wykorzystanych arkuszy egzaminacyjnych jest dostępny dla każdego przedmiotu (rys. 60.3.).



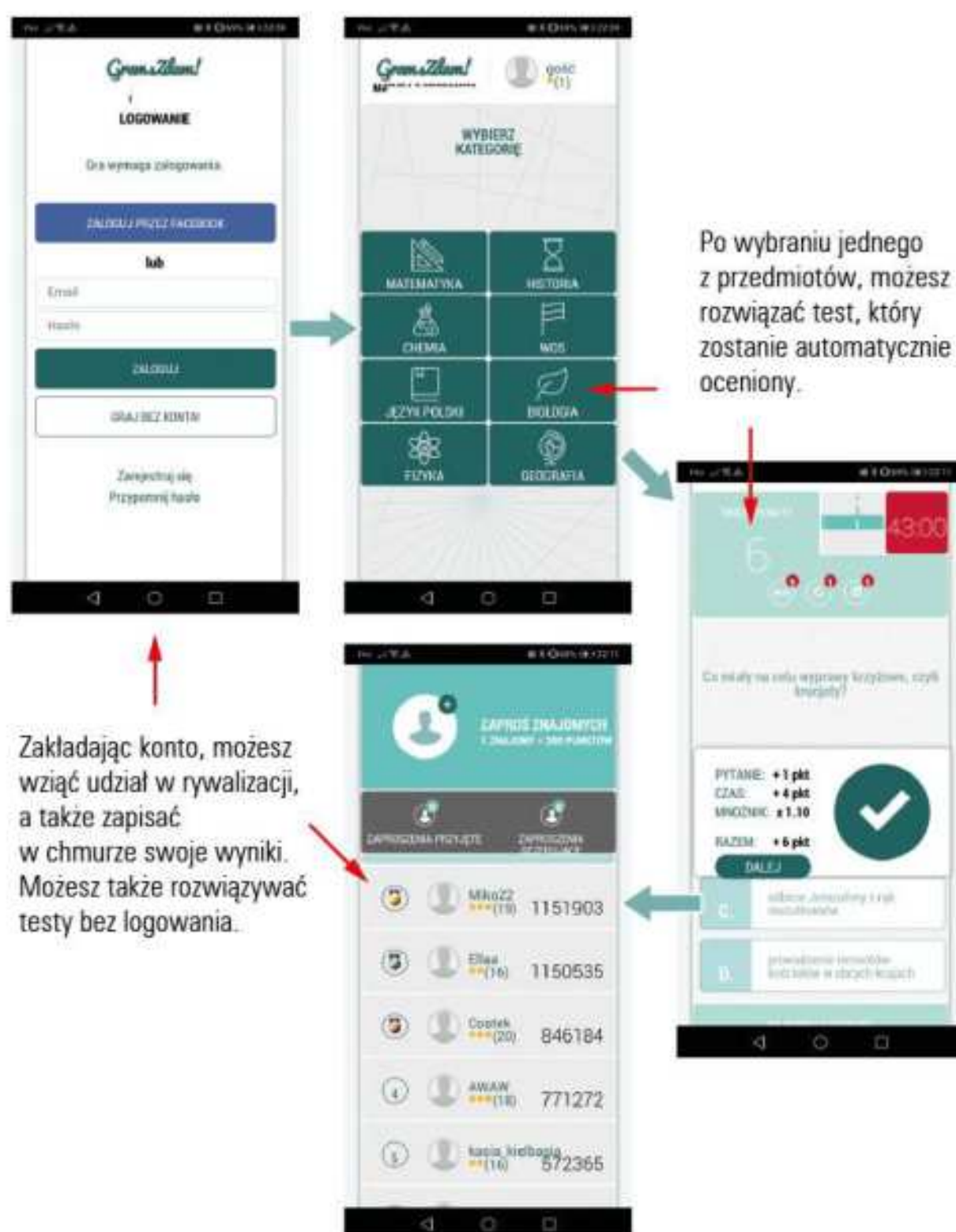
Rys. 60.3. Zadania egzaminacyjne dostępne na stronie CKE po zakończeniu egzaminu



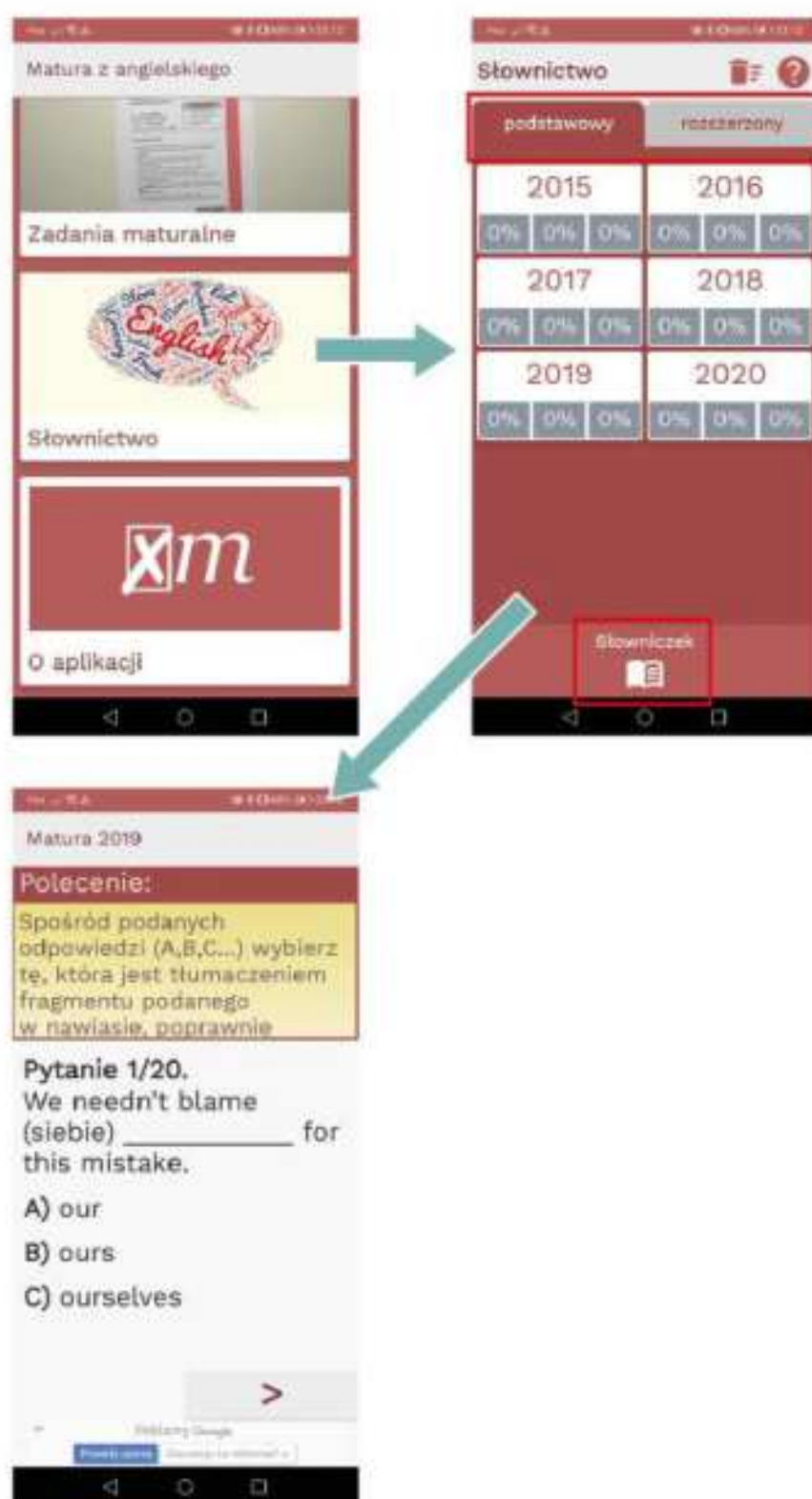
60.2. Testy w telefonie, czyli ostatni etap nauki do egzaminu

Samodzielne testowanie swoich umiejętności i wiedzy niezbędnej do zdania egzaminu jest jedną z końcowych faz przygotowań. Jeśli skorzystacie z arkuszy pobranych ze strony CKE, to w przypadku testów powinniście je wydrukować, rozwiązać i porównać swoje rozwiązania z kluczem. Oczywiście można to sobie ułatwić, korzystając z aplikacji, np. Gram&Zdam (rys. 60.4.). Możesz w niej wybrać jeden z przedmiotów maturalnych, rozwiązywać samosprawdzające się testy i rywalizować z innymi użytkownikami. Najważniejszym jednak zadaniem tej aplikacji jest weryfikacja postępów w przygotowaniu do matury. Warto z niej skorzystać. Aplikacja nie zawiera reklam i jest całkowicie darmowa.

aplikacje
wspomagające
naukę



Rys. 60.4. Aplikacja Gram&Zdam z możliwością rywalizacji pomiędzy użytkownikami



Rys. 60.5. Przykład aplikacji z zadaniami maturalnymi z języka angielskiego

Uwaga!

W sklepach Play i App Store pojawiają się coraz to nowe aplikacje pomagające w nauce. Wybieraj mądrze, np. te udostępniane przez wydawnictwa pedagogiczne i instytucje oraz profesjonalnie prowadzone portale zajmujące się edukacją.



ZADANIA DO WYKONANIA

1. Odwiedź stronę Centralnej Komisji Egzaminacyjnej, <https://cke.gov.pl/>. Wykonaj mapę tej strony w postaci grafu, dzięki której bez błędu trafisz do danego zasobu związanego z maturą lub egzaminem zawodowym z dowolnego przedmiotu lub kwalifikacji.
2. Na stronie CKE odnajdź zadania maturalne z informatyki. Wybierz dowolne z części pierwszej lub drugiej i wykonaj polecenia z arkusza. Na podstawie wyniku oceń swój poziom wiedzy z informatyki.
3. Skorzystaj z aplikacji Gram&Zdam. Wybierz jeden z testów egzaminacyjnych i go rozwiąż. Zapamiętaj np. poprzez zrzut ekranu zadania, które sprawiają ci problemy. Wybierz szczególnie te dotyczące tematów już przerabianych na lekcjach. Znajdź w podręczniku lub internecie prawidłowe rozwiązania.
4. Przeglądaj propozycje aplikacji wspomagających przygotowania do egzaminów w sklepie Play lub App Store. Przeczytaj opinie. Sprawdź przydatność wybranych aplikacji. Oceń ich jakość.

PODSUMOWANIE LEKCJI

- s. 350 • Na stronie CKE znajdziesz nie tylko wszelkie informacje o egzaminach i zadania z odbytych już egzaminów, lecz także **zasady oceniania**.
- s. 352 • W sieci znajdziesz wiele aplikacji wspomagających przygotowanie do egzaminów. Weryfikuj je pod kątem wiarygodności i treści.



PODSUMOWANIE PRZED SPRAWDZIANEM

55. Korzystamy z office.com, czyli jak wykorzystać aplikacje chmury w nauce

- **To Do** jest aplikacją działającą w połączeniu z kontem Microsoft. s. 318
- Można ją instalować **w urządzeniach przenośnych i komputerach**. Aplikacje synchronizują się ze sobą. Zmiana w jednej instalacji tego samego użytkownika będzie widoczna na pozostałych. s. 320
- Opcje w aplikacji To Do pomagają w organizacji pracy i nauki. **Przypomnienie** powoduje wyświetlanie komunikatu o czekającym nas zadaniu na wszystkich urządzeniach z aplikacją danego użytkownika. s. 320
- W Sway, podobnie jak innych aplikacjach office.com, można używać oferowanych przez chmurę obiektów **upowszechnianych na licencji CC**. Wyszukiwane są kontekstowo, co przyspiesza ten proces. s. 321
- **Sway** podobnie jak To Do jest aplikacją, którą **można zainstalować stacjonarnie i w smartfonach**. s. 321
- Prezentacje Sway można udostępniać za pośrednictwem **odnośnika** i wklejać w treść strony dzięki udostępnionemu **kodowi HTML**. s. 322

56. Nie tylko w firmie, czyli wykorzystanie aplikacji komunikacyjnych w nauczaniu

- Działalność firmy można **przenieść w całości lub w części do chmury**. Wtedy pracownicy mogą pracować w domu. s. 325
- Zastosowanie chmury i pracy zdalnej pozwala także na usunięcie barier (np. długi czas dojazdu, brak windy w miejscu pracy) i **wykluczenia zawodowego** osób niepełnosprawnych. s. 326
- Spotkania w Teams można organizować dla wskazanych **użytkowników** lub członków tworzonych zespołów. s. 329
- **Zespołami można zarządzać** za pomocą zmian uprawnień i dodawania lub usuwania członków. s. 330
- Do utworzenia nowego spotkania może posłużyć **kalendarz**. s. 332
- Przed spotkaniami **można przygotować i udostępnić** uczestnikom materiały w postaci plików, odnośników lub dokumentów. s. 333

57. Każdy ma notatki, czyli jak wykorzystać chmurę do wspólnej nauki

- Chmury wyróżniają się cechami, które można wykorzystać w organizowaniu pracy zespołu, np. przygotowującego się do egzaminu. s. 335
- Zakres **wykorzystania tych cech** zależy od użytkowników. Dobrze zaplanowana praca z chmurą da dobre wyniki. s. 335



- s. 337 • Dzięki właściwościom chmury korzystanie z oprogramowania i zasobów **nie zależy od rodzaju komputera osobistego lub smartfona**.
- s. 337 • Za pośrednictwem chmury można także zorganizować **zdalne nauczanie lub pomoc koleżeńską** osobom nieobecnym na lekcjach.

58. Walidacja, czyli sprawdzamy wyniki swojej pracy

- s. 339 • Nieoficjalnie obowiązująca zasada **1-10-100** mówi o tym, jak rosną koszty wyprodukowania programu lub strony w stosunku do etapu jego powstawania.
- s. 340 • **Walidacja** to kontrola jakości kodu oparta na standardach. W przypadku stron internetowych są to standardy ustanowione przez organizację **World Wide Web Consortium** (<https://www.w3.org/>).
- s. 340 • **W3C udostępnia narzędzie do walidacji** kodu strony internetowej zarówno umieszczonego w sieci, jak i zapisanego w pliku.
- s. 340 • Narzędzie do walidacji oznacza miejsca z **błędami (Error)** i miejsca, które **wymagają uwagi (Warning)**.

59. Matura, czyli jak komputery wspomagają przygotowanie do egzaminu

- s. 342 • Geogebra.org **jest darmową chmurą** zawierającą wiele narzędzi informatycznych wspomagających naukę matematyki.
- s. 342 • Możesz korzystać z GeoGebry w wersji **on-line lub aplikacji** zainstalowanej w PC i urządzeniu mobilnym.
- s. 343 • Dzięki **interaktywnym samouczkom** możesz zgłębić tajniki aplikacji chmury geogebra.org.
- s. 344 • Możliwość **obserwowania zmian wartości obliczeń lub kształtu funkcji** w zależności od wartości parametrów jest dużą zaletą programu i ułatwia zrozumienie danego tematu.
- s. 345 • **Osoby** pracujące w chmurze geogebra.org **udostępniają swoje projekty** pozostałym użytkownikom.
- s. 346 • W sieci są dostępne aplikacje ułatwiające przygotowanie się do egzaminów. Szczególnie przydatne są **programy dla urządzeń mobilnych**, ponieważ pozwalają wykorzystać np. czas podróży do powtórek i nauki.

60. Rozwiązywanie testów pomaga w nauce, czyli jak aplikacje testują wiedzę

- s. 350 • Na stronie CKE znajdziesz nie tylko wszelkie informacje o egzaminach i zadania z odbytych już egzaminów, lecz także **zasady oceniania**.
- s. 352 • W sieci znajdziesz wiele aplikacji wspomagających przygotowania do egzaminów. Weryfikuj je pod kątem wiarygodności i treści.