

Informatyka

ZAKRES PODSTAWOWY



Zmiana MEN 2024



Oznaczono usunięte
i zmienione treści

1

Podręcznik dla **szkół ponadpodstawowych**

Wojciech Hermanowski

Informatyka

Podręcznik dla **szkół ponadpodstawowych**

1

ZAKRES PODSTAWOWY

 **OPERON**
Edukacja jest podróżą

2024

Projekt okładki: Paweł Kwacz
Redaktor prowadzący: Sebastian Przybyszewski
Redakcja językowa: zespół
Redakcja graficzna i skład: zespół
Korekta: Katarzyna Kupracz
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.

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. hab. Henryka Nogi, dr. Przemysława Ogonowskiego, dr. Iwony Grygiel.

Etap edukacyjny: III
Typ szkoły: szkoła ponadpodstawowa
Rok dopuszczenia: 2019
Numer dopuszczenia: 1052/1/2019

© Copyright by OPERON Sp. z o.o. & Wojciech Hermanowski
Gdynia 2019

Wszelkie prawa zastrzeżone. Kopiowanie w całości lub we fragmentach bez zgody wydawcy zabronione.
N6843

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 978-83-7879-948-1

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. Przestrzeganie prawa i zasad bezpieczeństwa podczas pracy przy komputerze

1. Bądź uczciwy, czyli przestrzeganie prawa w świecie informatyki.....	8	
2. 8, 16, 32, 64, czyli jak rozwój technologii wpływa na rozwój społeczeństw.....	16	
3. Kim jestem, czyli jak bezpiecznie budować wizerunek w sieci.....	22	
Podsumowanie przed sprawdzianem	28	

II. Podstawy programowania i środowisko programistyczne

4. Przypomnij sobie, czyli podstawy tworzenia algorytmów	30
5. Narzędzia, czyli jak dobrać i skonfigurować środowisko programistyczne	36
6. C z plusami, czyli uruchamiamy pierwsze programy w C++	47
7. C z plusem, czyli podstawy programowania.....	54
8. C z plusem, czyli podstaw programowania ciąg dalszy	62
Podsumowanie przed sprawdzianem	68

III. Programowanie i algorytmy

9. Przez 1 i siebie, czyli jak zbadać, czy dana liczba jest liczbą pierwszą.....	70
10. Która pierwsza, czyli jak program bada liczby	75
11. Największy i najmniejszy, czyli jak znaleźć NWD i NWW	81
12. Dodawanie ułamków, czyli jak wykorzystać NWW i NWD w programie komputerowym ..	87
13. Skracamy i wyłączamy, czyli dodawania ułamków ciąg dalszy.....	92
14. Szyfrowanie, czyli poznajemy szyfr Cezara.....	97
15. Konspiracja, czyli tworzymy własne szyfry.....	102
16. Automat wydaje, czyli jak dynamicznie wydawać resztę	110
Podsumowanie przed sprawdzianem	120

IV. Grafika i druk

17. Modele w przestrzeni, czyli podstawy druku 3D.....	122
18. Modelujemy, czyli jak projektować obiekty 3D	129
19. Wizualizacja pomysłów, czyli projektujemy dom w edytorze 3D.....	138
20. Wypukłości nie tylko 3D, czyli tworzymy elementy graficzne publikacji.....	147
21. Z wydruku do komputera, czyli skanujemy i odczytujemy dokumenty.....	156
22. Dokumentujemy wydarzenia, czyli aktywna praca z aparatem fotograficznym.....	164
23. Szturmowiec w chmurze, czyli poprawiamy zdjęcia w edytorze grafiki rastrowej	172
24. Tego tu nie było, czyli poprawiamy rzeczywistość na fotografiach.....	180
25. Własny film, czyli jak twórczo wykorzystać kamerę	185
26. Klatki do komputera, czyli jak transmitować strumieniowo	194
Podsumowanie przed sprawdzianem	201

V. Komputer w sieci

27. Nie wszystko jest takie oczywiste, czyli jak działa internet	204
28. Sieć to nie tylko internet, czyli poznajemy topologię sieci komputerowych.....	213
29. Kto tam, czyli identyfikujemy komputery w sieci.....	218
30. Kupujemy świadomie, czyli poznajemy parametry urządzeń peryferyjnych	223
Podsumowanie przed sprawdzianem	231

Wstęp

Jak jest zbudowany nasz podręcznik

Podręcznik pozwala na pełne wykorzystanie nowoczesnych metod dydaktycznych w nauczaniu informatyki. Zgodnie z założeniami reformy programowej uczniowie I klasy liceów i techników rozpoczną naukę tego przedmiotu w zakresie podstawowym w zwiększonym wymiarze godzin. Program nauczania przewiduje naukę informatyki w klasach 1–3 w wymiarze po 1 godzinie tygodniowo.

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 dział podręcznika otwiera spis tematów z danego obszaru.

Bezpośrednio pod tytułem rozdziału znajdziecie moduł „Na tej lekcji” – jego zadaniem jest ukierunkowanie was na cele realizowane w danym temacie.

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

Przystępny język publikacji ułatwi zrozumienie treści i pozytywnie nastawi was do powtarzanego materiału.

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

Schematy i rysunki pomogą wam usystematyzować wiedzę.

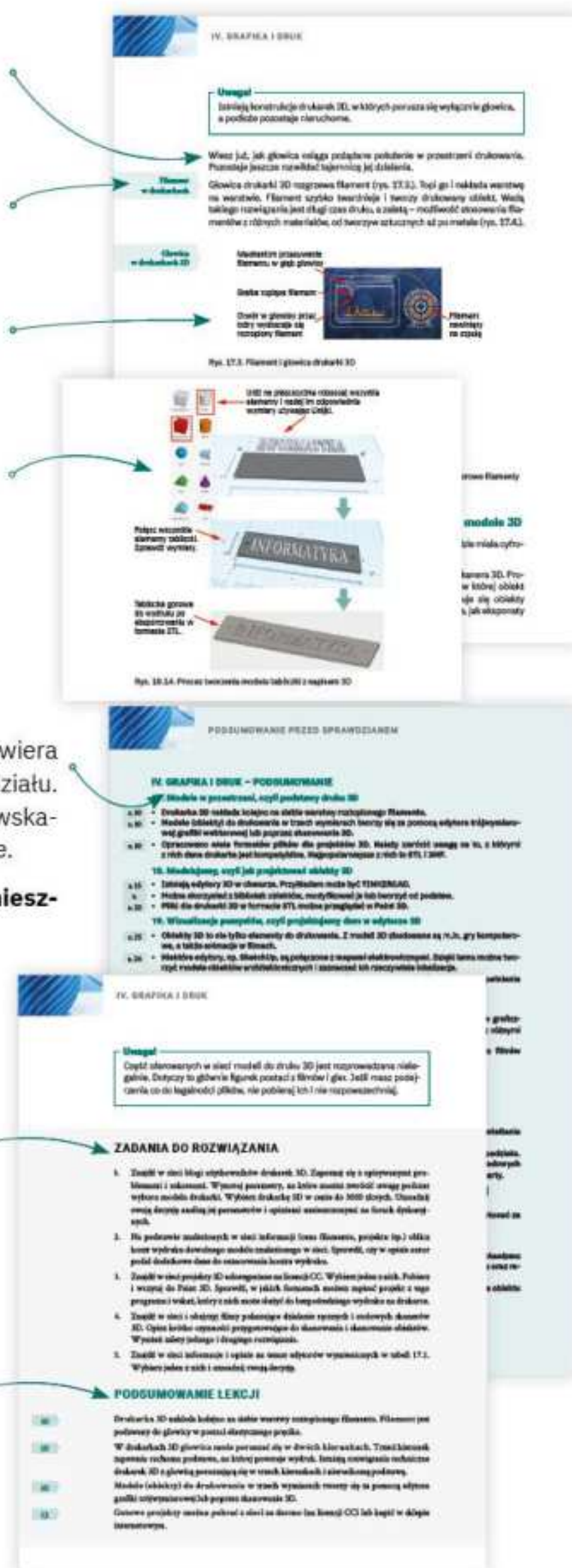
Zrzuty ekranu pełnią funkcję tutoriali, dzięki którym będziecie mogli szybko je zastosować w praktyce.

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

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

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

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.



I. Przestrzeganie prawa i zasad bezpieczeństwa podczas pracy przy komputerze

▶ Prawo w informatyce

▶ Ustawa o prawie autorskim

▶ Postęp technologiczny a wydajność komputera

▶ E-learning

▶ Wizerunek w sieci

▶ Prywatność w sieci





1. Bądź uczciwy, czyli przestrzeganie prawa w świecie informatyki

NA TEJ LEKCJI:

- dowiesz się, jak legalnie korzystać z cudzych publikacji;
- poznasz ustawę o prawie autorskim i prawach pokrewnych.

Korzystasz z internetu od wielu lat – czytasz informacje, oglądasz filmy, słuchasz muzyki. Czy wiesz, skąd biorą się te materiały? Kto je tworzy? Kto publikuje? Czy robi to za darmo, czy może dostaje wynagrodzenie, a może utrzymuje się z tantiem? To bardzo ważne pytania w kontekście uczciwości i przestrzegania prawa w sieci. Co więc jest legalne, a co nie? Co jest dopuszczalne, a czego nie należy tolerować?

1.1. Prawa majątkowe

Nie trzeba być hakerem lub oszustem internetowym, by złamać prawo. Często niestety robi się to nieświadomie, ale – jak wiadomo – nieznajomość prawa nie zwalnia z odpowiedzialności. Być może zastanawiasz się teraz, czy ty również popełniasz przestępstwa lub wykroczenia, nie wiedząc o tym. Spróbuj samodzielnie odpowiedzieć na to pytanie, rozważając poniższe przykłady.

Przykład 1.

Złodziej wchodzi do sklepu z płytami z muzyką lub filmami. Podchodzi do półki, zdejmuje z niej płytę i chowa pod kurtkę. Wychodzi ze sklepu bez zapłacenia za towar, przez co popełnia przestępstwo. Czy pobierając utwory z tej płyty w postaci plików bez wnoszenia opłaty, dokonujesz takiego samego czynu, jak ów złodziej?

pobieranie treści
z internetu

Odpowiedź na to pytanie jest złożona. Z jednej strony bezpłatne pobranie piosenek niczym nie różni się od wyniesienia ze sklepu płyty i również jest moralnie naganne. Jednak polskie prawo pozwala na pobieranie utworów opublikowanych w sieci na własny użytek. W takim przypadku mówi się o dozwolonym użytku. Powinniśmy być przy tym świadomi, że w ten sposób pozbawiamy twórcę i wydawcę należnego zysku. Jeśli pliki pobieramy na przykład z sieci Torrent, to rozpoczęcie pobierania jest równoznaczne z udostępnianiem pliku (a nawet fragmentu pliku) innym użytkownikom tej sieci. Ponieważ udostępnianie treści chronionych prawem autorskim jest przestępstwem, nieświadomie można stać się przestępcą. Należy liczyć się z konsekwencjami złamania prawa.

**Przykład 2.**

Student musi wykonać pracę zaliczeniową. Do jej wykonania potrzebuje programu komputerowego, na który go nie stać. Pobiera więc piracką wersję tego programu, instaluje na swoim komputerze i tworzy swoją pracę. Czy jest złodziejem?

Każde pobranie, zainstalowanie lub użycie pirackiej wersji programu komputerowego jest przestępstwem.

Przykład 3.

Ukazała się najnowsza wersja pewnej popularnej gry komputerowej. Wiadomo, że tego typu wydawnictwa są kosztowne. Pewien gracz pobiera piracką wersję, instaluje na komputerze i gra. Czy jest złodziejem?

W świetle prawa każde ściągnięcie z internetu programu komputerowego lub gry bez zgody właściciela praw (czyli np. pobranie z pirackiego serwera) jest złamaniem prawa. To taka sama kradzież jak wyniesienie pudełka z grą ze sklepu bez zapłaty. Nie ma tu wyjątków. Współczesne gry aktywnie korzystają z internetu, w związku z tym komputer z pirackim programem jest łatwy do namierzenia.

Przykład 4.

Pewien uczeń przygotowuje wypracowanie. Znajduje w sieci pracę na podobny temat. Pobiera ją, przepisuje do zeszytu i oddaje nauczycielowi do sprawdzenia. Czy jest złodziejem?

Z pewnością dokonał kradzieży czyjejś pracy. Nie szanuje też praw autorskich, ponieważ podpisał ją własnym nazwiskiem. Popełnił w ten sposób plagiat. Ustawa o prawach autorskich i prawach pokrewnych w takim przypadku mówi, że:

Art. 115.

1. Kto przywłaszcza sobie autorstwo albo wprowadza w błąd co do autorstwa całości lub części cudzego utworu albo artystycznego wykonania, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 3.
2. Tej samej karze podlega, kto rozpowszechnia bez podania nazwiska lub pseudonimu twórcy cudzy utwór w wersji oryginalnej albo w postaci opracowania, artystyczne wykonanie albo publicznie zniekształca taki utwór, artystyczne wykonanie, fonogram, wideogram lub nadanie.



Plagiat jest więc przestępstwem przeciwko prawu. Jeśli taki czyn popełni student lub naukowiec, poniesie jeszcze kary przewidziane przez uczelnię i oczywiście utraci autorytet i wiarygodność. W przypadku ucznia oczywistą reakcją jest ocena niedostateczna i obniżenie oceny z zachowania. Nawet jeśli oszustwo się uda, to pozostaje kwestia sensowności takiego działania. Kopiując pracę, nie uczysz się, a więc jej cel nie zostaje spełniony.

Przykład 5.

W kabriolecie na siedzeniu leży telefon. Przechodzień dostrzega go i zabiera. Czy to, że urządzenie było dostępne w łatwy sposób, usprawiedliwia kradzież? Czy to, że komercyjne programy, gry i aplikacje są dostępne w sieci oznacza, że można je pobierać bez opłaty?

Odpowiedź na to pytanie jest oczywista.



Rys. 1.1. Przykłady kradzieży za pośrednictwem internetu

A teraz zastanów się i odpowiedz na jeszcze jedno pytanie: czy kiedyś dopuściłeś/dopuściłaś się kradzieży. Wyciągnij wnioski.



1.2. Prawa autorskie, czyli jak legalnie korzystać z cudzych publikacji

W przepisach używa się pojęć, które musisz zrozumieć, aby znać prawo. Wiesz, czym jest utwór? Warto to wiedzieć, ponieważ właśnie utwory są chronione prawem. Pojęcie to zdefiniowano w rozdziale 1 Ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (DzU 1994, nr 24, poz. 83, z późn. zm.).

prawa autorskie

Rozdział 1.

Przedmiot prawa autorskiego

Art. 1.

1. Przedmiotem prawa autorskiego jest każdy przejaw działalności twórczej o indywidualnym charakterze, ustalony w jakiejkolwiek postaci, niezależnie od wartości, przeznaczenia i sposobu wyrażenia (utwór).
2. W szczególności przedmiotem prawa autorskiego są utwory:
 - 1) wyrażone słowem, symbolami matematycznymi, znakami graficznymi (literackie, publicystyczne, naukowe, kartograficzne oraz programy komputerowe);
 - 2) plastyczne;
 - 3) fotograficzne;
 - 4) lutnicze;
 - 5) wzornictwa przemysłowego;
 - 6) architektoniczne, architektoniczno-urbanistyczne i urbanistyczne;
 - 7) muzyczne i słowno-muzyczne;
 - 8) sceniczne, sceniczno-muzyczne, choreograficzne i pantomimiczne;
 - 9) audiowizualne (w tym filmowe).

Zatem programy komputerowe, a więc aplikacje, gry, programy sieciowe itp., są dziełami chronionymi. Teraz warto zadać sobie pytanie: w jaki sposób zabezpieczyć swoje prawa na przykład jako programista. Oczywiście to także wyjaśnia ustawa.

Art. 1.

- 2¹. Ochroną objęty może być wyłącznie sposób wyrażenia; nie są objęte ochroną odkrycia, idee, procedury, metody i zasady działania oraz koncepcje matematyczne.
3. Utwór jest przedmiotem prawa autorskiego od chwili ustalenia, chociażby miał postać nieukończoną.
4. Ochrona przysługuje twórcy niezależnie od spełnienia jakichkolwiek formalności.

Art. 4.

Nie stanowią przedmiotu prawa autorskiego:

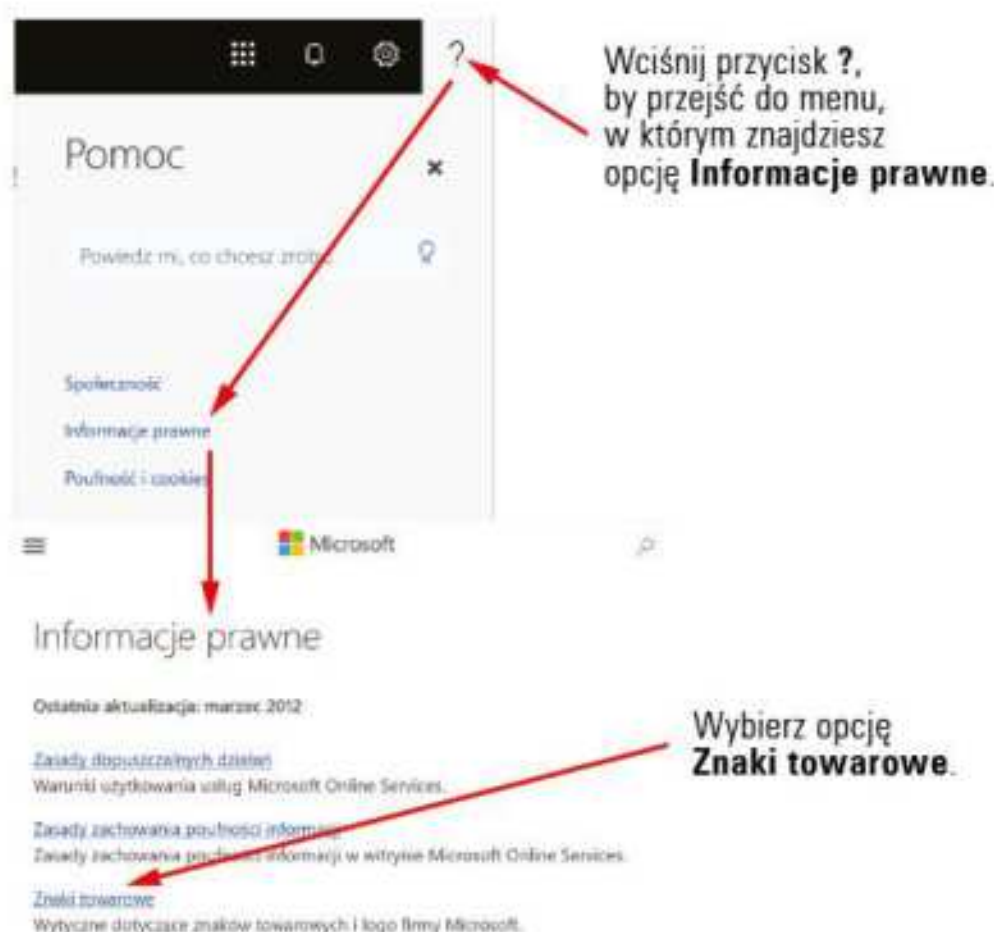
- 1) akty normatywne lub ich urzędowe projekty;
- 2) urzędowe dokumenty, materiały, znaki i symbole;
- 3) opublikowane opisy patentowe lub ochronne;
- 4) proste informacje prasowe.



Warto zastanowić się, na czym polega ochrona wytworów ludzkiego umysłu wymienionych w punkcie 3 art. 1. Aby chronić je prawem, należy je opatentować w Urzędzie Patentowym. Może to zrobić każdy po złożeniu odpowiedniego zgłoszenia i dokumentacji opisującej przedmiot patentu.

Oczywiście przepisy szczegółowe są skomplikowane i często nie zdajemy sobie sprawy, że niektóre elementy urządzeń komputerowych (także innych), części kodu programów komputerowych, technologie i rozwiązania stosowane w układach scalonych itp. są chronione prawem patentowym lub autorskim. Zazwyczaj mowa o tym w informacji dołączonej do urządzenia czy programu, gdzie znajdziemy słowa: „Wszelkie prawa zastrzeżone”. Prawo patentowe chroni także znaki towarowe, w tym nazwy własne programów, na przykład Office 365, ikony systemu operacyjnego, programów komputerowych, a nawet kroje liter używane w grach, filmach.

Uruchom dowolny program, na przykład MS Word lub chmurę Office 365, i znajdź zapisy informujące o ochronie praw. Jak myślisz, w której opcji menu je znajdziesz? Zazwyczaj są one zapisane w **Pomocy** lub informacjach **O programie**. Nie będziesz mieć kłopotów z ich odnalezieniem (rys. 1.2.).



Rys. 1.2. Opcja wyświetlania wszystkich znaków towarowych chronionych prawem będących własnością Microsoft i wykorzystywanych w Office 365

Uwaga!

W przypadku Office 365 znaków towarowych jest tak wiele, że posegregowano je alfabetycznie i utworzono menu wyszukiwania. Aby je odczytać, należy wybrać odpowiednią literę.



Należy odróżnić prawa osobiste od praw majątkowych do utworów. Pierwsze są niezbywalne i nie wygasają, co oznacza, że pozostają zawsze przy jego autorze – nawet gdy sprzeda on lub odstąpi prawa majątkowe do dysponowania utworem i zarabiania na jego publikowaniu. Natomiast prawa majątkowe są zbywalne, a po ich nabyciu można czerpać z nich zyski. Należy jednak oznaczyć utwór nazwiskiem jego autora, tak jak w przypadku książek (chyba że w umowie zapisano inne warunki). Prawa majątkowe wygasają po 70 latach od śmierci twórcy lub pierwszego upowszechnienia dzieła.

Z doświadczenia wiesz, że wiele utworów, artykułów czy opracowań zawiera fragmenty innych dzieł. Czy to jest legalne? Tak – ale po spełnieniu pewnych warunków. Określono je w artykule 29 ustawy o prawie autorskim i prawach pokrewnych:

ustawa o prawie
autorskim
i prawach
pokrewnych

Art. 29. Wolno przytaczać w utworach stanowiących samoistną całość urywki rozpowszechnionych utworów oraz rozpowszechnione utwory plastyczne, utwory fotograficzne lub drobne utwory w całości, w zakresie uzasadnionym celami cytatu, takimi jak wyjaśnianie, polemika, analiza krytyczna lub naukowa, nauczanie lub prawami gatunku twórczości.

Cytat musi więc spełniać funkcję posiłkową, na przykład stanowić dowód na poparcie tezy, ilustrować omawiane zagadnienia, być przedmiotem analizy, recenzji lub polemiki. Cytat należy wyraźnie oznaczyć. Trzeba pamiętać, że nie wolno dokonywać w nim jakichkolwiek zmian. Najważniejsze jest jednak wskazanie autora. Przykład tekstu, w którym można zacytować utwór, to recenzja. Jej autor, omawiając dzieło, często posługuje się jego fragmentami. Twórcy korzystają również z innych form zapożyczeń. W porozumieniu z autorem przetwarzają dzieło za jego zgodą. Tak dzieje się z piosenkami, które wykonują i interpretują różni wykonawcy.

plagiat

Co zrobić, gdy tworzymy projekt i potrzebujemy na przykład ilustracji na dany temat lub podkładu muzycznego do filmu? Najbezpieczniejszym rozwiązaniem jest skorzystanie z treści udostępnionych na licencjach CC (Creative Commons, czytaj: kriejtiw komons). Wszystkie treści na licencji CC są darmowe i nie wymagają pozwolenia na użycie. Jeśli zdecydujesz się skorzystać z takiego dzieła, sprawdź dokładnie zasady, na których będzie to możliwe. Istnieją bowiem różne odmiany tego typu licencji. Na przykład jeśli ktoś chce wykorzystać utwór muzyczny na licencji CC, w napisach końcowych lub w innym miejscu musi wymienić autora muzyki.

licencja CC

W trakcie poszukiwań darmowych utworów możesz spotkać się z pojęciem „domena publiczna”. To zbiór utworów, z których można korzystać bez ograniczeń, na przykład z powodu wygaśnięcia praw majątkowych. W Polsce w tym rozumieniu stosuje się pojęcie „własność publiczna państwa”. Jeśli chcesz wykonać i nagrać utwór powstały w XIX wieku, możesz to zrobić bez opłat.



Gdzie szukać treści na licencji CC? Oczywiście możesz użyć dowolnej wyszukiwarki i dodać do szukanych haseł literę CC lub skorzystać z zaawansowanych opcji wyszukiwania określających prawo do użytkowania. W sieci można znaleźć także wyszukiwarki darmowych treści. Przykładem może być:

<https://search.creativecommons.org>.

Wykonaj ćwiczenie

Jeśli korzystasz z pakietu MS Office, Office 365 lub darmowej chmury OneDrive, masz ułatwione zadanie, ponieważ edytory w tej chmurze mają opcję wyszukiwania treści z domyślnie ustawioną opcją licencji CC.

Uruchom edytor tekstu lub prezentacji. Wyszukaj taką ilustrację na temat równi pochytej, której możesz legalnie użyć w swoim dokumencie (rys. 1.3.)

We wstępie **Wstawianie** edytora Word lub PowerPoint wybierz opcję **Obrazy w trybie online**.

W polu wyszukiwania wpisz frazę określającą cel poszukiwań.

Wyszukiwarka Bing domyślnie ustawi opcję szukania rysunków na licencji CC.

Rys. 1.3. Wyszukiwanie ilustracji na licencji CC z zastosowaniem opcji edytora Word

Uwaga!

Zastosuj zasadę ograniczonego zaufania i sprawdzaj, czy znalezione przez Bing (rys. 1.3.) ilustracje rzeczywiście udostępniono na licencji CC. Zanotuj także autora ilustracji.



ZADANIA DO ROZWIĄZANIA

1. Użyj wyszukiwarki i znajdź strony internetowe o prawie autorskim. Oceń ich wiarygodność i wybierz godne zaufania, na przykład portale prawnicze. Odrzuć fora dyskusyjne, ponieważ każdy uczestnik dyskusji może wpisywać w nich swoje opinie, a nie wszystkie są wiarygodne. Zapoznaj się z treścią artykułów ustawy mówiących o prawie do cytatu i dozwolonym użytku. Określ, czy twoje dotychczasowe działania w sieci są zgodne z ustawą.
2. Za pomocą wyszukiwarki znajdź w sieci ilustracje związane z ostatnio omawianym tematem na lekcjach fizyki. Upewnij się, że wybrana ilustracja jest udostępniona na licencji CC. Wklej ilustrację do dokumentu tekstowego lub prezentacji. Podpisz ją nazwiskiem i imieniem autora.
3. Napisz krótką pracę o konsekwencjach pobierania i używania pirackich kopii programów lub gier komputerowych. W pracy użyj dozwolonych prawem cytatów z wiarygodnych źródeł. Zwróć uwagę na to, by posłużyć się cytatami zgodnie z prawem.
4. Wyszukaj w sieci artykuły prasowe dotyczące plagiatów z różnych dziedzin. Zwróć uwagę na przykład na prace naukowe, filmy, muzykę. Dowiedz się, jakie konsekwencje ponieśli oszuści.
5. Sprawdź, które znaki towarowe zastrzegł producent twojej ulubionej gry komputerowej. Postaraj się odnaleźć je w grze.

PODSUMOWANIE LEKCJI

- Prawo do dozwolonego użytku stanowi, że można legalnie **pobierać treści z internetu**, np. filmy, muzykę, rysunki, na własny użytek. Jeśli jednak robisz to za pośrednictwem sieci takich jak Torrent, pobierając plik, jednocześnie udostępniasz go innym użytkownikom, a to jest już przestępstwem. Każde pobranie, zainstalowanie lub **użycie pirackiej wersji** programu komputerowego, w tym gry, **jest przestępstwem**. s. 8
- **Przestrzeganie prawa autorskiego** jest podstawowym obowiązkiem każdego korzystającego z zasobów sieciowych. Nieznajomość prawa nie chroni przed konsekwencjami jego naruszenia. s. 11
- Podstawowym dokumentem określającym prawa do utworów jest **ustawa o prawie autorskim i prawach pokrewnych**. Możesz używać w swoich pracach fragmentów lub całości niewielkich utworów na zasadzie cytatu. s. 13
- Możesz używać w swoich projektach utworów rozpowszechnianych na **licencji CC** (Creative Commons). Najczęściej spotykanym warunkiem skorzystania z takiego utworu jest umieszczenie w dziele imienia i nazwiska autora. s. 13
- **Plagiat** jest przestępstwem. Nie wolno kopiować fragmentów lub całości prac innych autorów i publikować ich jako swoich. s. 13



2. 8, 16, 32, 64, czyli jak rozwój technologii wpływa na rozwój społeczeństw

NA TEJ LEKCJI:

- dowiesz się, jak postęp technologiczny wpływa na wydajność komputera;
- dowiesz się, do czego można dziś wykorzystywać komputery osobiste.

Technologie zmieniają świat. Zgodzisz się z tym zdaniem? Maszyna parowa wpłynęła na rozwój transportu i produkcji masowej, silnik spalinowy dał możliwość szybkiego przemieszczania się po drogach, ale największym osiągnięciem ludzkości było wykorzystanie prądu elektrycznego. Wyobrażasz sobie życie bez niego? Większość osiągnięć technologicznych współczesnego świata przetwarza energię elektryczną. Na co przetwarza ją komputer? Jak zmienił świat?

2.1. Postęp technologiczny, czyli jak technologie wpływają na wydajność

postęp
technologiczny
a wydajność
komputera

Czas na wyjaśnienie liczb z tytułu rozdziału: 8, 16, 32 i 64 to liczby ściśle związane z rozwojem technologii mikroprocesorowej. Określają one liczbę bitów liczby binarnej, która może być przechowywana w pamięci, przetwarzana i przesyłana magistralą w mikrokomputerze. Załóżmy, że masz dwa komputery. Jeden z nich operuje na liczbach 8-bitowych (tyle, co pierwsze komputery osobiste i konsolki do gier telewizyjnych), a drugi na 64-bitowych (współczesny komputer osobisty). Umiesz wyznaczyć zakres liczb przetwarzanych w obu maszynach?

Przyda ci się powtórka ze szkoły podstawowej, czyli wiedza o dwójkowym zapisie liczb. Czy pamiętasz, że dziesiętną wartość kolejnych bitów określa kolejna potęga dwójki? To oznacza, że wagi bitów w 8-bitowej liczbie zapisanej w NKB (Naturalnym Kodzie Binarnym) to: 128, 64, 32, 16, 8, 4, 2, 1. Dlaczego w kolejności malejącej? A w jakiej zapisujesz pozycje liczby dziesiętnej? Waga 1 to pierwsza cyfra z prawej, a 10 – to druga z prawej. Tu jest tak samo. Czym jest waga pozycji cyfry w liczbie zarówno w systemie dwójkowym, jak i dziesiętnym? Przypomni ci to rysunek 2.1.



Liczba zapisana
w naturalnym kodzie binarnym

$$\begin{array}{cccccccc}
 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
 \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 &
 \end{array}
 \quad
 \begin{array}{l}
 \text{Wagi dziesiętne} \\
 \text{pozycji w liczbie binarnej}
 \end{array}$$

$$10110011_{(b)} = 1 \cdot 128 + 0 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 0 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 179_{(10)}$$

Rys. 2.1. Wagi dziesiętne pozycji liczby binarnej

Wróćmy do naszego problemu. Na rysunku 2.1. przedstawiono liczbę ośmiobitową. Wiadomo, że osiągnęłaby ona największą wartość dla ośmiu jedynek. Jak obliczyć wartość dziesiętną takiej liczby? Oczywiście sumując wagi dziesiętne wszystkich pozycji. Da to liczbę $255_{(10)}$. Z tego wniosek, że procesor 8-bitowy potrafił przetwarzać liczby z zakresu od 0 do $255_{(10)}$. Jak obliczyć ilość liczb możliwych do zapisania w postaci 64-bitowej? Sumowanie wag byłoby czasochłonne. Spróbuj opracować prosty sposób na obliczanie maksymalnej wartości dziesiętnej liczby zapisanej z użyciem dowolnej liczby bitów. Pomoże ci w tym rysunek 2.2.

$$1111\ 1111_{(b)} = 1 \cdot 128 + 1 \cdot 64 + 1 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 255_{(10)}$$

$$1111\ 1111_{(b)} + 1_{(b)} = 1\ 0000\ 0000_{(b)} = 1 \cdot 256_{(10)} = 2^8_{(10)}$$

Wniosek:

$$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 = 2^8 - 1$$

Rys. 2.2. Podpowiedź do rozwiązania problemu obliczania maksymalnej wartości dziesiętnej liczby binarnej

Jeśli spróbowałeś/spróbowałaś rozwiązać problem, to dostrzegasz, że dodanie 1 do maksymalnej liczby (wszystkie pozycje mają wartość 1) zapisanej za pomocą n bitów spowoduje zwiększenie liczby bitów do $n+1$. Pozycja ta przybierze wartość 1, a pozostałe 0. Szukany wzór powinien więc mieć postać:

$$L_{\max} = 2^{n+1} - 1$$

Maksymalna wartość dziesiętna liczby binarnej zapisana za pomocą n bitów

Zgodnie ze wzorem maksymalna liczba dziesiętna zapisana za pomocą 64 bitów to:

$$2^{64} - 1 = 18446744073709551616 - 1 = 18446744073709551615$$



Zwiększenie liczby bitów z 8 do 64 (8-krotne) w słowie przetwarzanym przez procesor lub przesyłanym magistralami zwiększyło aż 72 340 172 838 076 673-krotnie liczbę możliwych do przetwarzania liczb. Dotyczy to także liczby przesyłanych informacji za pośrednictwem magistral.

Czy teraz doceniasz postęp, jaki nastąpił w technologii informatycznej? Jak bardzo wzrosła wydajność komputerów?

Postęp w budowie mikroprocesorów i innych układów cyfrowych polega nie tylko na zwiększaniu liczby przetwarzanych bitów, lecz także na unowocześnianiu technologii ich budowy.

Poniżej wymieniono kilka cech postępu technologicznego mających wpływ na funkcjonalność komputerów.

Tab. 2.1. Cechy postępu technologicznego w dziedzinie informatyki

Cecha	Znaczenie
Częstotliwość pracy	W pierwszych komputerach domowych procesory taktowano z częstotliwością od 1 MHz (np. Comodore 64). Współcześnie może ona przekraczać 4 GHz, czyli około 4000 razy szybciej. Znacząco wpływa to na wydajność komputerów i możliwość współpracy z zaawansowanymi systemami operacyjnymi.
Pobór energii i zasilanie	Zmiana technologii wytwarzania układów scalonych znacznie zmniejszyła zapotrzebowanie na energię. Obecnie komputery przenośne mogą pracować nawet kilkanaście godzin zasilane wyłącznie z akumulatora. Świadczy to również o dużym postępie w dziedzinie źródeł zasilania.
Stopień scalenia	Pierwszy procesor firmy Intel potrzebował kilkunastu układów towarzyszących. Obecnie nawet cały komputer można zbudować w jednym układzie scalonym. Dzięki miniaturyzacji i pozostałym cechom można np. budować coraz to wydajniejsze telefony komórkowe.
Architektura układu	Inżynierowie pracują także nad nowymi rozwiązaniami technicznymi. Poszczególne bloki procesorów i innych układów różnią się od poprzedników nie tylko wydajnością, lecz także budową i zastosowanymi rozwiązaniami technicznymi. Dzięki temu można na przykład budować układy, które przy tej samej częstotliwości taktowania, co ich poprzednicy, pracują znacznie wydajniej.
Pojemność pamięci RAM	Współczesne systemy operacyjne, zarówno w PC, jak i telefonach, wymagają coraz większej pojemności pamięci operacyjnej. Pierwszym domowym komputerem starczyło 48 kB. Obecnie standardem jest kilka lub kilkanaście GB. Pozwala to także na instalowanie nowych wersji programów i aplikacji, które przy takiej ilości pamięci pracują płynnie i pewnie.



2.2. Zaczęło się od rozrywki, czyli do czego wykorzystujemy komputery osobiste

W filmie *Piraci z Krzemowej Doliny* umieszczono scenę, w której młody Steve Wozniak, bliski przyjaciel i wspólnik Steve'a Jobsa, przedstawił prototyp komputera Apple specjalście z HP. Ten po chwili zastanowienia zadał pytanie: „A do czego zwyktemu człowiekowi komputer?”. Jobs miał podpisany kontrakt i wszystko, co wymyślił, miało należeć do firmy. Jak wiemy, HP nie widziało zastosowania dla komputera osobistego. Ale pytanie zostaje otwarte. Nikt wtedy nie był w stanie na nie odpowiedzieć. Obecnie możemy jedynie powiedzieć, czym jest komputer, ale nikt nie zdoła określić, do czego będzie wykorzystywany w przyszłości.

Czy umiesz wymienić choćby kilka zastosowań współczesnych komputerów w kontekście rozwoju naszej cywilizacji? Można dzielić je na wiele sposobów. Z jednej strony postęp informatyki znacznie przyspiesza rozwój wszelkich dziedzin nauki, pomaga w życiu codziennym, wspomaga leczenie, edukację, komunikację itp., a z drugiej pozwala na budowanie coraz to groźniejszej broni, używanie technologii do przestępstw czy wykorzystanie jej do przekazywania fałszywych informacji.

Komputery przyczyniły się ponadto do zmian cywilizacyjnych i rozwoju społeczeństw. Nie było chyba nigdy w historii świata wynalazku, który nie tylko zmienił sposoby produkcji czy komunikacji, lecz także znacząco wpłynął na zmiany zachowań społecznych.

Obecnie każdy posiadacz telefonu komórkowego może mieć dostęp do sieci przez całą dobę. Jeśli do tego korzysta z komunikatorów i portali społecznościowych, to może nie tylko utrzymywać ciągły kontakt ze znajomymi, lecz także na bieżąco śledzić aktualności oraz wypowiedzi i opinie artystów, polityków itp. Z jednej strony pozwala to na większą aktywność i świadomość obywatelską, a z drugiej – uzależnia od informacji. Bez względu na wiek i zaawansowanie w posługiwaniu się środkami IT ludzie uzależniają się od ciągłego uczestniczenia w życiu wirtualnym. Nie ma chyba użytkownika sieci, który kilka razy dziennie nie czytałby bieżących informacji, wypowiedzi na portalach społecznościowych czy korespondencji.

Powszechny dostęp do internetu zmienił także oblicze polityki. Dawniej, jeśli polityk chciał przekazać coś swoim wyborcom, musiał zgłosić się do mediów. Obecnie większość polityków używa na przykład Twittera. Publikowane tam wypowiedzi docierają do wielu milionów ludzi w tym samym czasie. Polityk obecnie może pominąć oficjalne media, chociażby stacje TV, i nie musi już zabiegać o ich przychylność. Rola mediów ogranicza się więc do komentowania wypowiedzi. Zanim jednak to się stanie, swoje opinie zdążą już wypowiedzieć użytkownicy danej usługi. Wszystko dzięki temu, że większość użytkowników telefonów jest on-line przez całą dobę.



Oczywiście od sieci nie da się uciec. Współczesny świat rozwija się głównie dzięki szybkiej i nieograniczonej możliwości wymiany informacji. Problem w tym, by robić to świadomie i rozsądnie.



Komunikacja

Telefon – dawniej stacjonarny,
dziś mobilny globalnie.

Wymiana dokumentów natychmiastowa
bez względu na miejsce nadania i odbioru.

Komunikatory wykorzystujące transmisje
strumieniowe dźwięku i obrazu.
Komunikacja bezpośrednia bez ograniczeń.

E-mail, bez ograniczeń czasowych.

Wymiana nieograniczonej ilości danych.

Profesjonalne i amatorskie transmisje
strumieniowe na żywo. Dawniej dostępne
jedynie dla stacji telewizyjnych.



Życie
codzienne

Telefon – on-line przez całą dobę.
Nie trzeba już łącza i komputera PC.

Szybkie usługi i zakupy przez sieć.
W tym także zakup biletów komunikacji,
na koncerty itp.

Bankowość i płatność on-line,
w tym także za pomocą telefonu.

Informacje i wiadomości dostępne
natychmiast po opublikowaniu.
W tym także fałszywe, które
powinniśmy umieć ocenić.

E-urzędy. Obywatel może
kontaktować się (składać i otrzymywać
dokumenty itp.) z instytucjami państwa,
administracji, samorządu lokalnego
i innych instytucji za pośrednictwem sieci.

E-learning, czyli nauka na odległość
za pomocą środków elektronicznych.
Część zajęć na uczelniach i w szkołach
odbywa się z wykorzystaniem platformy
e-learningowej. Kursy doskonalące oraz
inne szkolenia także są dostępne w sieci.

e-urzędy
i e-learning

ewolucja
komputerów
a zmiana
zachowań
społecznych

Rys. 2.3. Niektóre z zastosowań IT mających wpływ na zmiany cywilizacyjne



ZADANIA DO ROZWIĄZANIA

1. Zapisuj lub archiwizuj w dowolny sposób czas używania swojego telefonu. Przy czym nie rozróżniaj wykonywanych czynności ani okoliczności. Załóż, że twój aktywny dzień ma 12 godzin. Oblicz, ile procent tego czasu patrzyłeś/patrzyłaś na ekran telefonu. To dużo czy mało? Oceń wynik doświadczenia i uzasadnij swoją ocenę.
2. Od 12 godzin odejmij czas spędzony na lekcjach w szkole (nie licząc przerw) i popraw obliczenia z zadania 1. Dokonaj ponownej analizy i oceń, który z przypadków daje więcej informacji na temat twojej aktywności w sieci. Wyciągnij wnioski. Znasz siebie i wiesz, na co wykorzystaleś/wykorzystałaś ten czas. Jaka część czasu spędzonego z telefonem miała niewielkie znaczenie dla twojego życia i rozwoju? Jaką część tego czasu mogłeś/mogłaś poświęcić na odpoczynek, relaks na świeżym powietrzu, naukę itp.? Co wartościowego można by w tym czasie zrobić?
3. Na podstawie wyników z zadania 2. określ procentowo ilość czasu spędzonego przed ekranem telefonu, która była niezbędna (np. poświęcona na rozmowy z bliskimi, robienie zdjęć, dzielenie się ważnymi informacjami itp.).
4. Wyświetl stronę internetową urzędu miasta, gminy, powiatu, sołectwa z terenu, na którym mieszkasz. Sprawdź, czy oferuje ona usługi e-urzędu. Jeśli tak, dowiedz się, do czego służą i jak ułatwiają życie mieszkańcom.
5. Za pomocą kalkulatora oblicz, ile różnych liczb można przesłać magistralą liczącą 128 bitów. Porównaj wynik z magistralą 64-bitową. Znajdź w sieci przykłady urządzeń komputerowych, które przetwarzają słowa 128-bitowe. Sprawdź, czy dostępne są 256-bitowe.

PODSUMOWANIE LEKCJI

- **Rozwój komputerów** osobistych rozpoczął nowy okres w rozwoju cywilizacji. 8-krotne zwiększenie liczby bitów (z 8 do 64) przesyłanych lub przetwarzanych jednocześnie przez procesor znacznie zwiększyło liczbę jednocześnie przesyłanych lub przetwarzanych informacji – liczb (72 340 172 838 076 673-krotnie). Ma to wielki wpływ na **wydajność komputerów**. s. 16
- **E-learning** to nauczanie lub uczenie się na odległość za pośrednictwem środków elektroniczno-informatycznych. Platformy e-learningowe wykorzystuje się na wielu płaszczyznach – od wspomagania nauczania w szkołach, przez kursy doskonalące i nauczanie akademickie. Również sprawy urzędowe można załatwiać **za pośrednictwem usług elektronicznych**. s. 20
- Ewolucja komputerów osobistych zmieniła także zachowania społeczne. Każdy może publicznie wypowiadać swoje poglądy i ogłaszać informacje z pominięciem tradycyjnych mediów, np. **za pośrednictwem Twittera**. s. 20



3. Kim jestem, czyli jak bezpiecznie budować wizerunek w sieci

NA TEJ LEKCJI:

- dowiesz się, kto i co chroni twoje dane;
- dowiesz się, jak dbać o swój wizerunek w sieci.

Internet jest źródłem olbrzymiej ilości informacji. Dostęp do nich jest praktycznie nieograniczony. Czy jednak masowy dostęp do wszelkich informacji nie zagraża twojej prywatności? Czy zdjęcia z dyskoteki, na których widać, jak się bawisz, umieszczone na portalach społecznościowych bez twojej wiedzy są takim naruszeniem? Czy zastanawiasz się przed umieszczeniem w internecie różnych fotografii, filmów i tekstów, że mogą one komuś sprawić przykrość, spowodować różne kłopoty? Gdzie leży granica prywatności? Czy można dysponować czymś wizerunkiem bez jego wiedzy? Czy instytucje mogą żądać od ciebie danych osobowych? Jakie skutki może mieć fakt, że informacjami o tobie dysponują firmy, banki, urzędy i osoby prywatne?

3.1. Kto i co chroni twoje dane, czyli co mówi prawo

Problem prywatności i bezpieczeństwa danych osobowych istnieje tak długo, jak istnieją środki przekazu informacji. Dlatego 19 grudnia 1966 roku ONZ ratyfikowała Międzynarodowy Pakt Praw Obywatelskich i Politycznych, a Polska ratyfikowała go w 1977 roku. Artykuł 17 brzmi:

1. Nikt nie może być narażony na samowolną lub bezprawną ingerencję w jego życie prywatne, rodzinne, dom czy korespondencję ani też na bezprawne zamachy na jego cześć i imię.
2. Każdy ma prawo do ochrony prawnej przed tego rodzaju ingerencjami i zamachami.

**naruszenie
prywatności w sieci**

Masz więc prawo do ochrony swojej prywatności i domagania się usunięcia informacji ją naruszających. Do niedawna proces dochodzenia swoich praw był długotrwały i mało skuteczny z powodu braku odpowiednich przepisów prawa. Parlament Europejski opracował rozporządzenie w sprawie ochrony osób fizycznych w związku z przetwarzaniem danych osobowych i w sprawie swobodnego przepływu takich danych oraz uchylenia dyrektywy 95/46/WE (2016/679 z 27 kwietnia 2016 r.), które wprowadzono 25 maja 2018 roku we wszystkich krajach członkowskich UE. Obywatele Unii są więc chronieni tym prawem na jej terenie. Rozporządzenie w skrócie określa się jako RODO.



Czym właściwie są dane osobowe? Domyślasz się, że są nimi imię i nazwisko, PESEL, prywatny lub firmowy numer telefonu, adres poczty elektronicznej, także dane identyfikujące twój komputer, na przykład adres IP lub MAC oraz inne identyfikatory elektroniczne.

Bez wiedzy osoby, której dotyczą dane, nie wolno ich przetwarzać ani automatycznie, ani przez działalność człowieka. A czym jest przetwarzanie? To każda operacja na tych danych, przykładowo: zbieranie, udostępnianie, utrwalanie, porządkowanie, adoptowanie, modyfikacja, przeglądanie, łączenie, usuwanie, niszczenie, używanie między innymi do wysyłki korespondencji czy towarów.

Uwaga!

Każdy, kto nie stosuje przepisów RODO, podlega wysokiej karze. Każdy obywatel Unii Europejskiej ma prawo do przeglądania danych, które zostały do tej pory zgromadzone, np. w sklepie wysyłkowym. Jeśli uzna, że nie są one już potrzebne danemu podmiotowi lub osobie, można kategorycznie żądać ich usunięcia, a operator danych musi je usunąć.

RODO
DLA OBYWATELA

RODO jest po to, żeby lepiej chronić Twoją prywatność. Jako obywatel - właściciel swoich danych - zyskasz nowe prawa. Jako pracownik - będziesz miał nowe obowiązki, po to, by lepiej chronić dane innych.

MASZ WIELE PRAW - PAMIĘTAJ O NICH I KORZYSTAJ Z NICH!

- Będziesz mógł zażądać usunięcia swoich danych, szczególnie gdy nigdy ich nie podawałeś (to mogą być np. telefony od marketerów).
- Możesz zażądać przeniesienia swoich danych osobowych tam, gdzie chcesz. Na przykład z Banku A do Banku B, po to, by na nowo nie wypełniać całej dokumentacji kredytowej.
- Będziesz mógł uzyskać kopie danych. Masz pełne prawo wglądu w swoje dane i dysponowania nimi wedle swojego uznania.
- Zostaniesz poinformowany, jeśli dojdzie do wycieku danych, który zagraża Twojej prywatności.
- Zmniejszą się procedury handlu detalicznego - powinieliś to odczuć np. przez mniej telefonów z ofertami zakupu różnych towarów. A jeśli Twoje dane trafią do nowej bazy - oświadcz się o tym.

Państwo obszernie informuje o zasadach i przepisach RODO pod adresem: <https://www.gov.pl/cyfryzacja/rodo-informator>. Znajdziesz tam szczegółowe informacje, a także ulotki i filmy przybliżające obywatelowi (rys. 3.1.) i przedsiębiorcy jego prawa i obowiązki w związku z wprowadzeniem RODO. Wczytaj tę stronę i zapoznaj się z tymi materiałami. To jedno z najważniejszych dla każdego obywatela praw. Twoim obowiązkiem jest zapoznanie się z nim, by uniknąć kłopotów w związku z obrotem danymi osobowymi.

Rys. 3.1. Ulotka informacyjna Ministerstwa Cyfryzacji dotycząca wprowadzenia RODO

RODO a usuwanie informacji z bazy danych



dbanie o wizerunek
w sieci

3.2. Dbaj o swój wizerunek, czyli nie ponad konieczność

Przypadki umieszczania w sieci fotografii lub filmów bez zgody osób na nich uwiecznionych są częste. Osoby publikujące bezprawnie takie treści, wśród których znaczną grupę stanowią ludzie młodzi, robią to dla żartu lub z chęci ośmieszenia kogoś. Nie zdają sobie sprawy z krzywdy, jaką mogą wyrządzić danej osobie i jej bliskim. Zdarza się, że ofiara takiego działania wykazała się zbyt wielkim zaufaniem, a nawet naiwnością, w stosunku do sprawcy, na przykład sama przesała mu swoje zdjęcia. Upublicznienie materiałów nierzadko ma tragiczne skutki – może prowadzić do depresji, a nawet samobójstw.

Udostępnianie zdjęć lub nagrań bez zgody osób na nich uwiecznionych nie tylko jest niezgodne z prawem, RODO i paktem ONZ, lecz także jest sprzeczne z normami moralnymi. To przestępstwo, za które sprawca może odpowiadać przed sądem.

Jednak każdy człowiek powinien zdawać sobie sprawę, że zdjęcia i filmy, na których został uwieczniony, mogą zostać opublikowane. Pamiętaj o tym i dbaj o swój wizerunek. Nie pozwalaj się fotografować nieznanym osobom. Jeśli już to nastąpi, wyraźnie zastrzeż swoje prawa lub zażądaj skasowania materiału.

Twoje prawa do zdjęć i filmów są także chronione przez ustawę o prawie autorskim i prawach pokrewnych. W artykule 81 mowa:

1. Rozpowszechnianie wizerunku wymaga zezwolenia osoby na nim przedstawionej. W braku wyraźnego zastrzeżenia zezwolenie nie jest wymagane, jeżeli osoba ta otrzymała umówioną zapłatę za pozowanie.

Jednocześnie ta sama ustawa stwierdza, że:

Art. 83. Do roszczeń w przypadku rozpowszechniania wizerunku osoby na nim przedstawionej oraz rozpowszechniania korespondencji bez wymaganego zezwolenia osoby, do której została skierowana, stosuje się odpowiednio przepis art. 78 ust. 1; roszczeń tych nie można dochodzić po upływie dwudziestu lat od śmierci tych osób.

Osoba, która umieściła zdjęcie lub film, nie jest ścigana przez prawo automatycznie, a jedynie po zgłoszeniu tego faktu (możliwości popełnienia przestępstwa) przez osobę poszkodowaną.



Jakie zdjęcia i filmy zawierające wizerunek obcej osoby można pokazywać bez konsekwencji? Określono to w artykule 81 powyższej ustawy:

2. Zezwolenia nie wymaga rozpowszechnianie wizerunku:

- 1) osoby powszechnie znanej, jeżeli wizerunek wykonano w związku z pełnieniem przez nią funkcji publicznych, w szczególności politycznych, społecznych, zawodowych,
- 2) osoby stanowiącej jedynie szczegół całości takiej jak zgromadzenie, krajozraz, publiczna impreza.

Artykuł 83 powyższej ustawy dotyczy również ochrony korespondencji elektronicznej. Nie wolno czytać lub rozpowszechniać korespondencji bez zgody autora lub adresata. Dotyczy to także przypadku, w którym nie następuje włamanie do skrzynki pocztowej. Mimo tak sformułowanego prawa należy dbać o bezpieczeństwo haseł do skrzynek i innych kont internetowych.

Większość osób korzystających z internetu nie podejrzewa, że może paść ofiarą przestępców. Często są łatwowierni i opowiadają nieznanym o sobie lub upowszechniają w portalach społecznościowych informacje o tym, gdzie mieszkają, jaki mają samochód, ile zarabiają itp. Zdarza się, że takie dane ich rozmówca skrzętnie gromadzi. Jeśli na przykład umieścisz w portalu zdjęcie swojego nowego domu i podasz adres lub inne dane, które umożliwią jego ustalenie, oraz napiszesz, że właśnie wybierasz się na 2 tygodnie za granicę, to jest to doskonała wiadomość dla złodzieja. Pamiętaj, że nie tylko twoi znajomi mają dostęp do informacji które opublikowałeś/opublikowałaś.

W portalach społecznościowych nie ma mechanizmu sprawdzania tożsamości. Każdy może w nich podszyć się pod ciebie. Takie zachowanie to kradzież tożsamości. Szkodzi ono zarówno tobie, jak i twojej rodzinie oraz znajomym.

Wiele usług internetowych wymaga rejestracji. Jeśli wymaga to podania danych osobistych, na przykład twojego imienia i nazwiska z adresem i numerem telefonu, to poważnie zastanów się, czy koniecznie musisz korzystać z tych usług. Jeśli wymagany jest tylko mail, warto utworzyć skrzynkę pocztową na jednym z darmowych serwerów i używać jej do takich rejestracji. Zapisane podczas rejestracji dane trafiają do bazy. Często ignoruje się mały znaczek stawiany domyślnie przy formule, że zgadzasz się na przesyłanie między innymi informacji handlowych, a w przypadku użycia roboczej skrzynki pocztowej będzie to miało mniejsze znaczenie.

Zapoznaj się ze schematami dotyczącymi prywatności w sieci (rys. 3.2. i 3.3. na następnej stronie).

internetowe
formularze a dane
osobiste



Rys. 3.2. Działania zwiększające bezpieczeństwo wizerunku w sieci



Rys. 3.3. Jak nie naruszać prawa do prywatności



ZADANIA DO ROZWIĄZANIA

1. Przejrzyj dokładnie swoje konta na portalach społecznościowych i usuń z nich wszystkie informacje, które mogą być niewłaściwie wykorzystane przez innych użytkowników, np. data urodzenia, adresy, nieudane zdjęcia, zdjęcia znajomych, na których publikację nie wyrazili zgody.
2. Przejrzyj zawartość portali internetowych, w szczególności społecznościowych, oraz strony internetowe, aby sprawdzić, czy ktoś narusza twoje prawa do wizerunku i prywatności. Skorzystaj przy tym z wyszukiwarek. Zareaguj odpowiednio, jeśli stwierdzisz nieprawidłowości.
3. Wykonaj prezentację, w której przedstawisz przykłady łamania prawa do prywatności. Dane zaczerpnij z portali informacyjnych. Omów, jak można było uniknąć tych nadużyć lub zminimalizować ich skutki.
4. Wyszukaj w sieci dokumenty prawa mówiące o ochronie praw człowieka do prywatności oraz komentarze prawników do nich. Znajdź także artykuły i opracowania na temat interpretacji tych praw. Przeczytaj je i wyciągnij wnioski dotyczące zasad zachowania się w sieci.
5. Odwiedź stronę <https://www.gov.pl/cyfryzacja/rodo-informator> i przeczytaj informacje na niej zawarte. Zapoznaj się także z dokumentami, które można pobrać z tej strony. Na ich podstawie przeanalizuj twoje nowe prawa i zrób listę portali i instytucji, z których powinieneś/powinnaś wycofać swoje dane.

PODSUMOWANIE LEKCJI

- W przypadku odkrycia rozmyślnego, złośliwego lub drastycznego **naruszenia twojej prywatności** natychmiast reaguj – powiadom bliskie osoby i policję. Nie daj się szantażować. s. 22
- Na podstawie RODO **możesz żądać usunięcia swoich danych z każdej bazy danych**. Wyjątek stanowią te, które są niezbędne do realizacji wcześniej podpisanej umowy, np. na usługi bankowe, teleinformatyczne. Jednak i te instytucje muszą ujawnić, jakie dane na twój temat zgromadziły. s. 23
- Dbaj o swój wizerunek poprzez **rozsądne umieszczanie materiałów** na własny temat oraz danych potrzebnych do uruchomienia niektórych usług. Dzięki temu znacznie zwiększysz bezpieczeństwo twoje i twoich danych. s. 24
- **Nie wypełniaj internetowych formularzy danymi osobistymi**, jeśli nie jest to absolutnie konieczne. Nie udostępniaj swoich danych osobom oferującym pracę w internecie. Prawo do takich danych mają jedynie pracodawcy, z którymi podpisujesz umowę o pracę. Dotyczy to także pracy sezonowej, np. w wakacje. s. 25



PODSUMOWANIE PRZED SPRAWDZIANEM

1. Bądź uczciwy, czyli przestrzeganie prawa w świecie informatyki

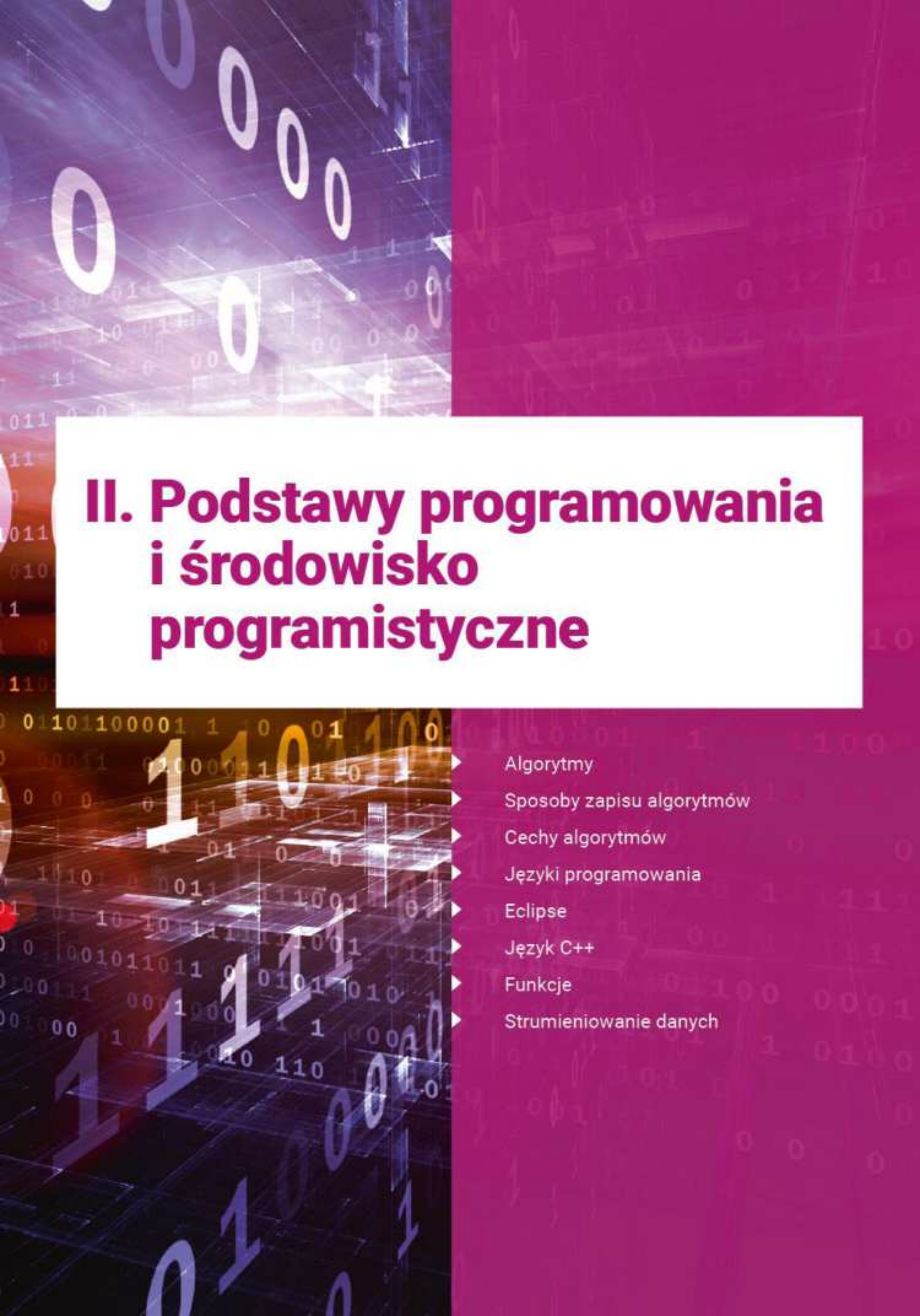
- s. 11 • **Prawa majątkowe** to nie **prawa autorskie**. Naruszenie obu jest przestępstwem przeciw własności intelektualnej.
- s. 11 • **Ochroną może być objęty wyłącznie** sposób wyrażenia; **nie są objęte ochroną** odkrycia, idee, procedury, metody i zasady działania oraz koncepcje matematyczne.
- s. 11 • **Utwór jest przedmiotem prawa autorskiego** od chwili ustalenia, chociażby miał pozostać nieukończony.
- s. 11 • **Ochrona przysługuje twórcy** niezależnie od spełnienia jakichkolwiek formalności.
- s. 13 • **Licencja CC** (Creative Commons) pozwala na używanie i rozpowszechnianie utworu bez zezwolenia. Należy jednak podać autora utworu, np. w napisach końcowych filmu, w którym użyto muzyki udostępnionej na licencji CC.

2. 8, 16, 32, 64, czyli jak rozwój technologii wpływa na rozwój społeczeństw

- s. 16 • **Binarny system liczbowy** jest podstawą konstrukcji i działania urządzeń cyfrowych, w tym także komputerów.
- s. 19 • **Powszechny dostęp** do kanałów informacyjnych zmienił sposób postrzegania świata, życia społeczeństw.
- s. 19 • Dzięki rozwojowi systemów komputerowych **każdy ma możliwość głoszenia swoich poglądów**.
- s. 19 • Poprzez **ogólnodostępną** telefonię komórkową i internet całkowicie zmieniły się sposoby komunikowania się, pracy i rozrywki społeczeństwa. Rozwój cywilizacji przyspieszył.

3. Kim jestem, czyli jak bezpiecznie budować wizerunek w sieci

- s. 22 • **Masz prawo do ochrony swojej prywatności** i domagania się usunięcia informacji ją naruszających.
- s. 23 • **Obywatele Unii Europejskiej i ich dane osobowe są chronieni** prawem na jej terenie. Rozporządzenie **RODO** opracował Parlament Europejski i Rada Europy. Zgodnie z RODO możesz żądać usunięcia z baz danych informacji o sobie.
- s. 24 • **Rozpowszechnianie wizerunku** wymaga zezwolenia osoby na nim przedstawionej. Dbanie o własny wizerunek **jest obowiązkiem** każdego obywatela.
- s. 25 • **Nigdy nie** udostępniaj lekkomyślnie swoich danych osobowych.



II. Podstawy programowania i środowisko programistyczne

- ▶ Algorytmy
- ▶ Sposoby zapisu algorytmów
- ▶ Cechy algorytmów
- ▶ Języki programowania
- ▶ Eclipse
- ▶ Język C++
- ▶ Funkcje
- ▶ Strumieniowanie danych

4. Przypomnij sobie, czyli podstawy tworzenia algorytmów

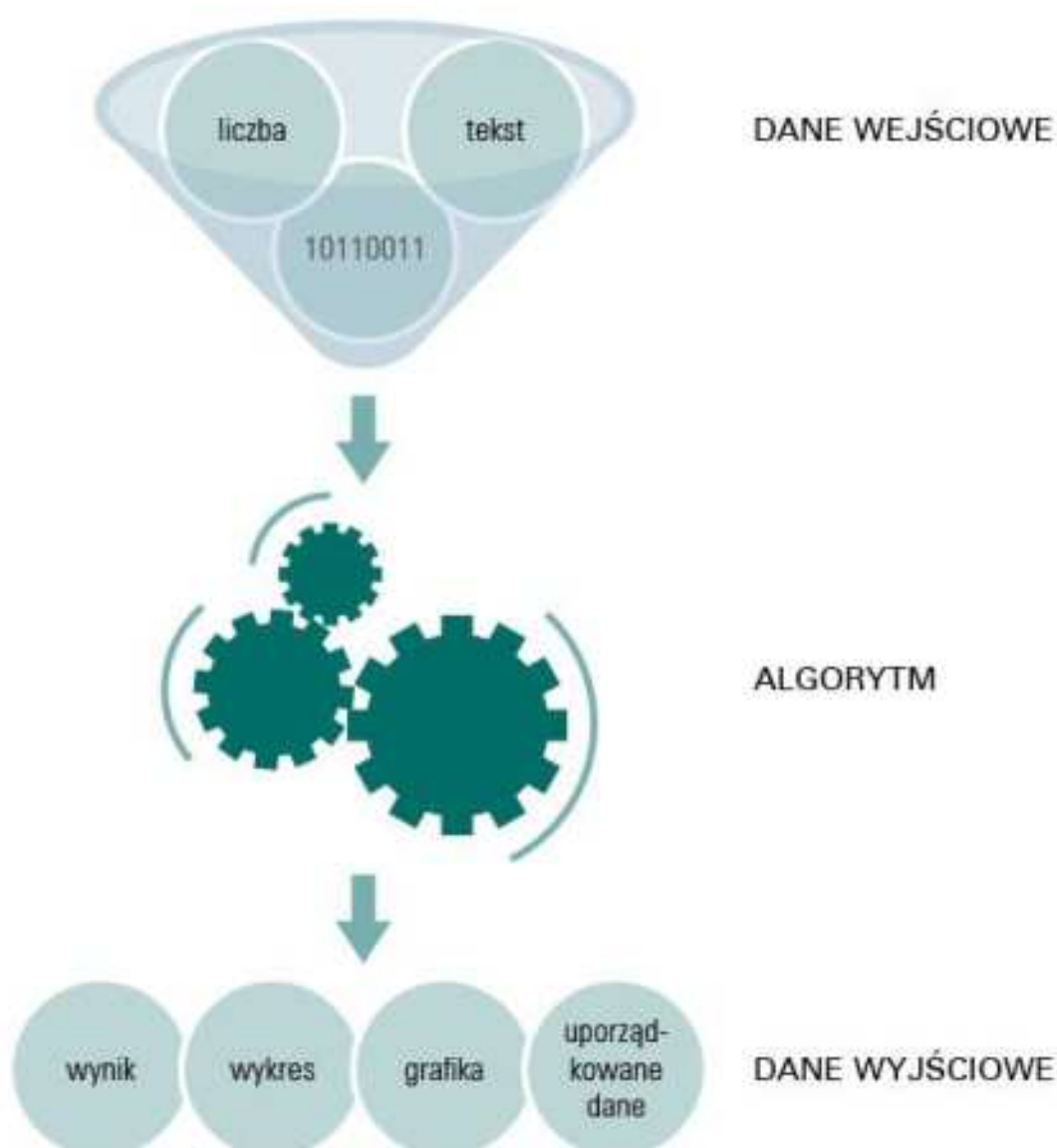
NA TEJ LEKCJI:

- poznasz sposoby zapisu algorytmów;
- nauczysz się specyfikacji algorytmu.

W trakcie nauki informatyki niejednokrotnie spotkałeś/spotkałaś się z algorytmami. Sposoby ich zapisu, interpretacji i edycji nie powinny być ci obce. Jest to na tyle ważne, że warto powtórzyć i przypomnieć sobie najważniejsze zagadnienia z tej dziedziny.

4.1. Prozą i graficznie, czyli różne sposoby zapisu algorytmów

Czy pamiętasz z poprzednich lat nauki definicję algorytmu? Jeśli nie, poniższy rysunek ci ją przypomni.



Rys. 4.1. Mechanizm przetwarzania danych za pomocą algorytmu

algorytm –
przetwarzanie
danych

Algorytm to prowadzący do przetworzenia danych wejściowych na wyjściowe skończony i uporządkowany opis precyzyjnie sformułowanych czynności.

Dane wejściowe nie muszą się jednoznacznie kojarzyć z danymi komputerowymi. Algorytmy mogą opisywać różne procesy. Jeśli nie wiesz, jak wykonać jakąś czynność, na przykład za pomocą edytora grafiki, sięgasz po instrukcje lub tutoriale. Znajdziesz ich wiele, chociażby w serwisie YouTube. Te instrukcje także można nazwać algorytmami. Większość doprowadzi cię do celu, jednak nie wszystkie są czytelne i zrozumiałe. Ze sposobami zapisu algorytmów jest podobnie – ich dobór zależy od specyfiki opisywanego problemu.

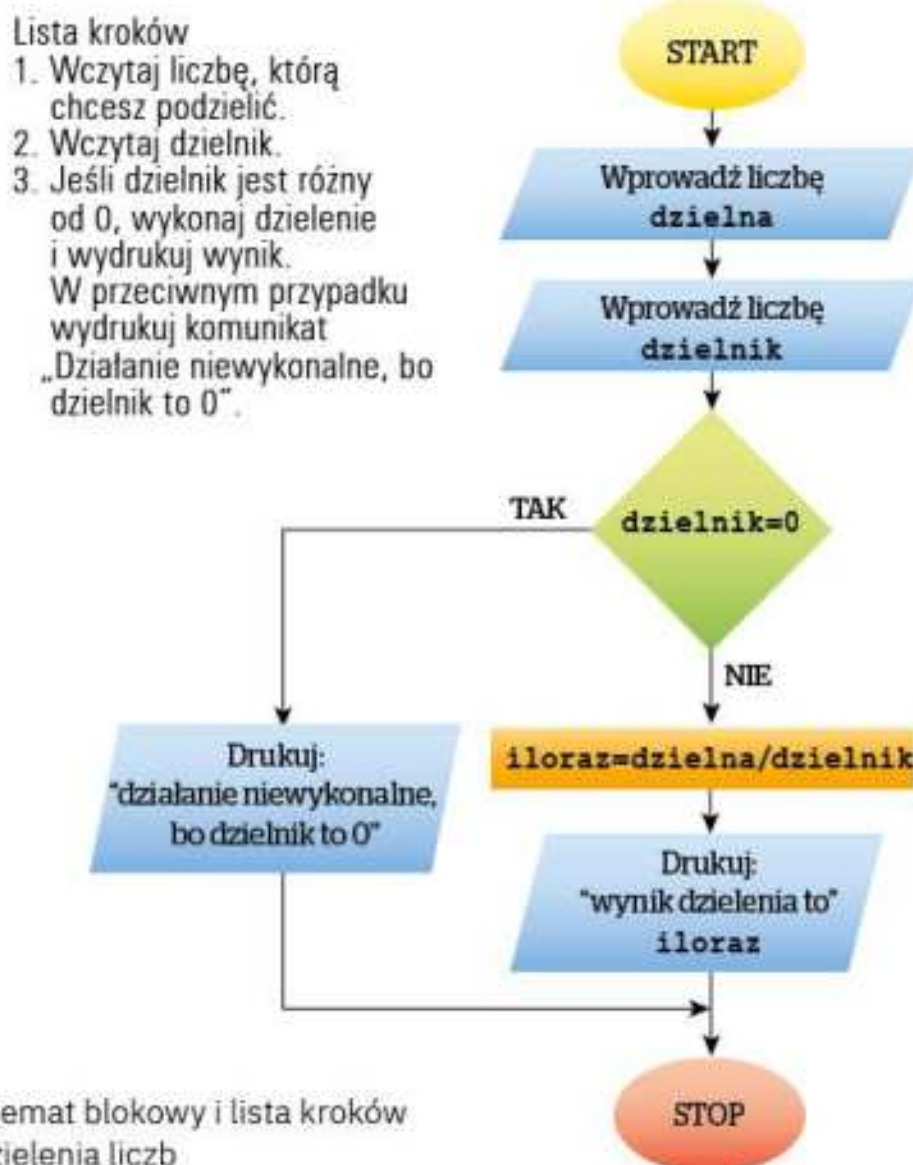
Najprostszym sposobem opisu rozwiązania problemu jest **zapis słowny**. Autor po prostu zdanie po zdaniu opisuje kolejne czynności.

Precyzyjny opis powinien zawierać tak zwaną **listę kroków**. Ten sposób sprawdza się w nieskomplikowanych przypadkach. Ma jednak kilka wad. Po pierwsze, lista sporządzona w języku polskim raczej nie będzie czytelna dla programisty z Chin. Po drugie, opis komplikuje się, gdy w algorytmie występują pętle i warunki. Wtedy odczytujący go programista może łatwo się pogubić.

lista kroków

Wiesz już, że innym, bardzo popularnym sposobem zapisu algorytmu jest **schemat blokowy**. Przedstawia on w sposób graficzny kolejne czynności połączone strzałkami, co pozwala na tworzenie dowolnie skomplikowanych schematów

schemat blokowy



Rys. 4.2. Schemat blokowy i lista kroków algorytmu dzielenia liczb

i łatwą ich weryfikację oraz testowanie. Jeśli w poszczególnych blokach znajdują się ogólnie stosowane znaki, jak wzory i symbole matematyczne, algorytm będzie czytelny i zrozumiały. Przykład prostego algorytmu dzielenia zapisanego w postaci schematu blokowego i listy kroków znajdziesz na rysunku 4.2. Analizując zapis graficzny, przypomnij sobie znaczenie poszczególnych bloków – **wejścia–wyjścia, decyzyjnego i operacyjnego**.

pseudokod

Istnieje jeszcze jeden sposób zapisu algorytmu. Używają go głównie programiści, ponieważ zachowuje on strukturę podobną do języka programowania. Przy czym nie zachowuje reguł składniowych konkretnego języka. Dzięki temu na jego podstawie łatwiej tworzyć programy komputerowe w różnych językach. Są jednak w nim zawarte konkretne sugestie rozwiązania problemu. Programista może więc skupić się jedynie na tworzeniu kodu w konkretnym języku programowania.

Poniższy przykład dotyczy działania hotelowego zamka do drzwi z czytnikiem kart zbliżeniowych.

```
Jeżeli wykryłeś kartę to
    jeżeli karta prawidłowa to
        otwórz drzwi;
    w przeciwnym przypadku
        zapal czerwoną kontrolkę;
```

Tak zapisany algorytm można w prosty sposób zaimplementować (czyli przystosować do pracy) w dowolnym języku programowania. Jeśli polskie określenia zastąpimy angielskimi, znanymi z różnych języków programowania (if...else), algorytm będzie zrozumiały na całym świecie.

cechy dobrego algorytmu

Dobry algorytm powinien mieć następujące cechy:

- **Precyzyjne zdefiniowanie** – algorytm nie może pozostawiać dowolności w interpretacji zapisu. Każdy, kto z niego skorzysta, musi go rozumieć w taki sam sposób.
- **Skończoność** – algorytm musi po pewnej liczbie kroków dać wynik, czyli wygenerować dane wyjściowe. Liczba operacji musi być skończona.
- **Wykonywalność** – wszystkie operacje opisane w algorytmie muszą być możliwe do wykonania.
- **Efektywność** – algorytm powinien być tak ułożony, by wynik jego działania był dostępny po jak najmniejszym zaangażowaniu zasobów komputera i w jak najkrótszym czasie. Zależy to jednak nie tylko od algorytmu, lecz także od właściwości komputera, języka użytego do ułożenia programu, systemu operacyjnego i wielu innych parametrów.



4.2. Zanim zaczniesz, czyli tworzenie specyfikacji algorytmu

Bez względu na to, jakiego sposobu zapisu algorytmu użyjesz przy tworzeniu algorytmu, musi on spełniać założenia projektu. Zapisuje się je w postaci tak zwanej **specyfikacji**. Powstaje ona podczas wstępnych prac nad algorytmem. Mogą to być na przykład uzgodnienia z zamawiającym lub rozwinięcie własnego pomysłu na program. Specyfikacja zazwyczaj składa się z trzech części. Najważniejszą, uzgadnianą z autorem pomysłu lub zamawiającym program, jest szczegółowy opis problemu. W specyfikacji należy koniecznie podać warunki, które muszą być spełnione przez dane, wyniki oraz zależności zachodzące między danymi wejściowymi a wynikami, na przykład wzory. Specyfikację można zapisać w dowolny sposób. Ważne, by uwzględniała wszystkie aspekty problemu. Tak przygotowana specyfikacja staje się podstawą do tworzenia algorytmu. Przeanalizujmy to na prostym przykładzie.

Tab. 4.1. Przykład specyfikacji problemu

Problem	Obliczanie równania kwadratowego
Opis problemu	Dane jest równanie kwadratowe w postaci: $y = ax^2 + bx + c$ Znając współczynniki a, b, c , wyznacz pierwiastki równia w dziedzinie liczb rzeczywistych. Uwzględnij wszystkie przypadki.
Dane wejściowe	Współczynniki a, b, c
Dane wyjściowe	Pierwiastki równania Komunikat o braku rozwiązania w dziedzinie liczb rzeczywistych dla delty mniejszej od 0.

Tak przygotowana specyfikacja może stać się podstawą do utworzenia algorytmu (rys. 4.3.).

```
start
wprowadź liczby a, b, c;
delta=b2-4ac;
jeżeli delta<0 drukuj "Brak rozwiązań w dziedzinie
liczb rzeczywistych";
jeżeli delta=0 {x=-b/2a; drukuj x;};
jeżeli delta>0 {x1=(-b+pierwiastek(delta))/2a;
x2=(-b-pierwiastek(delta))/2a; drukuj x1, x2;}
stop.
```

Rys. 4.3. Przykładowy algorytm w postaci pseudokodu

W specyfikacji, a następnie algorytmie opisuje się dane wejściowe i wyjściowe. Przypomnij sobie, czym one są. **Dane wejściowe** w wielu przypadkach to nie

tylko liczby lub tekst wprowadzany za pośrednictwem klawiatury. Wiele z nich dostaje się do komputera za pośrednictwem urządzeń peryferyjnych, takich jak mikrofon, skaner, kamera, mierniki temperatury, ciśnienia, zawartości płyt zawieszonych i innych wielkości fizycznych. Wszystkie muszą być przetworzone przez algorytm na dane wyjściowe.



Rys. 4.4. Przykłady danych wejściowych algorytmu

Dane wyjściowe powstają na skutek przetwarzania danych wejściowych zgodnie z algorytmem. Tu także mamy do czynienia z różnymi typami danych. Dane wyjściowe mogą być jednocześnie wejściowymi dla innych algorytmów. Przykładem może być klawiatura elektroniczna, która w zasadzie jest samodzielnym komputerem, który odczytuje stan klawiszy i wysyła odpowiednie kody – dane do komputera nadrzędnego, na przykład komputera osobistego, centrali domofonu, centrali zamka elektronicznego. Innym przykładem jest drukarka, dla której program – sterownik tworzy dane na podstawie dokumentów przeznaczonych do wydrukowania.

Takich przykładów można znaleźć jeszcze bardzo dużo. Domyślasz się już, że rodzaj danych wyjściowych może być taki sam, jak danych wejściowych albo



różnić się od nich w znaczny sposób. Algorytm może wygenerować liczby, teksty, ale także obraz, dźwięk, dane dla innych algorytmów.

Przed tobą ćwiczenia z układania algorytmów. Powodzenia.

ZADANIA DO ROZWIĄZANIA

1. Algorytm z rysunku 4.2. zapisz w postaci pseudokodu. Porównaj wszystkie trzy sposoby zapisu rozwiązania tego problemu. Który z nich jest dla ciebie czytelniejszy?
2. Na rysunku 4.4. przedstawiono niektóre z możliwych rodzajów danych wejściowych. Scharakteryzuj każdy z przykładów pod kątem pochodzenia lub urządzenia, które je wytworzyło.
3. Poznałeś/poznałaś kilka rodzajów danych wyjściowych. Zastanów się, przejrzyj zasoby sieci i wymień kilka innych. Opisz je krótko w zeszycie.
4. Pewien algorytm przetwarza dane z kamery HD na obraz z nałożonymi efektami graficznymi. Został wykorzystany w pewnym programie. Jakie może być jego przeznaczenie?
- 5*. Jakiego rodzaju dane wejściowe będą wykorzystywane przez algorytm, który opisuje obliczanie prędkości obrotowej koła? Omów konstrukcję potrzebnego urządzenia.

PODSUMOWANIE LEKCJI

- Każdy algorytm **przetwarza dane wejściowe na wyjściowe** w ściśle określony sposób. **Dane wyjściowe algorytmu** mogą być danymi wejściowymi innego algorytmu. s. 31
- **Lista kroków** to zapis algorytmu za pomocą słów. s. 31
- **Schemat blokowy** to graficzny sposób zapisu algorytmu. s. 31
- Do **cech dobrego algorytmu** należą: **precyzyjne zdefiniowanie procesów, skończoność, wykonywalność, efektywność**. s. 32
- **Pseudokod** jest sposobem zapisu algorytmu zbliżonym do języka programowania. Nie stosuje się w nim składni konkretnego języka. s. 32

5. Narzędzia, czyli jak dobrać i skonfigurować środowisko programistyczne

NA TEJ LEKCJI:

- poznasz nazwy popularnych języków programowania;
- poznasz środowisko IDE;
- dowiesz się, jak zainstalować polskie menu w Eclipse.

Który z języków programowania jest najlepszy? Wiesz już, że nie można odpowiedzieć jednoznacznie na to pytanie. Można jednak ocenić przydatność do danych zastosowań i sprawdzić, który z nich cieszy się największym powodzeniem wśród programistów. Wybierzemy dwa języki, z których jeden (C++) będzie towarzyszyć ci do końca nauki programowania na poziomie podstawowym, a drugi (Java) poznasz, jeśli podejmiesz naukę na poziomie rozszerzonym. Wraz z nimi poznasz także środowiska programistyczne.

5.1. Wieża Babel, czyli jak wybierać język programowania

Wśród języków, którymi posługujemy się w życiu, dominuje angielski. Samoistnie stał się językiem międzynarodowym. Dlaczego? To zagadnienie dla socjologów, a zjawisko zapewne jest skomplikowane i szczegółowo badane. A co z językami programowania? Ich także jest wiele.

Wśród nich znajdują się takie, których powszechność stosowania można porównać z językiem angielskim. Sprawdzisz to na stronie <https://www.tiobe.com/tiobe-index/> (rys. 5.1.).

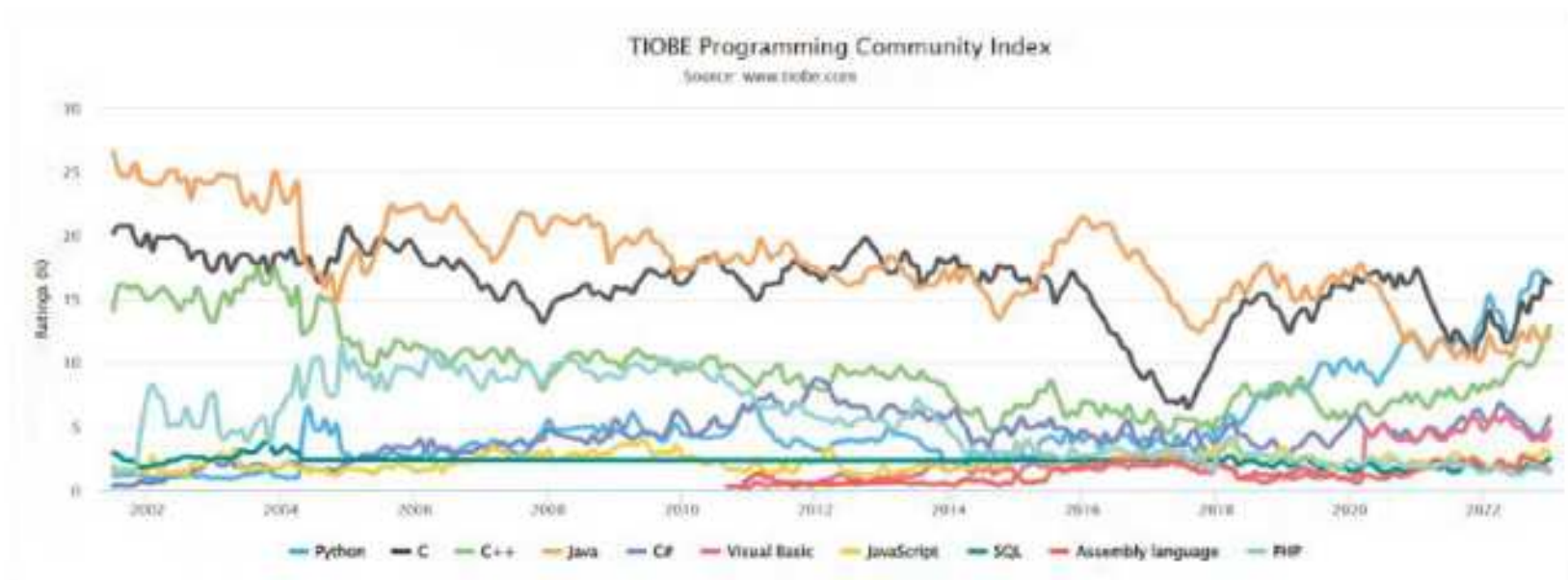
lista popularnych
języków
programowania

May 2019	May 2018	Change	Programming Language	Ratings	Change
1	1		Java	16.005%	-0.38%
2	2		C	14.243%	+0.24%
3	3		C++	8.095%	+0.43%
4	4		Python	7.830%	+2.64%
5	6	▲	Visual Basic .NET	5.103%	+1.07%
6	5	▼	C#	3.964%	-0.42%
7	8	▲	JavaScript	2.690%	-0.23%
8	9	▲	SQL	2.555%	+0.57%
9	7	▼	PHP	2.489%	-0.83%
10	13	▲	Assembly language	1.816%	+0.82%

Rys. 5.1. Przykład klasyfikacji popularności języków programowania wg TIOBE na podstawie ocen użytkowników



Popularność poszczególnych języków programowania zmienia się i zależy od wielu czynników – im szerszy jest zakres zastosowania, tym większy udział (rys. 5.2.). Dane ze strony TIOBE wynikają z badań opinii użytkowników.



Rys. 5.2. Przykład prezentacji popularności języków programowania na podstawie ocen użytkowników na przestrzeni lat

Analizując wyniki badań, można łatwo zauważyć dominację Java, C i C++. W związku z tym w naszych dalszych rozważaniach i nauce programowania na poziomie podstawowym będziemy wykorzystywać język C++.

Uwaga!

Porównując różne języki programowania, należy brać pod uwagę nie tylko popularność, lecz także specyfikę danego projektu. Nie można jednoznacznie określić, który z języków jest ogólnie najlepszy.

5.2. Krótka o filizance, czyli dlaczego Java jest tak popularna

Jeśli porównamy program ułożony w C++ i Java, możemy dojść do wniosku, że w Java jest on dłuższy (tab. 5.1.).

języki
programowania

Tab. 5.1. Fragmenty programów w Java i C++ wykonujących to samo zadanie

Java	C++
<code>int liczba=input.nextInt();</code>	<code>scanf("%d", &liczba);</code>

Czy to wada, czy zaleta? Z jednej strony używa się większej liczby znaków, ale z drugiej – polecenia są znacznie bardziej zrozumiałe. Przekonasz się o tym, wykonując ćwiczenia z naszego podręcznika i samodzielnie układając programy w obu językach.

Niedogodność wynikająca z używania dłuższych nazw rozkazów i poleceń nie utrudni pracy, jeśli zostanie dobrany edytor, który automatycznie uzupełnia tekst programu.

Wielu programistów wśród zalet Java wymienia jej niezależność od platformy, na której będzie uruchamiany program. Znając ten język, można tworzyć programy dla komputerów PC, smartfonów, serwerów stron itp.

Dla Java napisano wiele standardowych bibliotek (funkcji wykonujących różne działania). Dzięki nim nie ma potrzeby pisania ich od podstaw. Użycie bibliotek jest bardzo proste, o czym przekonasz się w dalszej części nauki programowania. Podobne biblioteki istnieją także dla innych języków. Ich liczba ciągle się zwiększa.

5.3. C++, czyli język nie tylko dla PC

język C++

C++ jest językiem uniwersalnym. Dzięki swoim właściwościom, na przykład możliwości precyzyjnego wykorzystania sprzętu, pozwala na efektywne gospodarowanie zasobami komputera, chociażby pamięcią, i aktywne oszczędzanie energii w urządzeniach mobilnych. Dużą zaletą jest szybkość działania, co oczywiście dla danego programu także zależy od umiejętności programisty.

Czy wiesz, w jakim języku pisze się na przykład systemy operacyjne, pakiety biurowe, skomplikowane gry? W większości tych zastosowań język C++ nie ma obecnie sobie równych. Oczywiście także i w tej dziedzinie powstają nowe języki i ich odmiany. Być może, gdy będziesz już zawodowym programistą, dzisiejszy układ zmieni się całkowicie. Jednak umiejętność programowania w Java i C++ pozwoli na szybkie opanowanie nowych narzędzi programistycznych i języków.

Języka C++ używa się także bardzo często do programowania mikrosterowników, robotów i innych urządzeń sterowanych komputerowo. Także amatorzy elektroniki i informatyki mogą programować w C++ niewielkie systemy mikroprocesorowe z rodziny ARDUINO lub bardziej skomplikowane Raspberry Pi.

5.4. IDE dla Java, czyli poznajemy środowisko programistyczne ECLIPSE

IDE

Czym jest **IDE** (ang. Integral Development Environment, czytaj: integral development environment)? Jeśli programowałeś/programowałaś już w szkole podstawowej, zapewne pamiętasz, że 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, jak Java i C++. Najprostsze, graficzne zapewne znasz z programowania w **Scratch** lub **mBlock**.

Eclipse

Do programowania w Java i C++ możesz użyć **Eclipse**. Jego zalety poznasz w trakcie nauki programowania.

Przygotowanie środowiska programistycznego do pracy wymaga kilku wcześniej zdobytych przez ciebie umiejętności. Możesz więc skorzystać z krótkiego algorytmu (rys. 5.3.) lub – jeśli będziesz miał/miała problem – sięgnąć do szczegółowego opisu.



Rys. 5.3. Instalacja IDE Eclipse w komputerze PC

Teraz poznasz proces instalacji i konfiguracji środowiska Eclipse dla języka Java. Mimo że w dalszej części będziesz wykorzystywać język C++, wykonaj to ćwiczenie. Przystosowanie Eclipse do C++ będzie wymagało niewielu dodatkowych czynności.

Jak każdy program komputerowy, IDE wymaga uruchomienia i konfiguracji. Proces przebiega nieco inaczej niż w przypadku większości programów, nie wszystko odbywa się automatycznie. Są komponenty, które musisz zainstalować samodzielnie. Przedstawiony opis instalacji (dotyczy to także opisu w rozdziale 6) jest przykładem takiego procesu. Jednakże podczas instalacji kolejnych wersji oprogramowania analiza opisu pomoże ci uporać się z takim zadaniem.

IDE nie zawiera niezbędnego do poprawnej pracy oprogramowania Java. W twoim systemie prawdopodobnie jest ono zainstalowane. Jeśli nie, Eclipse się nie uruchomi. Nie zrażaj się tym i odwiedź stronę www.oracle.com/technetwork/java/javase/downloads (rys. 5.4.) – postępuj według opisów lub rysunku 5.4. Wybierz JDK (Java Development Kit), ponieważ to darmowe oprogramowanie udostępniające środowisko niezbędne do programowania w języku Java. Takiego właśnie będziesz potrzebować do prawidłowej pracy Eclipse. Instalacja powinna przebiegać automatycznie. Czytaj uważnie komunikaty instalatora. Jeśli wykryje on starszą wersję oprogramowania i zaproponuje ci jej usunięcie – wyraż zgodę. Zwróć uwagę na wybór odpowiedniej wersji dla twojego systemu operacyjnego.

www.oracle.com/technetwork/java/javase/downloads

Java SE Development Kit 19.0.2 downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications and components using the Java programming language.

The JDK includes tools for developing and testing programs written in the Java programming language and running on the Java platform.

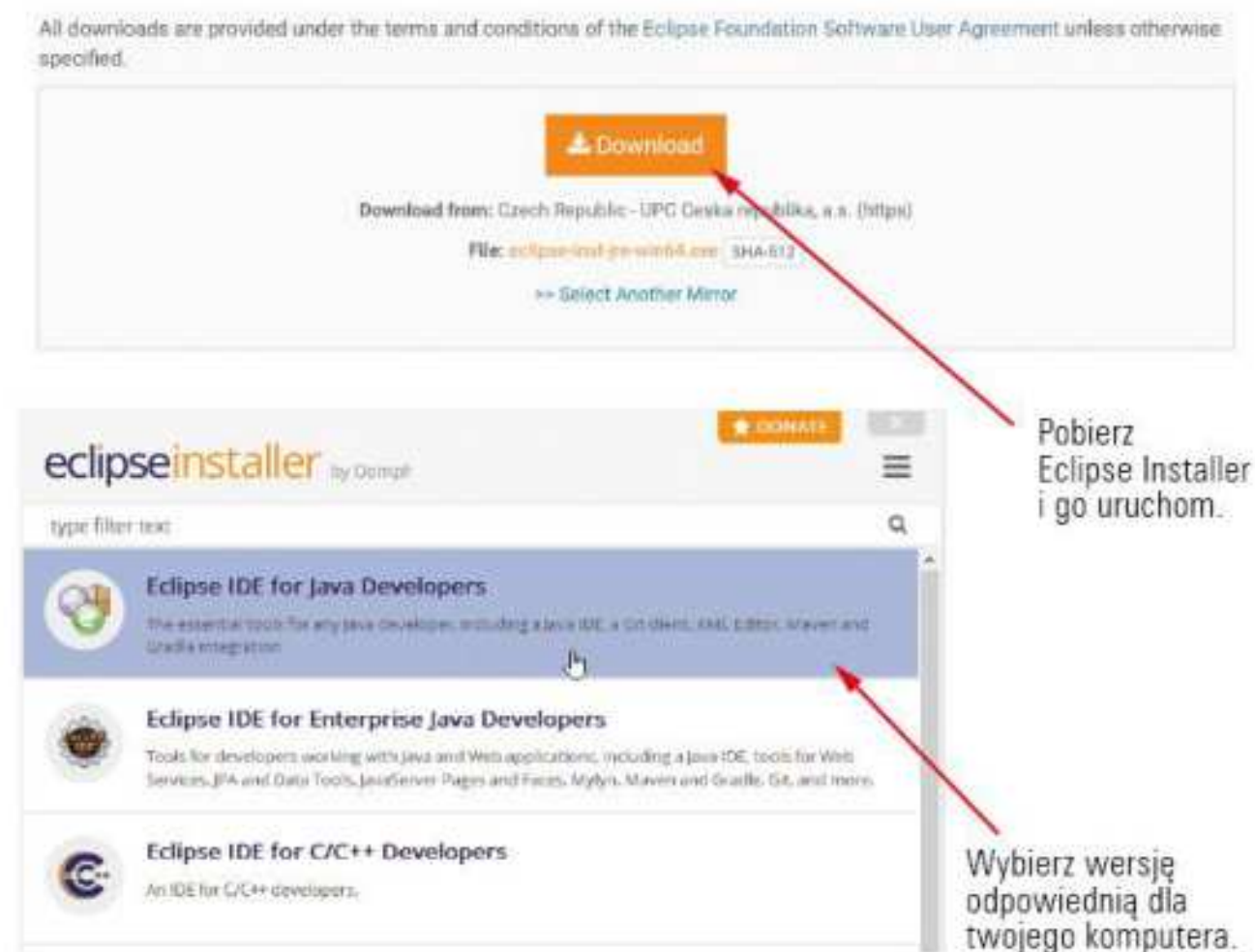
Linux	macOS	Windows
Product/Platform description	File size	Download
x64 Compressed Archive	17913 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.zip (sha256)
x64 Installer	158,914 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.exe (sha256)
x64 MSI Installer	15716 MB	https://download.oracle.com/java/19/latest/jdk-19_windows-x64_bin.msi (sha256)

Wybierz wersję odpowiednią dla twojego systemu.

Rys. 5.4. Instalacja oprogramowania Java Development Kit

Wszystko, czego teraz potrzebujesz do zainstalowania IDE dla Java, znajdziesz na stronie <http://www.eclipse.org>. Pobierz Eclipse Installer i go uruchom.

IDE dla Java i C++ różnią się, dlatego należy pobrać odpowiednie pliki – **Eclipse IDE for Java Developers**. Rozpocznijmy od środowiska do programowania w Java (rys. 5.5.). Przygotowaniem IDE dla C++ zajmiemy się w następnym rozdziale.



Rys. 5.5. Wybór IDE ze strony www.eclipse.org/downloads/packages/

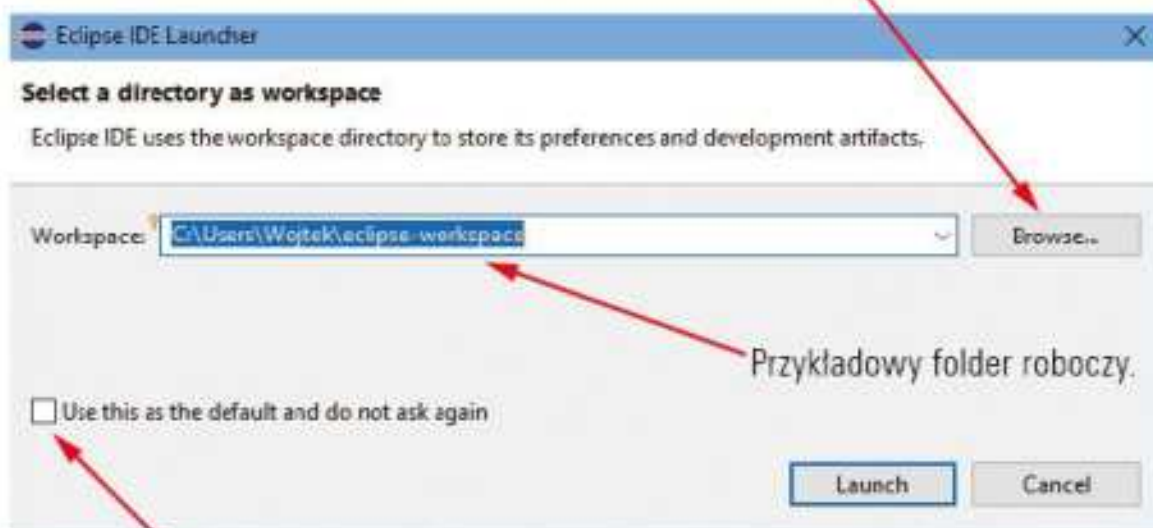
Pobrany plik musisz rozpakować. Został on poddany kompresji ZIP, dlatego twój system poradzi sobie z tym bez instalowania dodatkowego oprogramowania. Wszystkie pliki i foldery Eclipse wypakuj do jednego foldera. Może być dowolny, lecz dla porządku umieść je w folderze utworzonym w *C:\Program Files*, na przykład *C:\Program Files\Eclipse Java*. Aby zyskać szybszy dostęp do Eclipse, utwórz skrót do pliku *eclipse.exe* i umieść go na pulpicie. Nadaj mu nazwę Eclipse dla Java, by odróżnić go od skrótu do środowiska dla C++.

Podczas pierwszego uruchomienia IDE poprosi o umiejscowienie **foldera roboczego** (rys. 5.6.). Jeśli nie zdecydujesz inaczej, to w nim będą zapisywane wszystkie twoje projekty.

IDE uruchomi się z oknem startowym (rys. 5.7.). W nim możesz wybrać na przykład dostęp do tutoriali, które mogą pomóc w użytkowaniu środowiska. Przejrzyj je, zanim rozpoczniesz używanie IDE. Jeśli usuniesz zaznaczenie pola

w prawym, dolnym narożniku i zamkniesz tę zakładkę, przy następnych uruchomieniach nie będzie się ono już otwierało.

Możesz wskazać istniejący folder lub go utworzyć w dowolnym miejscu dysku.



Przykładowy folder roboczy.

Jeśli zaznaczysz to pole, prośba o nazwę foldera roboczego podczas tworzenia nowego projektu nie będzie się wyświetlać.

Rys. 5.6. Ustalenie foldera roboczego dla programów w Eclipse

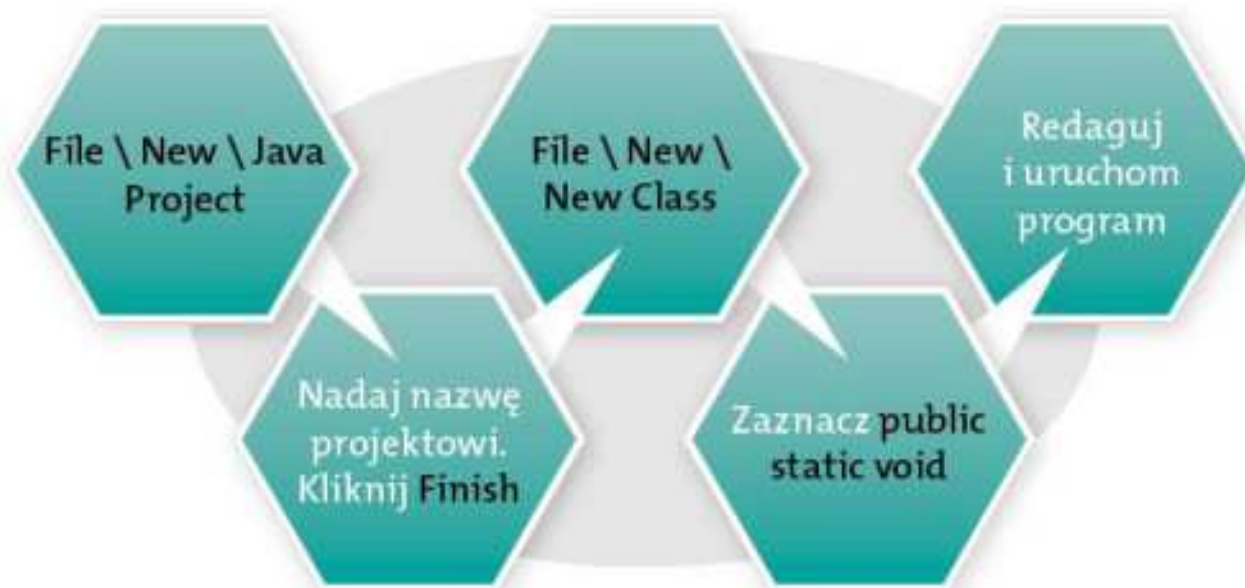


Jeśli usuniesz zaznaczenie pola w prawym, dolnym narożniku i zamkniesz tę zakładkę, przy następnych uruchomieniach nie będzie się ono już otwierało.

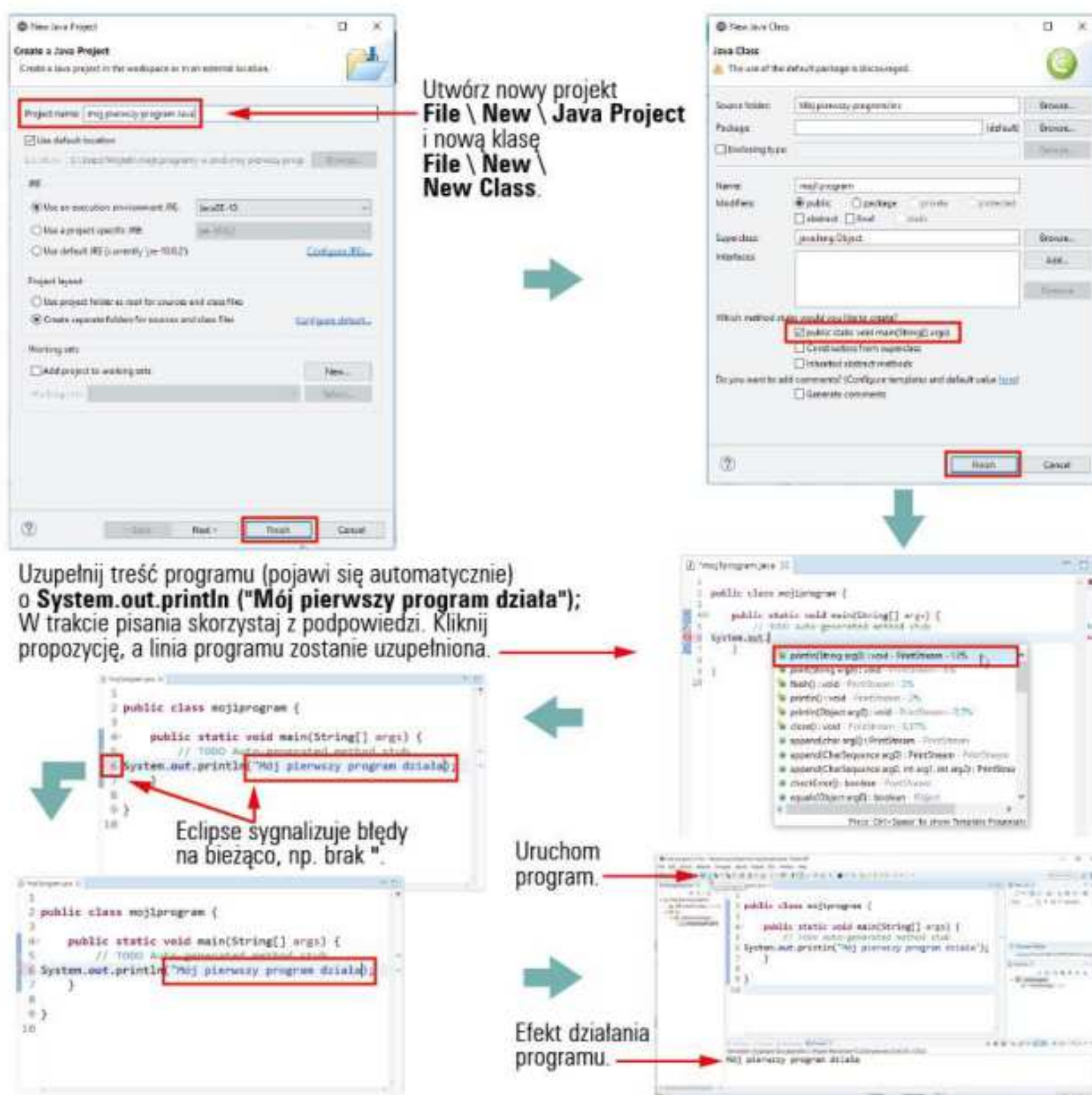
Rys. 5.7. Okno startowe Eclipse dla Java

IDE Eclipse jest już praktycznie gotowe do pracy. Czas sprawdzić, jak działa. Jedynym sposobem jest uruchomienie krótkiego programu, np. drukującego na ekranie „Mój pierwszy program działa”.

Działaj według algorytmu (rys. 5.8.). Jeśli potrzebujesz więcej szczegółów, posłuż się opisem z rysunku 5.9.



Rys. 5.8. Proces tworzenia nowego projektu w Eclipse dla Java



Rys. 5.9. Proces tworzenia i uruchamiania programu ułożonego w Java w IDE Eclipse



Efekt działania programu jest prawidłowy. Poddajmy analizie ten prosty kod.

```
public class moj1program
{
    public static void main(String[] args)
    {
        System.out.println("Mój pierwszy program działa");
    }
}
```

Po uruchomieniu tego programu będziesz pewny/pewna, że przygotowanie środowiska IDE zostało wykonane prawidłowo. Zapewne nie znasz języka Java, dlatego nie musisz jeszcze rozumieć znaczenia poszczególnych linii kodu programu. Jeśli jednak chciałbyś/chciałabyś poznać działanie tego programu, pomoże ci poniższy opis.

Pierwsza linia kodu określa metodę rozruchową programu, od której zaczyna się wykonywanie wszystkich instrukcji.

```
public class moj1program
```

Publiczna klasa o nazwie `moj1program`.

`public` – specyfikator dostępu mówi o zasięgu klasy.

`class` – słowo kluczowe występujące przed nazwą danej klasy.

```
public static void main(String[] args)
```

Taki zapis oznacza metodę rozruchową programu. W tym przypadku to metoda statyczna, co oznacza, że z wnętrza tej metody nie można odwoływać się do metod niezadeklarowanych jako statyczne. Jeśli chcesz poznać cechy metod i dokładne znaczenie słów kluczowych `public` i `static`, to sięgnij do źródeł opisujących język Java.

`void` – słowo to określa typ zwracany przez metodę – w tym przypadku oznacza, że metoda nie zwraca żadnej wartości.

Zasadnicza część programu:

```
System.out.println("Mój pierwszy program działa");
```

drukuje ciąg znaków podany w cudzysłowie.

5.5. Plik zamiast File, czyli Eclipse po polsku

Wszystkie menu ułożono w języku angielskim. Jeśli chcesz zmienić język na polski, postępuj jak na pokazano na rysunkach 5.10. i 5.11. Dokładną instrukcję znajdziesz pod adresem www.eclipse.org/babel/downloads.php.

Zauważ, że język polski w menu i oknach dialogowych pomaga zrozumieć przeznaczenie poszczególnych opcji, jednak utrudnia późniejszą pracę i naukę programowania. Większość samouczków, instrukcji i przykładów opisuje Eclipse w wersji angielskiej, a jeśli zostaniesz programistą, także będziesz korzystać z wersji po angielsku.

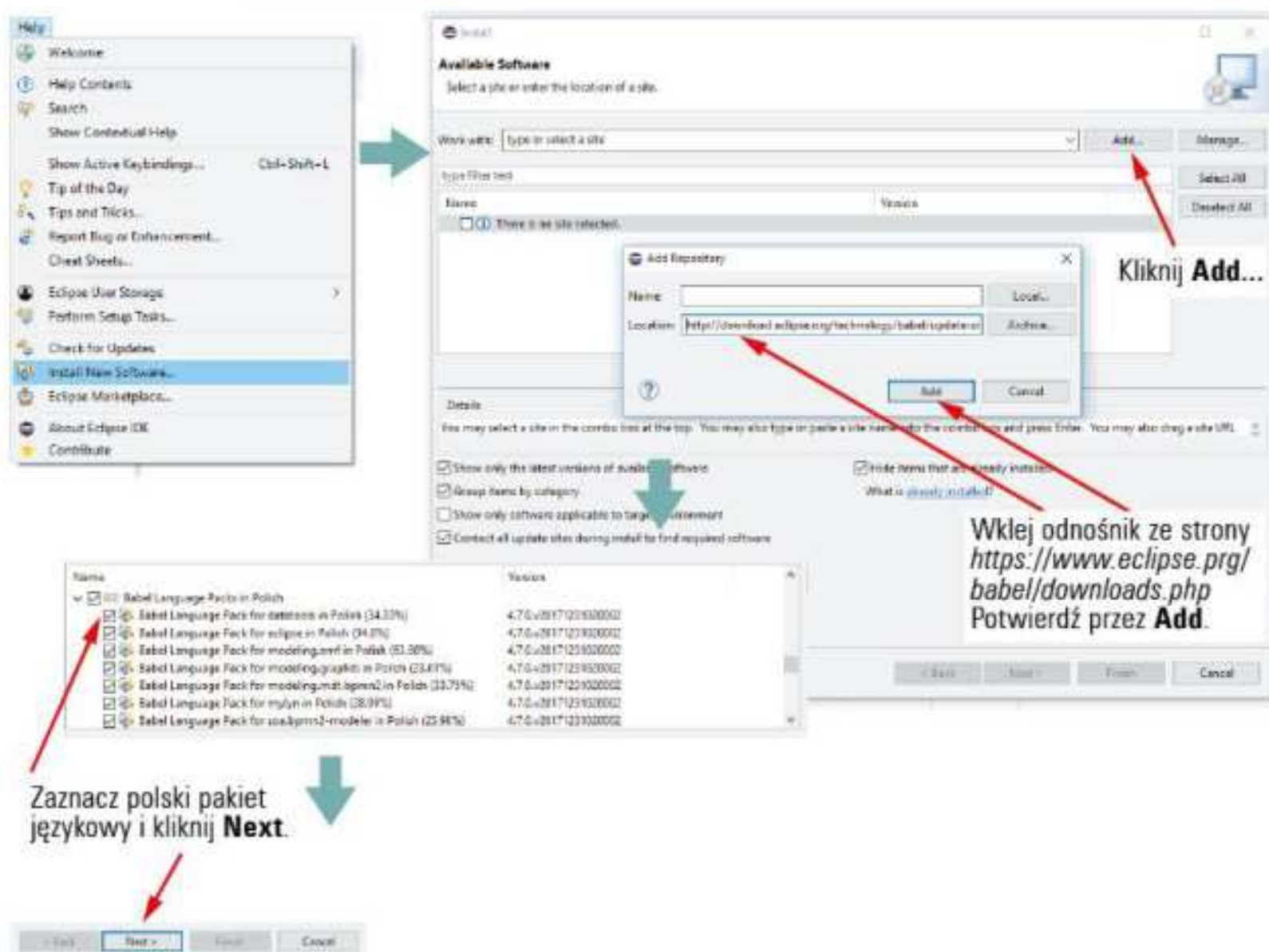
instalowanie
polskiego menu
w Eclipse

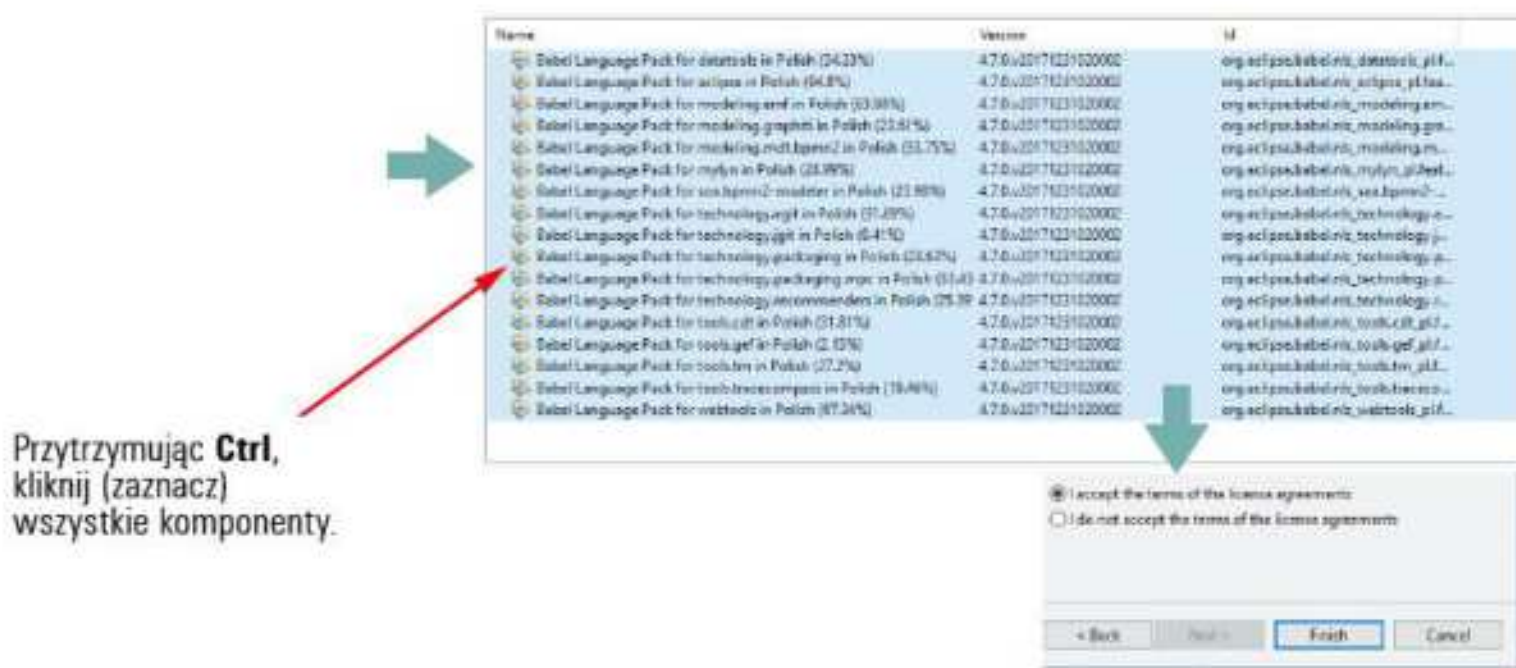


Rys. 5.10. Sekwencja czynności instalacji polskiego pakietu językowego dla Eclipse.

Uwaga!

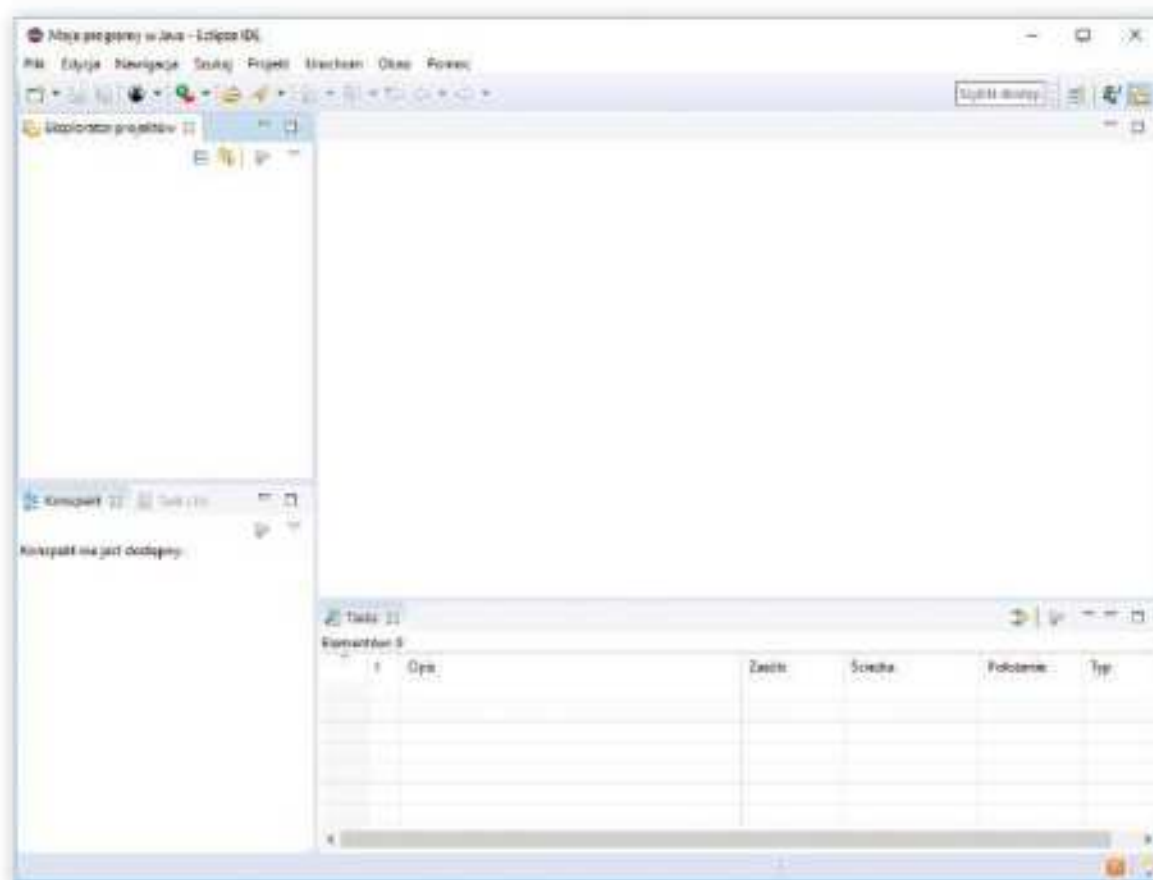
Jeśli nie zaznaczysz wszystkich komponentów polskiego pakietu, instalacja się nie powiedzie.





Rys. 5.11. Instalowanie polskiego pakietu językowego do Eclipse

Po wciśnięciu przycisku **Finish** nastąpi zamknięcie i ponowne uruchomienie Eclipse. Menu będą wyświetlane w języku polskim (rys. 5.12.).



Rys. 5.12. Okno IDE Eclipse z menu w języku polskim

Uwaga!

Jeśli chcesz korzystać z polskiego menu w obu instalacjach Eclipse – C++ i Java, w obu musisz wykonać te same czynności.

Polskie wersje oprogramowania IDE mogą być przydatne w początkowej fazie nauki programowania, lecz rzadko są stosowane przez zawodowych programistów.

ZADANIA DO ROZWIĄZANIA

1. Przeczytaj jeszcze raz fragment rozdziału o instalacji polskiego pakietu językowego. W nowym folderze umieść zawartość pliku (pobranego w trakcie ćwiczenia z rozdziału) z IDE Eclipse dla Java. Stamtąd uruchom `java.exe`, a następnie zainstaluj polski pakiet językowy. Porównaj polskie menu z menu w języku angielskim. Otwórz dowolny tutorial i sprawdź, na ile język polski menu utrudnia jego zrozumienie.
2. Sprawdź na stronie TIOBE informacje na temat popularności i oceny języków programowania. Oprócz danych wydrukowanych w rozdziale znajdziesz tam wiele innych. Przeanalizuj je i wyciągnij wnioski, np. na temat tendencji i trendów. Jeśli nie znasz dostatecznie języka angielskiego, skorzystaj z translatora, np. Google.
3. Znajdź w internecie fora dla początkujących programistów i wątki dotyczące porównania środowisk programistycznych. Poczytaj opinie na temat Eclipse i porównaj z opiniami o innych IDE. Przedstaw wnioski.
- 4*. Zapoznaj się z możliwościami użycia Notepad++ do edycji programów w Java. Dowiedz się, co jeszcze oprócz edytora będzie potrzebne do uruchomienia ułożonych w nim programów.
- 5*. Uruchom Eclipse dla Java. Z menu **Pomoc** wybierz opcję **Ściągawki**. Odnajdź temat „Utwórz aplikację Hello World”. Korzystając z tej ściągawki, ułóż i uruchom program Hello World.

PODSUMOWANIE LEKCJI

- s. 36 • Aktualną listę najchętniej stosowanych i popularnych języków programowania znajdziesz na stronie www.tiobe.com/tiobe-index/.
- s. 37 • Nie można jednoznacznie stwierdzić, który z **języków programowania** jest najlepszy. Przydatność danego języka zależy od rodzaju programu, jaki ma być w nim ułożony i w jakim urządzeniu lub systemie ma być uruchomiony.
- s. 38 • C++ jest językiem uniwersalnym. Dzięki swoim właściwościom, np. możliwości precyzyjnego wykorzystania sprzętu, pozwala na efektywne gospodarowanie zasobami komputera.
- s. 38 • IDE (ang. Integral Development Environment) to program lub zespół programów służących do tworzenia, testowania, modyfikowania i konserwacji programów komputerowych w danym języku.
- s. 38 • Eclipse to IDE zaprojektowane dla programistów posługujących się różnymi językami programowania w tym Java i C++. Eclipse zawiera system uzupełniania oraz podpowiadania poprawnej składni poleceń i rozkazów.
- s. 43 • Jeśli chcesz w Eclipse zainstalować menu w języku polskim, użyj opcji **Install New Software...** z menu Help. Będzie także potrzebny odnośnik do pakietu językowego ze strony <https://www.eclipse.org/babel/downloads.php>.



6. C z plusami, czyli uruchamiamy pierwsze programy w C++

NA TEJ LEKCJI:

- dowiesz się, jak poprawnie skonfigurować Eclipse dla C++;
- dowiesz się, jak postawić pierwsze kroki w programowaniu C++.

Zainstalowanie i uruchomienie IDE jest tylko pierwszym krokiem do nauki programowania. Język to tylko narzędzie. Dokonaliśmy już wyboru obu elementów, czas na tworzenie programów. Nauka programowania trwa długo, a sukces w dużym stopniu zależy od zaangażowania. Są jednak pewne zagadnienia, które powinien poznać każdy, zanim zacznie układać swoje pierwsze programy. Przypomnimy więc najważniejsze z nich. Przedtem przygotujemy środowisko IDE Eclipse dla języka C++.

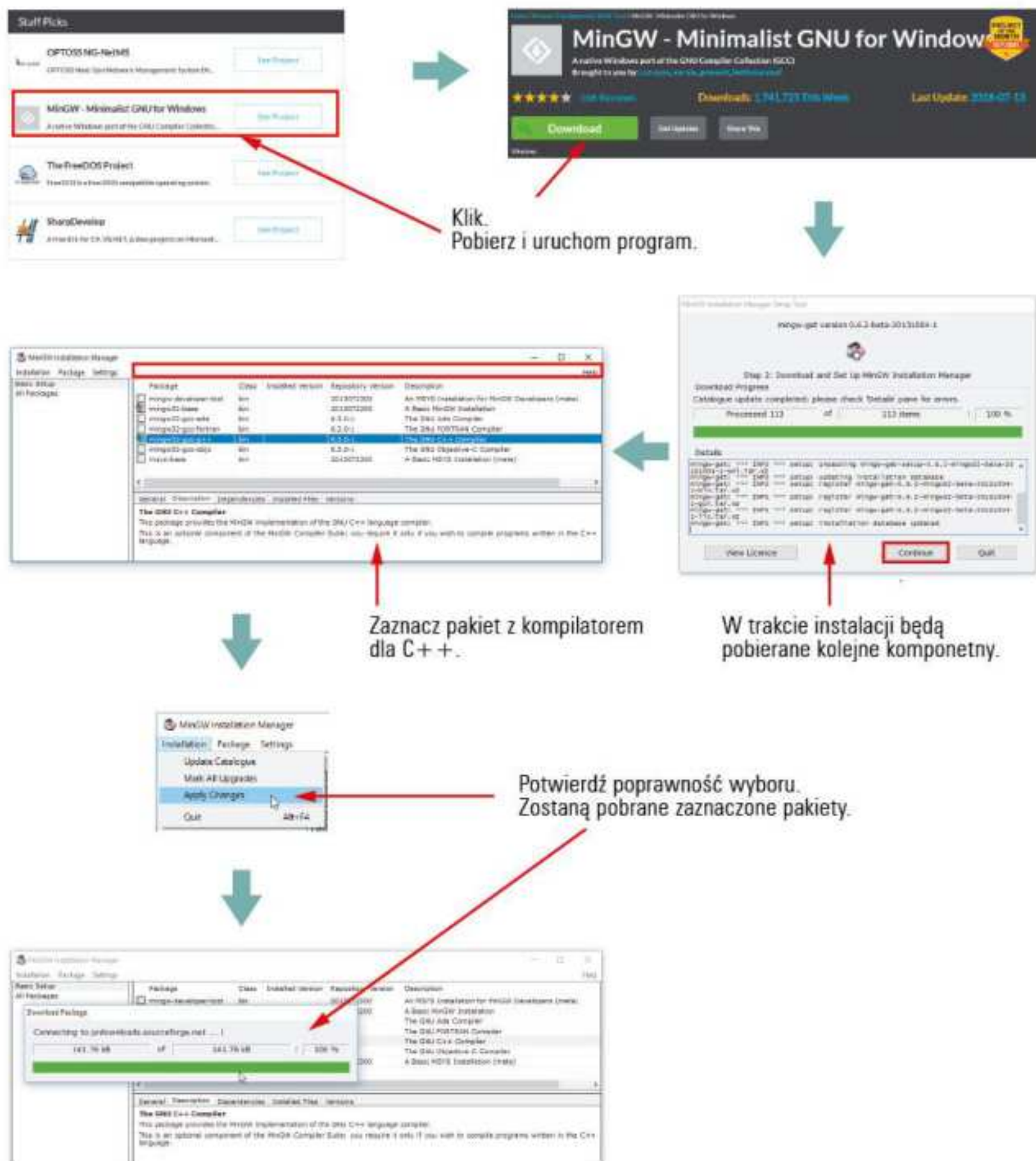
6.1. IDE dla C++, czyli czego jeszcze brakuje w Eclipse

Eclipse dla Java i C++ z pozoru się nie różnią. Te same okna i bardzo podobne menu. Jednak C++ wymaga zainstalowania dodatkowego oprogramowania, które zawiera narzędzia pozwalające na kompilację programów (kodów źródłowych) przeznaczonych dla Windows. Powszechnie używanym dla C++ jest **MinGW** (Minimalist GNU for Windows). Szczegółowe informacje na temat tego darmowego środowiska znajdziesz w internecie.

Zanim pobierzesz i uruchomisz Eclipse, przeprowadź instalację MinGW (rys. 6.1.). Odszukaj w sieci instalator tego środowiska. Możesz skorzystać ze strony *sourceforge.net*, na której znajdziesz wiele narzędzi typu Open-Source dla programistów.

Eclipse a język C++

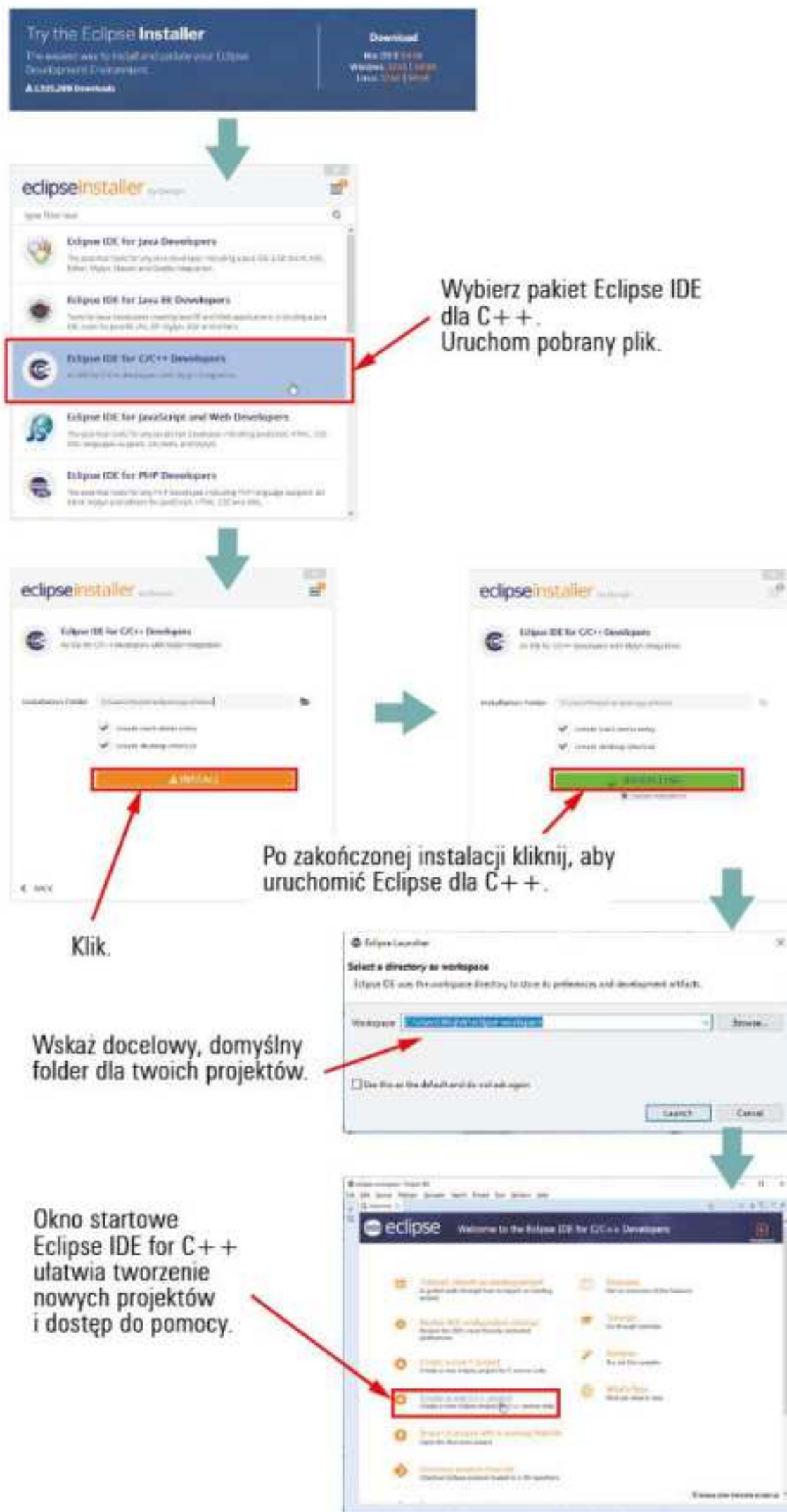
MinGW



Rys. 6.1. Instalacja pakietu MinGW

Po zainstalowaniu MinGW możesz już pobrać i uruchomić Eclipse dla C++. Proces wygląda podobnie jak w przypadku wersji dla Java.

Rozpocznij od pobrania pliku instalacyjnego ze strony <http://www.eclipse.org/downloads/packages/installer> (rys. 6.2.).



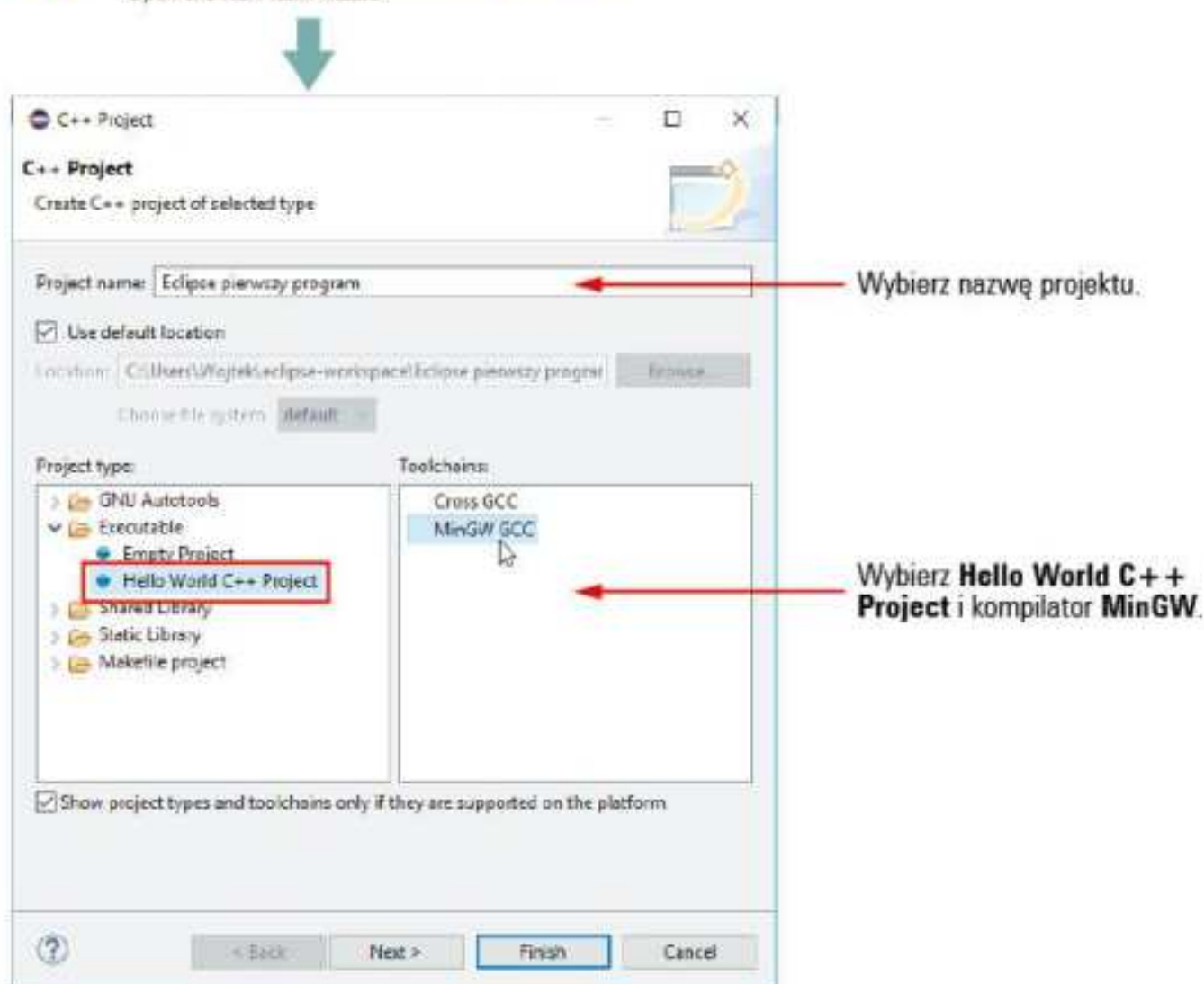
Rys. 6.2. Pobieranie i instalacja Eclipse dla C++

6.2. Zaczynamy, czyli pierwszy program w C++

Eclipse ułatwia stawianie pierwszych kroków w programowaniu w C++. Sprawdź to samodzielnie, wykonując czynności z rysunku 6.3.

-  **Create a new C project**
Create a new Eclipse project for C source code
-  **Create a new C++ project**
Create a new Eclipse project for C++ source code
-  **Import a project with a working Makefile**
Open the New Item wizard

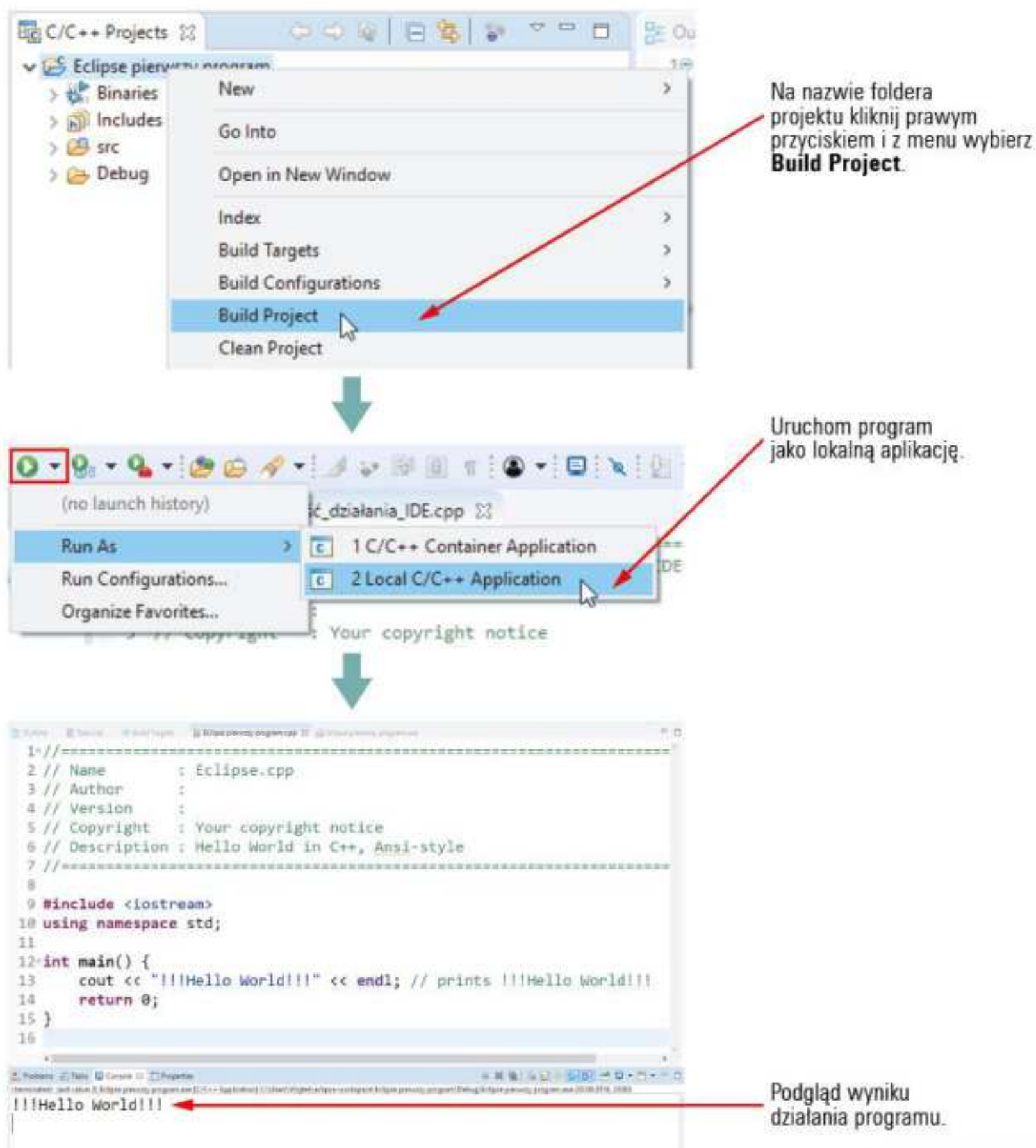
W oknie startowym wybierz tworzenie nowego projektu. Jeśli wyłączyłeś/wyłączyłaś wyświetlanie okna, użyj opcji z menu **File**.



Rys. 6.3. Tworzenie nowego projektu w Eclipse C++ z programem testowym



Twój projekt jest gotowy do uruchomienia. Napis **!!!Hello World!!!** możesz zmienić na dowolny. Aby sprawdzić działanie programu i przetestować IDE, uruchom ten program (rys. 6.4.).



Rys. 6.4. Uruchomienie przykładowego programu w Eclipse C++

Eclipse C++

Wiesz już, że program działa. Świadczy to o tym, że prawidłowo zainstalowano IDE Eclipse. Warto jednak bliżej przyjrzeć się programowi (rys. 6.5.).



```

1//=====
2// Name      : Eclipse.cpp
3// Author    :
4// Version   :
5// Copyright : Your copyright notice
6// Description: Hello World in C++, Ansi-style
7//=====
8
9#include <iostream>
10using namespace std;
11
12int main() {
13    cout << "!!!Hello World!!!" << endl; // prints !!!Hello World!!!
14    return 0;
15}

```

Informacje o autorze, właścicielu praw itp. bez znaczenia dla jego działania. Tekst po `//` jest komentarzem.

Rys. 6.5. Fragment programu w Eclipse C++

`#include <iostream>`

znak #

Znak `#` przed instrukcją oznacza, że jest ona dyrektywą preprocesora. Wykonanie takiego polecenia następuje 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).

`using namespace std;`

Oznacza informację 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 mieć takie same nazwy. 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 należąliby do różnych klas, to ich przestrzenie są oddzielne i te osoby nie zostaną pomyłone. W przestrzeni szkoły będą kolidowały i mogą prowadzić do pomyłek.

`int main()`

Oznacza początek głównej funkcji programu. Słowo kluczowe `int` oznacza, iż system operacyjny będzie informowany o wyniku działania programu. Najczęściej nie jest to konieczne i w takim przypadku można użyć słowa kluczowego `void`. Na końcu funkcji z nagłówkiem `int` należy dopisać `return 0;`. Będzie to informacja dla systemu o zakończeniu operacji (przekazanie wartości 0).

`cout << "!!!Hello World!!!" << endl;`

To zasadnicza część programu, która powoduje wyświetlenie napisu. Ciąg znaków komunikatu należy umieścić w cudzysłowie.

Słowo kluczowe – słowo, które w danym języku programowania ma już przypisane znaczenie.

**Uwaga!**

Długie komentarze umieszcza się pomiędzy `/*` a `*/`.

Skoro uruchomiłeś/uruchomiłaś IDE, możesz rozpocząć programowanie. Przypomnij sobie podstawy z poprzednich lat nauki. Następny rozdział pomoże ci uporządkować tę wiedzę.

ZADANIA DO ROZWIĄZANIA

1. Zmodyfikuj program opisany w rozdziale tak, by wyświetlał na ekranie inny tekst. Ustal doświadczalnie, jak wprowadzić poprawki, by tekst zajmował kilka linii. Przedstaw swoje rozwiązanie.
2. Po zainstalowaniu Eclipse dla C++ zdecyduj, czy będziesz korzystać z jego polskiej wersji językowej. Jeśli tak, zmień ją, opierając się na opisie wersji dla Java. 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 prawidłowej struktury programu ułożonego w C++.
- 5*. Zmodyfikuj opisany wyżej program w taki sposób, by pierwsze linie oznaczone jako komentarz zawierały twoje dane. Jeśli uważasz, że inne informacje, np. nazwa szkoły, klasy, powinny się tam znaleźć, dodaj linie. Znajdź sposób na to, by nowe projekty automatycznie rozpoczynały się od twoich komentarzy.

PODSUMOWANIE LEKCJI

- Środowisko programistyczne Eclipse może być wykorzystane do programowania w języku C++. Należy jednak pobrać i zainstalować odpowiednią wersję. Sposób posługiwania się podstawowymi opcjami Eclipse C++ jest taki sam, jak w przypadku środowiska Eclipse dla Java. s. 47
- Jednym z powszechnie stosowanych programów jest MinGW. Należy go zainstalować przed instalacją Eclipse dla C++. s. 47
- Do kompilacji C++ w Eclipse są potrzebne narzędzia pozwalające na kompilację programów w środowisku Windows. s. 51
- Dyrektywy oznaczone # są wykonywane przed kompilacją i powodują dołączenie odpowiednich bibliotek (zbiorów funkcji). s. 52

7. C z plusem, czyli podstawy programowania

NA TEJ LEKCJI:

- poznasz podstawową strukturę programu C++;
- dowiesz się, czym są operatory i instrukcje sterujące.

Twoje środowisko programistyczne jest już gotowe do pracy. Możesz zacząć programowanie. W tym rozdziale znajdziesz informacje, dzięki którym poznasz podstawowe zadania, jakie pełnią poszczególne polecenia i konstrukcje języka C++. W kolejnych rozdziałach poznasz ich działanie w praktyce. Tajniki C++ poznasz w zakresie niezbędnym do ułożenia mało skomplikowanych programów. Poznanie wszystkich zagadnień związanych z programowaniem w tym języku zajmuje dużo więcej czasu i ćwiczeń niż jest to przewidziane na szkolne lekcje.

7.1. Co i gdzie, czyli podstawowa struktura programu w języku C++

kompilacja

debugger

Układając program w języku C++, programista tworzy **kod źródłowy** składający się z wyrażen zrozumiałych dla człowieka. Kod ten po sprawdzeniu pod kątem poprawności składni jest **kompilowany**. Oznacza to, że jest budowany program wykonywalny, w którego skład wchodzi tłumaczenie na kod maszynowy kodu programu wraz z użytymi bibliotekami oraz nagłówek powodujący, że program będzie mógł być uruchamiany w systemie operacyjnym bez udziału środowiska programistycznego. Sprawdzaniem poprawności składni, uruchamianiem programu krok po kroku, podglądaniem bieżących wartości zmiennych oraz śledzeniem procedur i funkcji zajmuje się program określany jako **debugger** (ang. bug, czyli robak – potocznie o błędzie w programie). W trakcie kompilacji badaniem błędów składniowych zajmuje się **translator**, czyli tłumacz kodu programu na kod maszynowy. Jak już wiesz, edytor, debugger i kompilator składają się na środowisko programistyczne (rys. 7.1.).



Projektant oprogramowania

- projektowanie sposobu rozwiązania problemu
- tworzenie algorytmów



Praca programisty w środowisku programistycznym

- edycja kodu źródłowego
- debuggowanie – sprawdzanie poprawności kodu źródłowego
- kompilacja – tłumaczenie na kod maszynowy



Programista we współpracy z projektantem

- testowanie – sprawdzanie poprawności działania programu
- optymalizacja – poszukiwanie lepszych rozwiązań wewnątrz programu
- ponowne testowanie

Rys. 7.1. Kolejne etapy tworzenia programu komputerowego



Język, w którym mówimy, tworzą 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 funkcji.

Każdy program w C++ powinien mieć załączoną przynajmniej jedną bibliotekę, wyznaczoną przestrzeń nazw oraz przynajmniej jedną funkcję. Jeśli jest jedna, to powinna to być:

```
main();
```

Przykładowy program spełniający te warunki przedstawiono na rysunku 7.2.

Wszystkie wiersze zaczynające się od `#` to dyrektywy. Nie służą one do realizacji algorytmu. Informują jednak debugger i kompilator o dołączeniu do programu pliku zawierającego na przykład bibliotekę lub fragment kodu.

```
1 //=====
2 // Name      : program_w_cpp.cpp
3 // Author    : Wojtek
4 // Version   : 1.1
5 // Copyright : Wojtek
6 // Description : struktura programu w C++
7 //=====
8
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     cout << "Jak się masz?" << endl;
14     return 0;
15 }
```

Linia programu zaczynająca się `//` jest traktowana jako komentarz.

#include dołącza bibliotekę o nazwie podanej pomiędzy znakami `<` oraz `>`.
using namespace ustala przestrzeń nazw.

Funkcja **main()** tworzy blok główny programu. Wszystko, co znajduje się pomiędzy `{}`, będzie wykonywane aż do napotkania instrukcji **return**.

Rys. 7.2. Przykładowa struktura programu w języku C++

`using namespace` informuje kompilator, że będzie używana dana przestrzeń nazw. Dla standardowych bibliotek C++ to `std`. Dzięki temu możesz napisać:

```
cout << "jak się masz" << endl; zamiast std::cout << "jak się masz" << endl;
```

W przypadku jednej linii nie ma to większego znaczenia, jednak w programie takich linii jest bardzo dużo. Domyślasz się więc, że definiowanie przestrzeni nazw może znacznie uprościć układanie programu.

7.2. Operatory, czyli działania i relacje

Czym są operatory? Dzięki nim jest możliwe zapisanie działań matematycznych (tab. 7.1.) i relacji (tab. 7.2.).

operatory

Tab. 7.1. Operatory arytmetyczne w języku C++

Działanie	Operator	Przykład
dodawanie	+	$y = x + 1$
odejmowanie	-	$y = x - 4$
mnożenie	*	$y = x * z$
dzielenie	/	$y = (x + 25) / z$
reszta z dzielenia (modulo)	%	$y = z \% 2$

Tab. 7.2. Operatory relacyjne w języku C++

Opis	Operator
mniejszy	<
wiekszy	>
równy	==
różny	!=
mniejszy lub równy	<=
wiekszy lub równy	>=

Przykładowo: jeśli napiszesz $y = (x - 4) \% 7$, będzie to oznaczało, że y będzie równe reszcie z dzielenia $(x - 4)$ przez 7. Wynik takiego działania jest zawsze liczbą całkowitą. Przykładowo: dla $x = 20$ $y = (20 - 4) \% 7 = 2$.

7.3. Zmienne, czyli przechowywanie danych

zmienne

Definiując zmienną, nadajemy jej nazwę. Jeśli na przykład zdefiniujesz zmienną `komunikat = "wprowadź pierwszą liczbę";`, będzie to oznaczać, że pod nazwą `komunikat` kryje się tekst ujęty w cudzysłów. Jeśli danego tekstu będziesz używać wielokrotnie, możesz zapisać go w zmiennej. Będzie on także mógł być modyfikowany. W wielu programach i urządzeniach ten sam komunikat wyświetla się w różnych sytuacjach, na przykład „Dziękujemy i zapraszamy ponownie”. Jeśli zadeklarujesz zmienną (dla tekstu – `string`) i nadasz jej taką wartość, to w kodzie programu nie musisz używać całego komunikatu w poleceniach wyświetlania. Na przykład:

string

```
//deklaracja zmiennej przechowującej tekst
string komunikat;
//nadanie wartości zmiennej
komunikat = " Dziękujemy i zapraszamy ponownie ";
//wyświetlenie tekstu zawartego w zmiennej
cout << komunikat;
```



Jeżeli wyświetlany komunikat będzie długi, to oceń, jaki jest zysk dla programisty. Łatwo też na przykład zmienić język komunikatów. Wystarczy jedynie zmienić wartość zmiennej.

Można zmodyfikować program z rysunku 7.2. w taki sposób, by strumień z napisem zawierał wartość zmiennej `komunikat`.

```

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 }

```

Zadeklarowano zmienną typu **string**, nadając jej wartość.

Ta linia nie wydrukuje słowa „komunikat”, lecz to, co kryje się pod zmienną **komunikat**.

Rys. 7.3. Przykład zastosowania zmiennej typu string

Tab. 7.3. Typy danych liczbowych lub logicznych, które można przechowywać w zmiennych

Nazwa typu	Liczba bajtów	Zakres wartości
<code>bool</code>	1	false lub true (fałsz lub prawda; 0 lub 1)
<code>char</code>	1	od -128 do 127
<code>unsigned char</code>	1	od 0 do 255
<code>wchar_t</code>	2	od 0 do 65 535
<code>short</code>	2	od -32 768 do 32 767
<code>unsigned short</code>	2	od 0 do 65 535
<code>int</code>	4	od -2 147 483 648 do 2 147 483 647
<code>unsigned int</code>	4	od 0 do 4 294 967 295
<code>long</code>	4	od -2 147 483 648 do 2 147 483 647
<code>unsigned long</code>	4	od 0 do 4 294 967 295
<code>long long</code>	8	od -9 223 372 036 854 775 808 do 9 223 372 036 854 775 807
<code>unsigned long long</code>	8	od 0 do 18 446 744 073 709 551 615
<code>float</code>	4	3.4E +/- 38 (7 cyfr)
<code>double</code>	8	1.7E +/- 308 (15 cyfr)
<code>long double</code>	8	1.7E +/- 308 (15 cyfr)

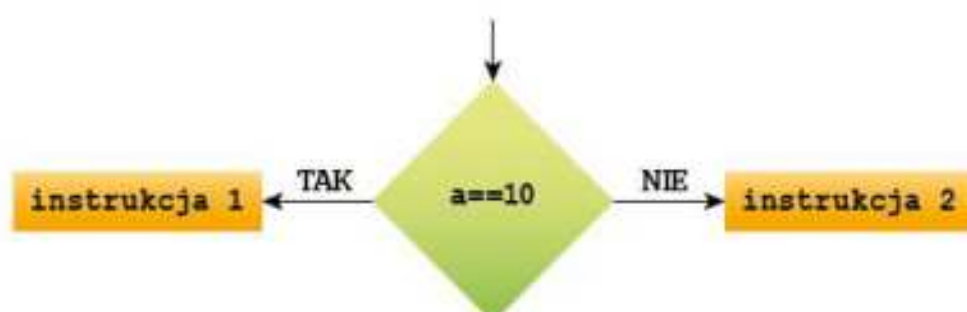
Pamiętasz zapewne, że zmienne mogą przechowywać różne typy danych. Dlaczego jest ich wiele? Otóż zdefiniowanie zmiennej rezerwuje dla niej miejsce w pamięci. Jeśli więc wiemy, że nasza zmienna przyjmuje wartości od 0 do 255, wystarczy zarezerwować dla niej jedynie bajt. Jeśli jednak zmienna przyjmuje wartość całkowitą dodatnią w bardzo dużym zakresie w C++, można zarezerwować jej aż 8 bajtów i największą liczbą, jaką można wpisać pod zmienną, będzie 18 446 744 073 709 551 615.

Wykaz wszystkich typów danych liczbowych, dla których można zadeklarować zmienną, znajdziesz w tabeli 7.3.

Jak myślisz, ile bajtów zajmie liczba 25 wpisana do zmiennej zadeklarowanej jako `long`? Oczywiście tego typu zmienna zawsze zajmie w pamięci 4 bajty, bez względu na to, jaką wartość z jej zakresu przyjmie.

7.4. Instrukcje sterujące, czyli realizacja bloków warunkowych algorytmu

W każdym algorytmie, który wymaga rozgałęzień, stosuje się bloki decyzyjne (rys. 7.4.). Jeśli w przedstawionym przykładzie warunek zostanie spełniony, to



jest realizowana `instrukcja 1`, jeśli nie – `instrukcja 2`.

Do zapisania warunków w języku C++ używa się operatorów logicznych i symboli porównawczych (tab. 7.4.).

Rys. 7.4. Przykład bloku decyzyjnego

Tab. 7.4. Operatory logiczne i relacyjne w języku C++

Operator	Symbol relacyjny	Działanie
<code>&&</code>		koniunkcja (I)
<code> </code>		alternatywa (LUB)
<code>!</code>		negacja (NIE)
	<code>==</code>	równy
	<code>!=</code>	różny
	<code>></code>	większy
	<code><</code>	mniejszy
	<code>>=</code>	większy lub równy
	<code><=</code>	mniejszy lub równy

Dzięki blokowi decyzyjnemu można rozgałęziać algorytmy i w zależności od spełnienia lub niespełnienia warunku podejmować odpowiednie działania. W językach programowania takie bloki zapisuje się na przykład za pomocą instrukcji warunkowej. W wielu językach programowania wygląda to bardzo podobnie. W C++ ma ona postać jak na rysunku 7.5.

```
//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. 7.5. Instrukcje warunkowe w C++

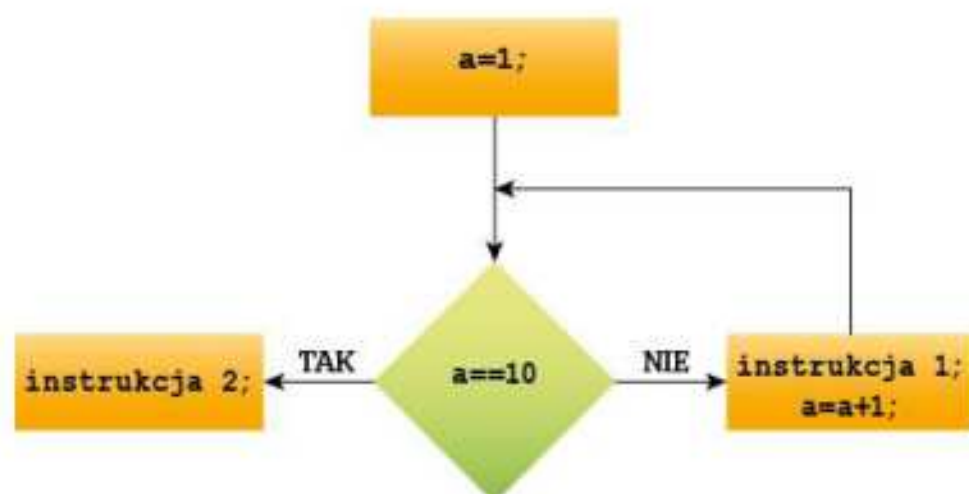
Jeśli uczyłeś/uczyłaś się programowania w Scratch, to zapewne pamiętasz badanie warunków (rys. 7.6.).

Użycie odpowiedniej postaci instrukcji warunkowej zależy od algorytmu, który należy zrealizować.

Jeśli algorytm zawiera zapętlenie (rys. 7.7.), realizowane dopóki spełniony jest warunek, można użyć **instrukcji pętli**.



Rys. 7.6. Instrukcje warunkowe Scratch



Rys. 7.7. Przykład pętli w algorytmie

Poznaj dwie postaci pętli stosowane w języku C++. Pierwsza z nich (rys. 7.8.) jest wykonywana wtedy, gdy spełniony jest warunek, aż do chwili, gdy warunek przestanie być spełniony; a w drugiej (rys. 7.9.) warunek zostaje sprawdzony po wykonaniu wszystkich instrukcji, które obejmuje instrukcja. Oznacza to, że w drugim przypadku wszystkie polecenia i instrukcje pętli zostaną wykonane przynajmniej raz i powtarzane tak długo, aż warunek zostanie spełniony.

```
while (a==10)
{
instrukcja 1; //Instrukcja 1 i 2 wykonywana dopóki
instrukcja 2  //jest spełniony warunek
}              //W przypadku jednej instrukcji {} nie są
              //wymagane
```

Rys. 7.8. Przykład pętli while w języku C++

```
do
{
instrukcja 1; //Instrukcje 1 i 2 wykonywane
instrukcja 2; //przynajmniej raz i powtarzane
}            //
while (a==10); //dopóki warunek jest spełniony
//W przypadku jednej instrukcji {} nie są wymagane
```

Rys. 7.9. Przykład pętli do ... while w języku C++

Fragment algorytmu z rysunku 7.7. można także zrealizować za pomocą pętli for (rys. 7.10.).

```
for (i=1; i<10; i++) //Począwszy od i=1, dopóki
{
// i<10,wykonuj instrukcje w { }
instrukcja 1;      // za każdym razem zwiększając
instrukcja 2;      // i o 1
}
```

Rys. 7.10. Instrukcja sterująca for w języku C++

Zauważ, że instrukcja z algorytmu $i = i + 1$; jest realizowana jako parametr instrukcji sterującej for (i++). Czym więc różni się w tym przypadku konstrukcja while od for? For jest wykonywana pewną ustaloną liczbę razy. W przykładzie dla a od 1 do 10 pętla for zostanie wykonana 9 razy. Przy a równym 10 nastąpi wyjście z pętli (rys. 7.7.). W przypadku pętli while lub do while liczba wykonywanych pętli zależy od spełnienia warunku. W przypadku pętli do while warunek jest sprawdzany na końcu, po wykonaniu instrukcji. Dlatego ta pętla zawsze będzie wykonana chociaż raz.

Pętle mogą się zagnieżdżać, czyli występować wewnątrz innych pętli. Podobnie jak instrukcje warunkowe.



ZADANIA DO ROZWIĄZANIA

1. Podaj na przykładzie różnicę pomiędzy instrukcją warunkową `if` a `if else`.
2. Określ, ile razy zostanie zrealizowana pętla `for`, jeśli będzie miała postać:

```
for (a=0; a<=10; a++)
{
    cout <<"pętla nr "<<a <<endl;
}
```

Porównaj tę pętlę z użytą w przykładzie (rys. 7.10.).

```
for (a=1; a<10; a++)
```

Czy ich działanie w programie będzie takie samo czy różne?

3. Na dowolnych przykładach wskaż różnice pomiędzy pętlą `while` a `do while`.
- 4*. Narysuj w zeszycie algorytm dla fragmentu programu ułożonego w języku C++. Wzoruj się na przykładach z rozdziału.

```
{
    a=0;
    do
    {
        cout << "informatyka" << endl;
        a=a+1;
    }
    while (a!=10);
    return 0;
}
```

- 5*. Ile razy w konsoli z zadania 4. zostałby wydrukowany napis „informatyka”? Uzasadnij swoją odpowiedź.

PODSUMOWANIE LEKCJI

- Języki programowania mają ściśle ustaloną składnię sprawdzaną w trakcie tworzenia kodu przez **debugger**. s. 54
- **Kompilacja** polega na zamianie kodu źródłowego programu na język zrozumiały przez komputer. Nazywamy go kodem maszynowym. s. 54
- **Działania arytmetyczne i logiczne** są zapisywane w języku C++ za pomocą operatorów. s. 55
- **Zmienne** mogą być deklarowane dla różnych typów danych. Dla różnych typów są rezerwowane w pamięci różne ilości bajtów. s. 56
- Tekst zapisuje się w zmiennych typu **string**. s. 56
- Do realizacji bloków decyzyjnych w algorytmach w zależności od tego, co realizuje algorytm, można użyć **pętli lub instrukcji warunkowych**. s. 59

8. C z plusem, czyli podstaw programowania ciąg dalszy

NA TEJ LEKCJI:

- poznasz zastosowanie i znaczenie funkcji oraz tablic i strumieni.

Znasz już zasady tworzenia algorytmów i niektóre elementy języka programowania. Przed tobą zadania programistyczne, do których niezbędne będzie poszerzenie tej wiedzy i zdobycie nowych umiejętności. Poznasz zastosowanie oraz znaczenie funkcji, tablic i strumieni. Po zdobyciu tej wiedzy przystąpisz do tworzenia własnych programów.

8.1. Tablice bez kredy, czyli jedna nazwa, wiele zmiennych

tablice

Wiesz, że przed użyciem zmiennej musisz ją zadeklarować, podając typ wartości, które będzie przechowywać i oryginalną nazwę. Istnieją jednak dane związane ze sobą, na przykład kolejne wyniki losowania, codzienne pomiary temperatury. Deklarowanie oddzielnej zmiennej dla każdej z nich nie byłoby wygodne i znacznie utrudniałoby układanie programów.

Przykład 1.

Lista uczniów klasy zawiera 30 nazwisk. Każde ma w dzienniku przydzielony swój numer. Jeśli zmienną nazwalibyśmy przykładowo: `klasala`, to zmienna `klasala[5]` zawierałaby nazwisko spod numeru 5 z dziennika.

Zmiennym przewidzianym dla wielu danych można więc nadać jedną nazwę. Sposób deklaracji jest podobny do deklarowania pozostałych zmiennych.

Przykład 2.

Zmienne dla pojedynczej wartości:

```
float temperatura;
```

Jeśli pomiar temperatury ma się odbyć co godzinę w ciągu jednej doby, zadeklarujemy zmienną tablicową (macierz) jednowymiarową.

```
float temperatura[23];
```

Zapewne zastanawiasz się, dlaczego dla 24 pomiarów w deklaracji tablicy użyto liczby 23. Nie ma tu błędu, ponieważ indeksacja (numeracja) poszczególnych elementów tablicy zaczyna się od 0. Zatem pierwszy pomiar zostanie zapisany pod `temperatura[0]`, a na przykład 6 pod `temperatura[5]`.



A co jeśli takie pomiary będzie trzeba wykonać przez 7 dni? Wtedy można użyć macierzy dwuwymiarowej, którą można porównać do rozmieszczenia krzeseł w sali kinowej. Jeśli znasz rząd i numer krzesła, trafisz na swoje miejsce.

Przykład 3.

Tablica przechowująca wyniki pomiarów temperatury co godzinę przez 7 dni:

```
float temperatura[6][23];
```

[illegible]

Rys. 8.1. Przykład danych wpisanych do zmiennej tablicowej

Domyślasz się, że możliwe jest deklarowanie tablic o wielu wymiarach. Jeśli będą ci potrzebne, z pewnością poradzisz sobie z ich deklaracją i używaniem.

tablice
wielowymiarowe

8.2. Strumienie, czyli wczytujemy i wyprowadzamy informacje

W programach zdarza się, że dane muszą być wprowadzone przez użytkownika, a wynik, na przykład obliczeń, ma zostać wyświetlony na ekranie. Jak można te czynności zrealizować w języku C++? W tym celu wykorzystuje się strumieniowanie, czyli kierowanie strumienia danych w odpowiednim kierunku. Do ich określenia wykorzystuje się słowa z języka angielskiego `out` wskazujące na zewnątrz oraz `in` – do środka. Jeśli dodasz do nich na początku np. literę `c` kojarzącą się z nazwą języka C++, otrzymasz nazwy strumieni: wejściowego `cin` i wyjściowego `cout`. Dodatkowo kierunki przepływu strumienia wskazują operatory `<<` dla strumienia na zewnątrz (wyprowadzanie danych) oraz `>>` dla strumienia do wewnątrz (wczytywanie danych). Wczytywanie i wyprowadzanie danych może dotyczyć zmiennych różnego typu.

strumienie

Przykład

Układamy program do mnożenia trzech liczb całkowitych. Przeanalizuj fragment kodu wprowadzania danych i strumieniowania wyniku (rys. 8.2.). Znaki, które mają pojawić się na ekranie, należy umieścić w cudzysłowie (na

rysunku niebieskie). Wartości zmiennych są reprezentowane przez ich nazwy (na rysunku czarne). Słowo `endl` oznacza koniec linii, czyli następny strumień danych będzie wyświetlany w nowej linii. Zwróć uwagę, że poszczególne strumieniowane dane są poprzedzone operatorem kierunku.

```
cout << "wprowadź liczbę a " << endl;
cin >> a;
cout << "wprowadź liczbę b " << endl;
cin >> b;
cout << "wprowadź liczbę c " << endl;
cin >> c;
```

Rys. 8.2. Fragment programu strumieniujący dane

Zastanów się i odpowiedz, co będzie efektem działania tego fragmentu kodu programu. Czy twoje przypuszczenia są zgodne z rysunkiem 8.3.?

```
wprowadź liczbę a
12
wprowadź liczbę b
13
wprowadź liczbę c
15
```

Rys. 8.3. Efekt działania fragmentu kodu programu wprowadzającego wartości do trzech zmiennych

8.3. Nie tylko zmienne mają nazwy, czyli budowa i rola funkcji w programie

funkcja

Jeśli takie same działania są wykonywane w wielu miejscach programu, można utworzyć funkcję i do niej się odwoływać. Po pierwsze, nie będzie trzeba wklejać tego samego kodu w każdym miejscu, w którym ma być wykonany. Po drugie, kod staje się krótszy. Po trzecie, wszelkie zmiany i poprawki są wykonywane jedynie w kodzie funkcji. Funkcja może przetwarzać argumenty – dane przekazane przez program. Niektóre funkcje nie potrzebują takich argumentów, na przykład funkcja drukująca stałe komunikaty. Dzięki funkcjom można więc wydzielić w kodzie bloki odpowiedzialne za poszczególne zadania.

argumenty

Cechy funkcji:

- ma nazwę, przez którą jest wywoływana;
- może przetwarzać argumenty wejściowe, jeśli tak przewidziano w programie;
- może zwracać przetworzone dane (wyniki działania funkcji).

Przykład 5.

W pewnym programie wielokrotnie używa się obliczania średniej z trzech liczb *a*, *b* oraz *c*. Opracowano więc funkcję, która dokonuje obliczeń. Przed nazwą funkcji podano typ wartości, jaką funkcja ma zwrócić. W tym przypadku float,



ponieważ średnia może nie być całkowita. Tworzenie funkcji rozpoczyna się podobnie, jak deklarowanie zmiennej tablicowej jednowymiarowej. W miejscu `[]` pojawił się jednak `()`. Kod funkcji zwany **ciałem funkcji** zaczyna się i kończy nawiasem klamrowym `{}`.

Zmienne utworzone w funkcji to zmienne tymczasowe. Oznacza to, że po zakończeniu działania funkcji zmienne nie pamiętają już swojego stanu. Za każdym wywołaniem funkcji wartości są im nadawane na nowo. Przypomnij sobie, na czym polega wykorzystanie kalkulatora. Jeśli mnożysz liczby, to kolejno je wprowadzasz i po wciśnięciu „=” otrzymujesz wynik.

zmienne

Podobnie przekazywane do funkcji są parametry. Przeanalizuj poniższy przykład (rys. 8.4.).

```

1 #include <iostream>
2 using namespace std;
3 int a,b,c;
4
5 int iloczyn (int d, int e, int f)
6 {
7     return (d*e*f);
8 }
9
10 int main()
11 {
12     cout << "wprowadź liczbę a " << endl;
13     cin >> a;
14     cout << "wprowadź liczbę b " << endl;
15     cin >> b;
16     cout << "wprowadź liczbę c " << endl;
17     cin >> c;
18     cout << endl << a << "*" << b << "*" << c << "=" << iloczyn(a,b,c) << endl;
19     return 0;
20 }

```

Definicja zmiennych globalnych

Ciało funkcji z określeniem parametrów funkcji

Główna funkcja programu

Rys. 8.4. Przykład zastosowania funkcji ze zmiennymi lokalnymi

W ciele funkcji zadeklarowano parametry lokalne `d`, `e` oraz `f` i określono, że dane w nich przechowywane będą typu `int` (liczby całkowite). Funkcja zwróci iloczyn `d*e*f`. To oznacza, że nazwa funkcji użyta w programie przyjmie taką wartość. Należy jednak przekazać jej trzy argumenty. Mogą to być zmienne lub konkretne liczby. Program (rys. 8.4.) prosi o wprowadzenie trzech liczb i zapisuje je w zmiennych `a`, `b` oraz `c`. Następnie drukuje ich iloczyn. Oblicza go, wywołując funkcję i przekazując jej wprowadzone dane. Wewnątrz funkcji parametr `d` przyjmuje aktualną wartość zmiennej `a`; i analogicznie: `e` przyjmuje wartość `b` oraz `f` wartość `c`.

Zauważ, że ten rozdział nie wyczerpuje zagadnień związanych z budową i wykorzystywaniem funkcji. Są one opisane w literaturze dla programistów. Jednak zdobyta wiedza wystarczy ci do układania prostych programów z użyciem funkcji.

instrukcje warunkowe

8.4. Zagnieżdżanie instrukcji, czyli pętla w pętli

Znasz już instrukcje warunkowe i pętle. W niektórych programach istnieje konieczność zagnieżdżenia pętli w pętli.

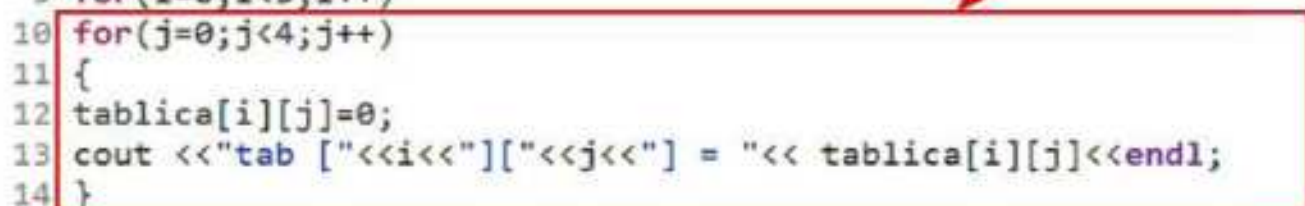
Przykład 6.

instrukcja pętli for

Twój program ma utworzyć tablicę dwuwymiarową `tablica[4][3]` wypełnioną zerami. Potrzebujesz dwóch zmiennych całkowitych do sterowania pętlami. Należy wykorzystać instrukcję pętli `for` zagnieżdżoną w pętli `for`.

```
1 #include <iostream>
2 using namespace std;
3
4 int i,j;
5 int tablica[4][3];
6
7 int main()
8 {
9     for(i=0;i<4;i++)
10        for(j=0;j<3;j++)
11        {
12            tablica[i][j]=0;
13            cout <<"tab ["<<i<<"]["<<j<<"] = "<< tablica[i][j]<<endl;
14        }
15 }
```

Pętla zagnieżdżona



Rys. 8.5. Przykładowy program wypełniający tablicę wartością 0

```
tab [0][0] = 0
tab [0][1] = 0
tab [0][2] = 0
tab [0][3] = 0
tab [1][0] = 0
tab [1][1] = 0
tab [1][2] = 0
tab [1][3] = 0
tab [2][0] = 0
tab [2][1] = 0
tab [2][2] = 0
tab [2][3] = 0
tab [3][0] = 0
tab [3][1] = 0
tab [3][2] = 0
tab [3][3] = 0
tab [4][0] = 0
tab [4][1] = 0
tab [4][2] = 0
tab [4][3] = 0
```

Wartości zmiennych *i* oraz *j* są indeksami zmiennej tablicowej. Zmieniają się zgodnie z zasadami działania pętli `for`, dlatego przyjmą wszystkie możliwe wartości indeksów zmiennej.

Na tej samej zasadzie można zagnieżdżać instrukcje warunkowe.

Rys. 8.6. Efekt działania programu wypełniającego tablicę



ZADANIA DO ROZWIĄZANIA

1. Naskicuj w zeszycie, jak będą wyglądały tablice trójwymiarowa i czterowymiarowa. Udowodnij, że twoje rysunki rzeczywiście są modelami takich tablic.
2. Zmodyfikuj program z rysunku 8.4. w taki sposób, by obliczał sumę kwadratów wprowadzonych liczb. Zweryfikuj wynik działania programu.
3. Zmodyfikuj program wpisujący zera do tablicy dwuwymiarowej w taki sposób, by zamiast zer wpisywał kolejne numery kolumn tablicy. Uzasadnij swoje rozwiązanie.
4. Co będzie wynikiem działania programu z rysunku 8.5., gdy zamieni się miejscami w pętlach zmienne pomocnicze *i* oraz *j*? Opracuj wnioski i udowodnij swoją tezę.
- 5*. Na podstawie programu z rysunku 8.4. ulóż program, który na wprowadzonych liczbach *a*, *b* oraz *c*, będzie wykonywał mnożenie, dzielenie lub dodawanie – w zależności od decyzji użytkownika. Uzasadnij użycie funkcji w tym programie.

PODSUMOWANIE LEKCJI

- **Tablice** to zmienne przechowujące więcej niż jedną wartość. Każda z nich jest opisana indeksem. s. 62
- Istnieją **tablice wielowymiarowe**. s. 63
- Strumieniowanie danych **cin** i **cout** umożliwia wprowadzanie i wyprowadzanie wartości zmiennych (**cout** umożliwia także wyprowadzanie znaków, np. komunikatów). s. 63
- Funkcje są fragmentami kodu programu mającymi nazwę. **Wynik działania funkcji** jest zwracany pod jej nazwą. s. 64
- Do funkcji można przekazać dane (**argumenty**). s. 64
- **Zmienne** deklarowane w funkcjach są zmiennymi lokalnymi i nie zachowują wartości po zakończeniu działania funkcji. s. 65
- Pętle **for** i instrukcje **warunkowe** mogą być zagnieżdżone. s. 66

PODSUMOWANIE PRZED SPRAWDZIANEM

4. Przypomnij sobie, czyli podstawy tworzenia algorytmów

- s. 31 • **Algorytm** to prowadzący do przetworzenia danych wejściowych na wyjściowe skończony i uporządkowany opis precyzyjnie sformułowanych czynności.
- s. 31 • **Najczęściej używane sposoby zapisu algorytmu** to: lista kroków, schemat blokowy, pseudokod.
- s. 31 • Specyfikacja algorytmu składa się najczęściej z trzech części: **opisu problemu, opisu danych wejściowych i opisu danych wyjściowych.**
- s. 32 • Do **cech dobrego algorytmu** należą: **precyzyjne zdefiniowanie, skończoność, efektywność.**

5. Narzędzia, czyli jak dobrać i skonfigurować środowisko programistyczne

- s. 36 • **Dane o popularności** języków programowania można znaleźć na stronie TIOBE.
- s. 38 • **C++** jest językiem uniwersalnym. Dzięki swoim właściwościom pozwala na efektywne gospodarowanie zasobami komputera.
- s. 38 • **Eclipse to IDE** zaprojektowane dla programistów posługujących się różnymi językami programowania.

6. C z plusami, czyli uruchamiamy pierwsze programy w C++

- s. 47 • **MinGW** (Minimalist GNU for Windows) jest niezbędne do kompilowania programów w C++ w Eclipse. Pobieranie i instalacja wymaga precyzji działania.
- s. 50 • Tworzenie nowego projektu w Eclipse C++ rozpocznij od wybrania z menu **File** opcji **New**, a następnie **Create a new C++ project**. Pamiętaj o zaznaczeniu kompilatora MinGW.
- s. 52 • **Dyrektywy** oznaczone **#** są wykonywane przed kompilacją i powodują dołączenie odpowiednich bibliotek (zbiorów funkcji). Bez **bibliotek** nie ma programów w C++.

7. C z plusem, czyli podstawy programowania

- s. 54 • **Kompilacja** polega na zamianie kodu źródłowego programu na język zrozumiały przez komputer.
- s. 55 • **Operatory** arytmetyczne, relacyjne i logiczne pozwalają na zapisywanie działań matematycznych i logicznych.
- s. 59 • **Instrukcje sterujące** pozwalają na realizację rozgałęzień (bloków warunkowych) algorytmów.

8. C z plusem, czyli podstaw programowania ciąg dalszy

- s. 63 • **Zmienne tablicowe** pozwalają przechowywać uporządkowane dane.
- s. 64 • **Funkcje** są fragmentami kodu programu mającymi nazwę. Mogą być wywoływane w programie dzięki użyciu ich nazw. Można do nich przekazywać wartości zmiennych.
- s. 65 • **Zmienne** mogą być deklarowane dla różnych typów danych.
- s. 66 • **Pętle i instrukcje warunkowe** można zagnieżdżać.

III. Programowanie i algorytmy

Sito Eratostenesa

Liczby pierwsze

Program na podstawie algorytmu

Algorytmy Euklidesa

NWW i NWD w programie komputerowym

Dodawanie ułamków

Szyfrowanie

Tablica kodów ASCII

9. Przez 1 i siebie, czyli jak zbadać, czy dana liczba jest liczbą pierwszą

NA TEJ LEKCJI:

- dowiesz się, czym jest sito Eratostenesa;
- nauczysz się sprawdzać, czy liczba jest pierwszą.

Matematycy badający liczby odkryli, że niektóre są podzielne jedynie przez 1 lub samą siebie. Nazwali je pierwszymi, a pozostałe złożonymi. Dlaczego? Otóż liczby pierwsze mają zadziwiającą cechę. Każdą liczbę naturalną (oprócz 1) można przedstawić za pomocą iloczynu liczb pierwszych. Ten proces nazwano rozkładem na czynniki pierwsze. Jak sprawdzić, czy dana liczba jest liczbą pierwszą? Okazuje się, że nie jest to takie proste.

9.1. Starożytni znaleźli sposób, czyli sito Eratostenesa wyszukuje liczby pierwsze z przedziału

Podobnie jak nie można wskazać największej z liczb, również nie ma możliwości wskazania największej liczby pierwszej. Uczni starają się wyznaczyć coraz to większe, ale okazuje się, że im dalej na osi liczbowej, tym występują one rzadziej. Obecnie największą odnaną jest 2 do potęgi 74207281 pomniejszona o 1. Zapisanie jej za pomocą cyfr zajęłoby resztę objętości podręcznika, ponieważ zawiera ona 22 338 618 cyfr. Sprawdzenie, czy liczba tego rzędu wielkości jest liczbą pierwszą, stanowi wielkie wyzwanie nawet dla najszybszych komputerów na świecie.

Znane algorytmy radzą sobie sprawnie do pewnego zakresu. Jedną z pierwszych metod została opracowana ponad 2000 lat temu przez Greka Eratostenesa. Nazwano ją sitem Eratostenesa. Oto algorytm w postaci listy kroków.

sito Eratostenesa

Sito Eratostenesa

1. Określ liczbę naturalną n z przedziału od 2 do n , czyli zakres badanych liczb.
2. Usuń ze zbioru wszystkie liczby większe od 2 podzielne przez 2.
3. Wybierz najmniejszą z niewykreślonych liczb i usuń ze zbioru wszystkie liczby przez nią podzielne, oprócz niej samej.
4. Wykonuj punkt 3. do wyczerpania zasobu liczb z założonego przedziału od 2 do n .

Tab. 9.1. Przykład zastosowania sita Eratostenesa dla $n=16$

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

n niewykreślone liczby z zakresu od 2 do n , gdzie $n=16$

n liczby wykreślone

n liczby pierwsze

Powyższy algorytm jest skuteczny w odnajdywaniu kolejnych liczb pierwszych. Zrealizowany za pomocą programu komputerowego wymagałby zarezerwowania znacznej ilości pamięci do ich przechowywania. Ilość ta nie byłaby stała, lecz zależna od zakresu badanych liczb.

9.2. Pierwsza czy złożona, czyli algorytm badania liczby

Liczby pierwsze odgrywają bardzo ważną rolę w informatyce.

Bardzo duże liczby pierwsze są wykorzystywane w niektórych trudnych do złamania szyfrach z kluczem publicznym, jak RSA (algorytm Rivesta-Shamira-Adlemana) wykorzystywanym między innymi do szyfrowania operacji bankowych. O kluczach publicznych przeczytasz w rozdziale o szyfrowaniu.

Jedną z metod badania liczby naturalnej wykorzystuje właściwość, że liczba pierwsza nie ma dzielników (liczb naturalnych) w przedziale od 2 do pierwiastka kwadratowego z siebie $\langle 2, [\sqrt{n}] \rangle$. Jeśli znajdzie się dzielnik, dla którego reszta z dzielenia będzie miała wartość 0, to liczba ta jest **złożona**. Na tym obliczenia można zakończyć. Inna metoda polega na badaniu reszty z dzielenia liczby n przez kolejne liczby naturalne z zakresu od 2 do n (rys. 9.1.).

wielkie liczby
pierwsze

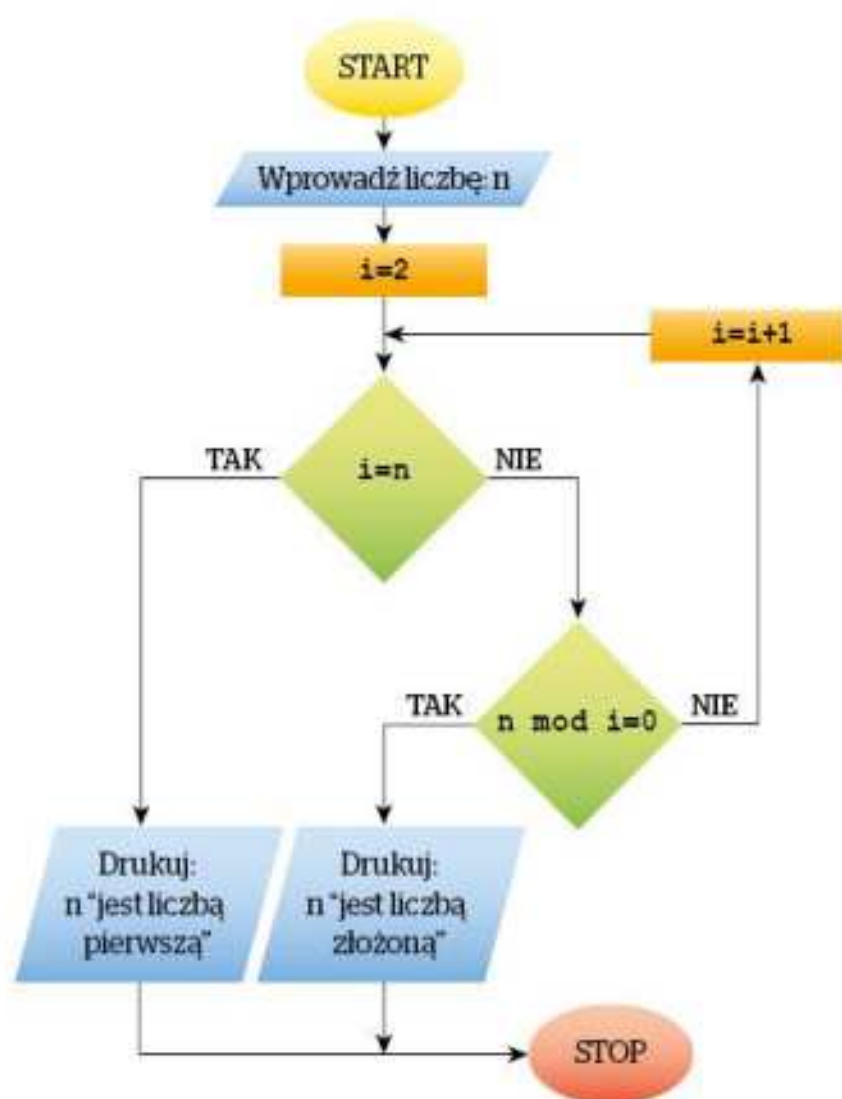
liczby pierwsze
w informatyce

```

length; c.unique = b.length;
c)); } return b.length;
, b), b = b.replace(
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY



Rys. 9.1. Schemat blokowy algorytmu wykrywającego liczby pierwsze

Uwaga!

Operator mod użyty w schemacie blokowym oblicza resztę z dzielenia. W języku C++ zostanie użyty operator %.

Prześledźmy algorytm z rysunku 9.1. i sprawdźmy jego działanie na przykładzie liczb 6 (tab. 9.2.) i 5 (tab. 9.3.).

Tab. 9.2. Test algorytmu wykrywającego liczby pierwsze dla liczby 6

Krok	Działanie	Wynik
1.	Wczytaj n	$n = 6$
2.	Zmienna i przyjmuje wartość 2	$i = 2$
3.	Badaj warunek $i = n$	$2 \neq 6$ NIE
4.	Zbadaj resztę z dzielenia n przez i	$6/2=3$ reszta 0 TAK
5.	Drukuj komunikat	6 jest liczbą złożoną
6.	STOP	Koniec algorytmu

Tab. 9.3. Test algorytmu wykrywającego liczby pierwsze dla liczby 5

Krok	Działanie	Wynik
1.	Wczytaj n	$n = 5$
2.	Zmienna i przyjmuje wartość 2	$i = 2$
3.	Badaj warunek $i = n$	$2 \neq 5$ NIE
4.	Czy reszta z dzielenia n przez $i = 0$	$5/2=2$ reszta 1 NIE
5.	Zwiększ wartość zmiennej i o 1	$i = 3$
6.	Badaj warunek $i = n$	$3 \neq 5$ NIE
7.	Czy reszta z dzielenia n przez $i = 0$	$5/3=1$ reszta 2 NIE
8.	Zwiększ wartość zmiennej i o 1	$i = 4$
9.	Badaj warunek $i = n$	$4 \neq 5$ NIE
10.	Czy reszta z dzielenia n przez $i = 0$	$5/4=1$ reszta 1 NIE
11.	Zwiększ wartość zmiennej i o 1	$i = 5$
12.	Badaj warunek $i = n$	$5 = 5$ TAK
13.	Drukuj komunikat	5 jest liczbą pierwszą
14.	STOP	Koniec algorytmu

Ten sam algorytm można zapisać w postaci listy kroków.

Specyfikacja i lista kroków

Tab. 9.4. Algorytm wykrywający liczby pierwsze w postaci listy kroków

Specyfikacja problemu	Zbadać, czy dana liczba naturalna większa od 1 jest liczbą pierwszą i wydrukować odpowiedni komunikat.
Dane wejściowe	n zmienna zawierająca badaną liczbę
Dane wyjściowe	komunikat o tym, czy n jest liczbą pierwszą czy złożoną
Zmienne	n zmienna zawierająca badaną liczbę i zmienna pomocnicza
Lista kroków	1. Wczytanie wartości zmiennej n 2. Ustalenie wartości zmiennej i=2 3. Jeśli i=n drukuj komunikat n jest liczbą pierwszą zakończ algorytm 4. Jeśli i≠n sprawdź resztę z dzielenia n mod i (n%i w języku c++) dla reszty 0 drukuj komunikat n jest liczbą złożoną w przeciwnym przypadku zwiększ zmienną pomocniczą o 1 i:=i+1 5. Ponownie badaj, czy i=n Jeśli NIE, powtarzaj punkt 4, w przeciwnym przypadku drukuj komunikat n jest liczbą pierwszą zakończ algorytm

```
a.replace(' ' + '='  
length; c.unique = b.len  
c)); } return b.length  
, b), b = b.rep  
np_array.length; a++) {  
use_class = use_array(b
```

ZADANIA DO ROZWIĄZANIA

1. Przedstaw właściwości liczb pierwszych i ich zastosowanie. Przejrzyj wiarygodne źródła, w tym także strony internetowe, aby znaleźć informacje o badaniach nad liczbami pierwszymi i ich zastosowaniach, np. nad opracowaniem wzorów na ich odnajdywanie.
2. Sprawdź działanie schematu blokowego algorytmu dla liczb 3, 7 i 11. Policz, ile razy dla każdej z nich zostanie wykonana pętla, w której badana jest reszta z dzielenia. Postaraj się ułożyć wzór, który dla liczby pierwszej poda prawidłowo liczbę przejścia przez tę pętlę.
3. Przetestuj algorytm podany w postaci listy kroków dla dwóch wybranych liczb pierwszych i dwóch złożonych. Podaj liczbę kroków dla każdej z nich.
- 4*. Wyszukaj w sieci informacje na temat modyfikacji poznanego algorytmu ograniczających liczbę kroków prowadzących do celu. Zapoznaj się z nimi i przedstaw, na czym one polegają. Uzasadnij ich skuteczność.
- 5*. Jedną z metod badania liczby naturalnej wykorzystuje właściwość, że liczba pierwsza nie ma dzielników w przedziale od 2 do pierwiastka kwadratowego z siebie. Zmodyfikuj algorytm omówiony w rozdziale tak, by wykorzystywał tę właściwość.

PODSUMOWANIE LEKCJI

s. 70

- **Metoda sito Eratostenesa** powstała w starożytności; pozwala znaleźć liczby pierwsze spośród liczb z zakresu od 2 do n .

s. 71

- **Wielkie liczby pierwsze** odgrywają dużą rolę w kryptologii i szyfrowaniu danych komputerowych. Dzięki nim można tworzyć bardzo trudne do przełamania szyfry z kluczem publicznym.

s. 71

- **Liczby 0 i 1** nie są zaliczane do liczb pierwszych. Zbiór liczb pierwszych jest **nieograniczony**. Największe odkryte liczby pierwsze można zapisać z użyciem ponad 22 milionów cyfr.

10. Która pierwsza, czyli jak program bada liczby

NA TEJ LEKCJI:

- nauczysz się układać program na podstawie algorytmu;
- dowiesz się, jak zmodyfikować algorytm.

Znasz już jeden z algorytmów, dzięki któremu można zbadać, czy liczba jest liczbą pierwszą. Wiesz, jak działa, i sprawdziłeś/sprawdziłaś go na przykładach. Czas na ułożenie na jego podstawie programu w języku programowania. Przypomnij sobie zagadnienia z działu II. Opisane tam narzędzia będą ci potrzebne.

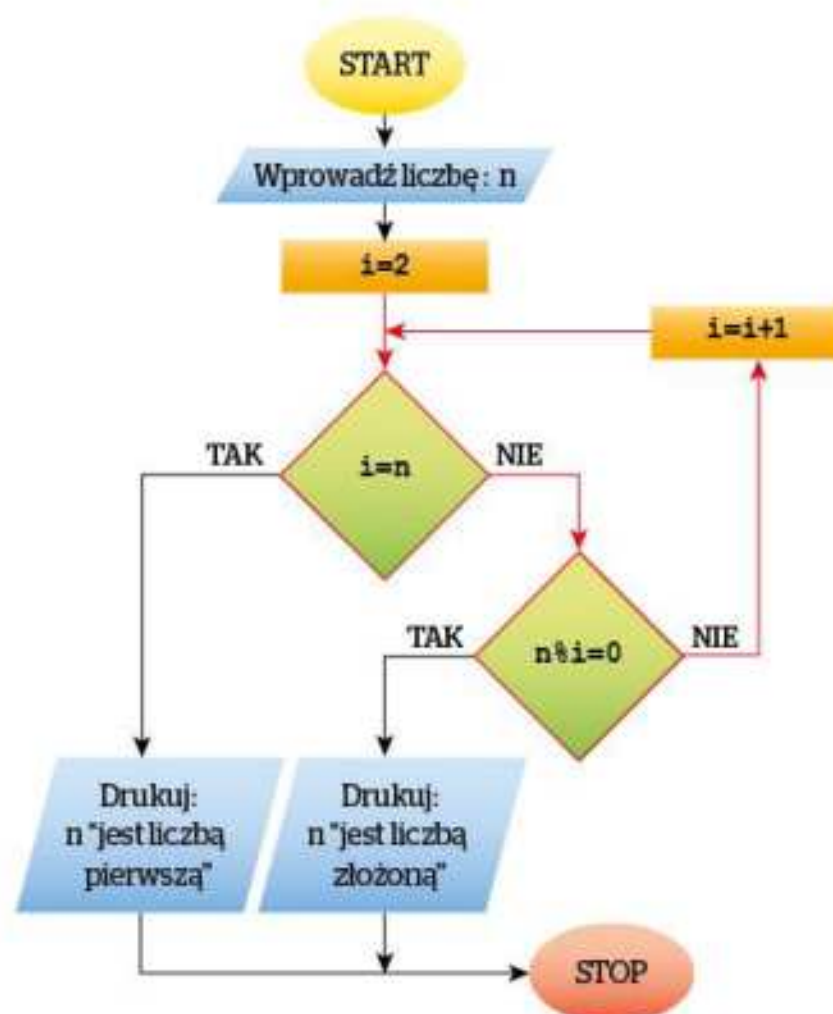
10.1. Ile razy powtórzyć, czyli pętla w programie badania liczb

Przeanalizuj algorytm, według którego będziesz układać program. Przyjrzyj się szczególnie pętli zawierającej oba warunki w przypadku ich wystąpienia (kierunki NIE; rys. 10.1.).

program
na podstawie
algorytmu

Jeśli badaną liczbą będzie na przykład 7, to ta pętla będzie „obchodzona” 5 razy, kolejno dla wartości zmiennej i od 2 do 6. Dla większych liczb będzie tych „obejść” odpowiednio więcej ($n-2$). W programie musi więc być polecenie, które zrealizuje tę pętlę. Spróbujmy zapisać słowami, co dzieje się, gdy warunek $i=n$ nie jest spełniony. Można to zrobić na kilka sposobów, na przykład tak:

Dla każdej kolejnej, naturalnej wartości i począwszy od 2 (to wynika z pełnego algorytmu), aż do liczby o 1 mniejszej od badanej, sprawdzaj, czy reszta z dzielenia n przez i jest zerem. Jeśli tak, opuść pętlę.



Rys. 10.1. Algorytm z zaznaczoną pętlą obliczania reszty z dzielenia

```

length; c.unique = b.length;
c)); } return b.length;
, b), b = b.rep
np_array.length; a++) {
use_class = use_array(b

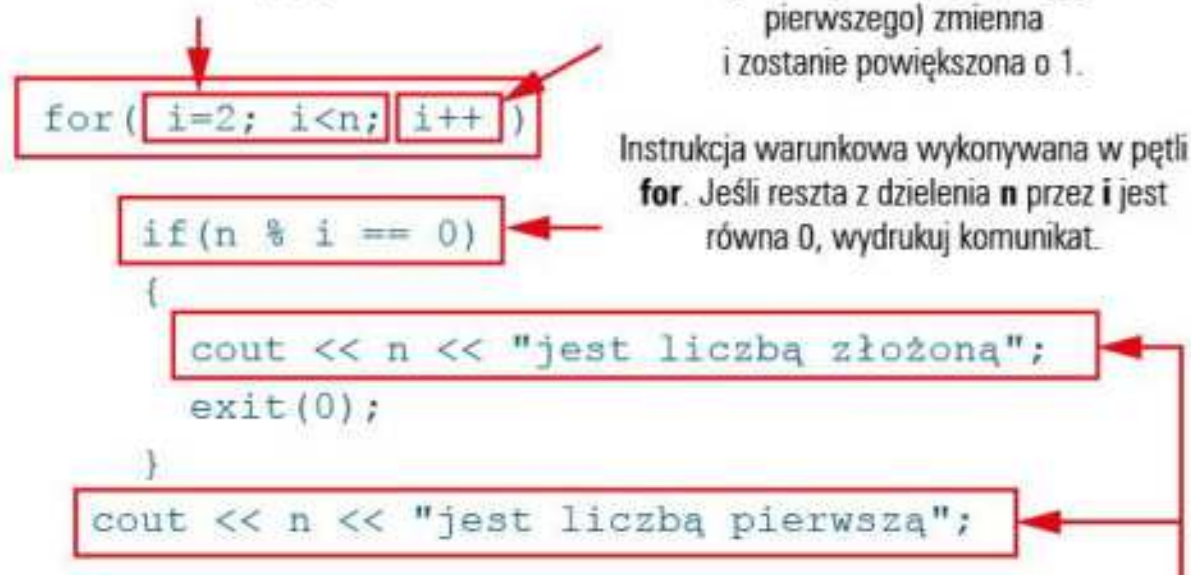
```

III. PROGRAMOWANIE I ALGORYTMY

Czy podobnie można to zapisać w języku C++? Nie identycznie, ponieważ C++, jak każdy język, ma swoją składnię. Jednak podobieństwa będą widoczne (rys. 10.2.).

Począwszy $i=2$, dopóki i będzie mniejsze od n , wykonuj następną instrukcję. Gdy warunek $i < n$ nie zostanie spełniony, zakończ pętlę.

$i++$ oznacza, że przed rozpoczęciem każdego „okrążenia” pętli (z wyjątkiem pierwszego) zmienna i zostanie powiększona o 1.



Instrukcja warunkowa wykonywana w pętli **for**. Jeśli reszta z dzielenia n przez i jest równa 0, wydrukuj komunikat.

Na ekranie zostanie wyświetlona wartość zmiennej n , czyli liczba badana, oraz komunikat ujęty w " ".

Rys. 10.2. Fragment programu w języku C++ realizujący pętlę

Ponieważ instrukcja warunkowa `if` jest jedyną instrukcją w pętli `for`, będzie wykonywana dopóki parametry pętli na to pozwolą. Czyli:

$i < n$.

Instrukcja zakończenia programu:

```
exit(0);
```

nie tylko kończy program, lecz także uniemożliwia wyświetlenie dwóch sprzecznych komunikatów w przypadku, gdy liczba okaże się złożona. Po prostu instrukcja:

```
cout << n << " jest liczbą pierwszą";
```

nie zostanie już wykonana.

Polecenia z rysunku 10.2. nie stanowią jeszcze programu. Są jedynie zapisem fragmentu algorytmu z użyciem poleceń i składni języka C++. W rzeczywistości działający program musi zawierać także inne polecenia (rys. 10.3.). Jak widać, omawiany wcześniej fragment stał się częścią programu ułożonego w języku C++.

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int n, p;
6     cout << "Podaj liczbę naturalną: ";
7     cin >> n;
8     for (p=2; p<n; p++)
9         if(n % p == 0)
10        {
11            cout << n << " jest liczbą złożoną";
12            exit(0);
13        }
14        cout << n << " jest liczbą pierwszą";
15        exit(0);
16    }
17

```

Rys. 10.3. Program sprawdzający, czy liczba n jest liczbą pierwszą

1 linia dołącza do programu podstawową bibliotekę obsługującą operacje wejścia–wyjścia. Dzięki niej można między innymi wprowadzać dane na przykład z klawiatury i wyświetlać dane na ekranie, jak w przykładzie.

Linia 2 to instrukcja, która ustala tak zwaną przestrzeń nazw. Dzięki temu nie trzeba pisać nazwy przestrzeni (`std`) przed każdym identyfikatorem (nazwą) z biblioteki `iostream`. Na przykład bez tej instrukcji linia 7 musiałaby mieć postać `using std::cin >> n;` zamiast `cin >> n`.

Linia 3 to początek głównej funkcji programu kończącej się w **linii 16**.

Linia 5 zawiera deklaracje zmiennych. W tym przypadku informujemy kompilator, że będziemy używać dwóch zmiennych o nazwach `n` oraz `p`, które będą liczbami całkowitymi. Świadczy o tym słowo `int` (typ zmiennej całkowitej – integer).

Linia 6 to polecenie wyświetlenia na ekranie napisu (wysłania strumienia na konsolę). Znaki `<<` symbolizują kierunek wysyłania tych danych – w tym przypadku napis ujęty w " " na wyjście (na ekran – konsolę). Każdy prawidłowo sformułowany komunikat ułatwi użytkownikowi korzystanie z programu.

kierunek
strumienia danych

Linia 7 zawiera instrukcję, która zapisze pod nazwą zmiennej wartość wprowadzoną z klawiatury. Ponieważ `n` było deklarowane jako `int`, użytkownik powinien wprowadzić liczbę całkowitą. Zatwierdzenie wpisu następuje po wciśnięciu Enter. Ponieważ badamy jedynie liczby naturalne >1 , informujemy o tym w komunikacie wyświetlanym po wykonaniu polecenia z linii 6. Można także wprowadzić do programu badanie tego warunku.

Linie od 8 do 15 zajmuje właściwe badanie liczby i wyświetlanie jego wyniku, które zostało opisane powyżej.

Linia 16 zawiera nawias `}` kończący główną funkcję `main()` programu.

Polecenie `exit (0)` oznacza zakończenie programu i usunięcie z pamięci jego danych.

```

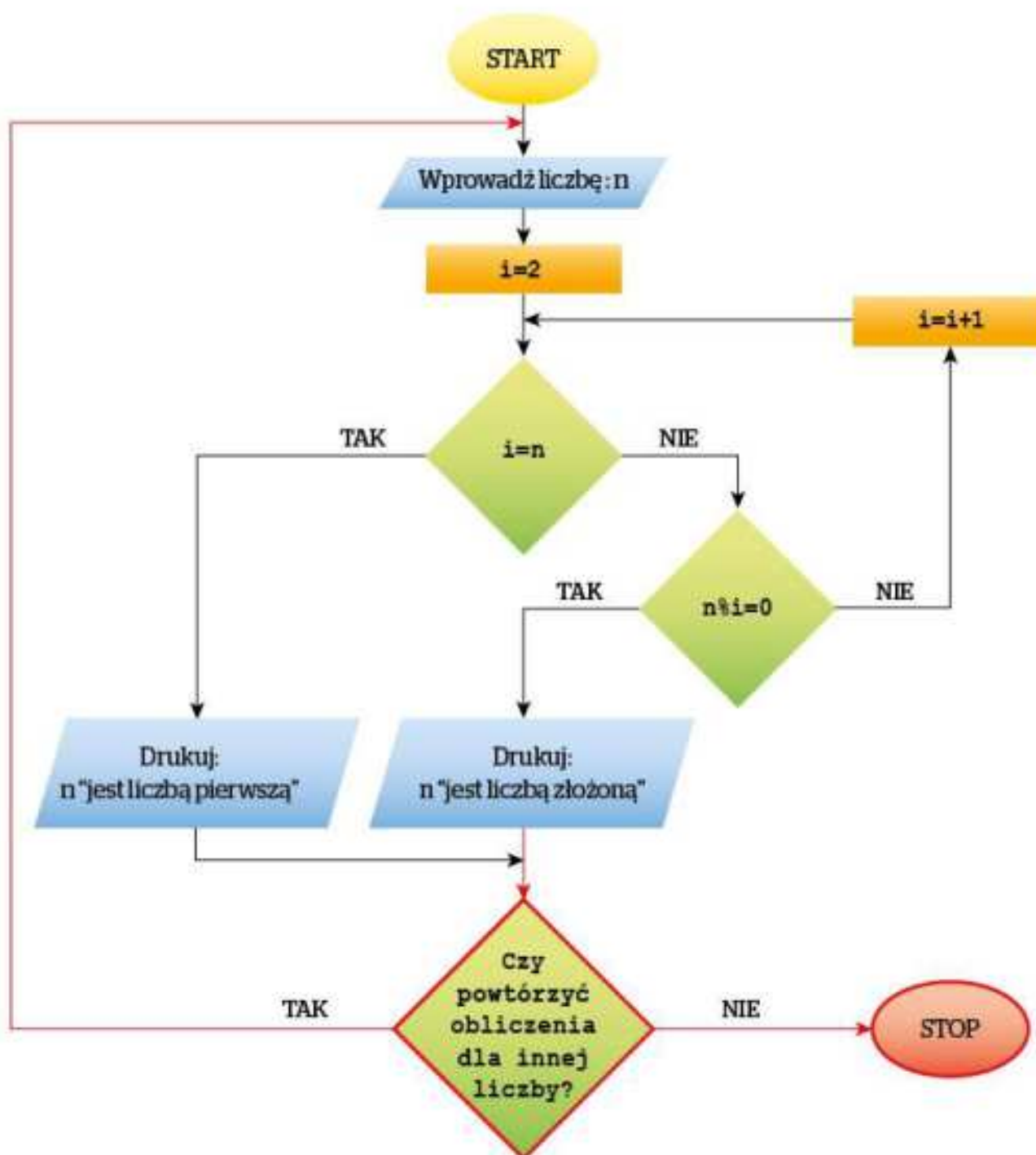
length; c.unique = b.length;
c)); } return b.length;
, b), b = b.rep
np_array.length; a++) {
use_class = use_array(b

```

modyfikowanie
algorytmu
i programu

10.2. Badamy następną, czyli modyfikujemy program

Ułożony program, mimo że działa prawidłowo, ma pewne wady – na przykład testuje tylko jedną liczbę i kończy pracę. Aby przetestować następną, należy ponownie go uruchomić. Zmodyfikujemy algorytm w taki sposób, by możliwe było testowanie następnych liczb lub zakończenie działania programu (rys. 10.4.).



Rys. 10.4. Algorytm umożliwiający sprawdzanie dowolnej ilości liczb

Modyfikacja nie jest skomplikowana i polega na dodaniu badania jednego warunku. Czy opracowanie na jego podstawie programu będzie równie łatwe? Po pierwsze należy użyć nowej zmiennej, na przykład *k*, która będzie badana w warunku. Po drugie należy znaleźć odpowiednie miejsce na umieszczenie w programie instrukcji warunkowej. Zastosujemy także nieco inną realizację algorytmu badania liczb. Użyjemy pętli *while* i *do while* (rys. 10.5.). Do dzieła.

```
use class = use_array(
  inp_array, length=a+
  )
  ( , d), d = d.re
  [c]] } return d.length
  length; c.unique = d.le
  a = a.replace(/+re
```

```

1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int i=2, n, k;
6
7     do
8     {
9         cout <<endl<<"podać liczbę naturalną większą od 1 ";
10        cin >> n;
11
12        while (n%i!=0 && i<=n)
13        {
14            i++;
15            if (i*i<=n)
16                cout <<n<< " jest liczbą złożoną"<<endl;
17            else
18                cout <<n<< " jest liczbą pierwszą"<<endl;
19        }
20        cout << "Wprowadzasz nową liczbę? 1-TAK, dowolny klawisz-NIE ";
21        cin >> k;
22        i=2;
23    }
24    while (k==1);
25
26    cout <<endl<<"koniec programu";
27    exit (0);
28
29 }
30

```

Rys. 10.5. Przykładowy program do badania wielu liczb

W miejscu pętli for z pierwszej realizacji algorytmu, pojawiła się pętla

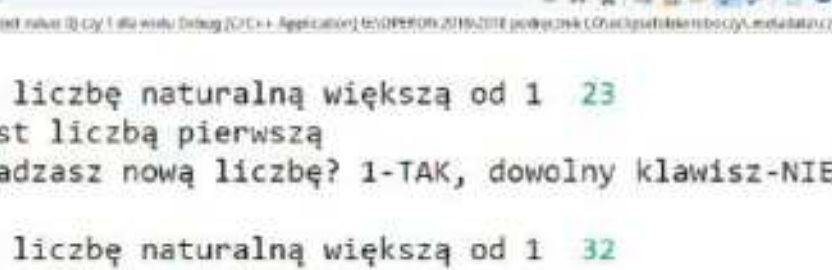
petle

`while (n%i!=0 && i<=n)`, która będzie realizowana, dopóki reszta z dzielenia `n` przez `i` jest różna od 0 (`!=`) oraz `i` jest mniejsza lub równa badanej liczbie.

A co z wprowadzaniem następnych liczb? Organizuje to pętla `do while (k==1);`, która działa nieprzerwanie, gdy `k` jest równe 1 (wartość zmiennej wprowadzona z klawiatury). Oznacza to, że podstawienie pod `k` każdej innej wartości spowoduje opuszczenie pętli.

Efekt działania programu można obserwować w konsoli (rys. 10.6.).

efekt działania programu



```
podaj liczbę naturalną większą od 1 23
23 jest liczbą pierwszą
Wprowadzasz nową liczbę? 1-TAK, dowolny klawisz-NIE 1

podaj liczbę naturalną większą od 1 32
32 jest liczbą złożoną
Wprowadzasz nową liczbę? 1-TAK, dowolny klawisz-NIE 1

podaj liczbę naturalną większą od 1 445
445 jest liczbą złożoną
Wprowadzasz nową liczbę? 1-TAK, dowolny klawisz-NIE 3

koniec programu
```

Rys. 10.6. Efekt działania programu dla przykładowych danych

```
a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length;
, b), b = b.rep
np_array.length; a++) {
use_class = use_array(b
```

III. PROGRAMOWANIE I ALGORYTMY

Algorytmy sprawdzania, czy liczba jest pierwszą, a więc i programy ułożone na ich podstawie mogą się różnić. Zapewne zauważyłeś/zauważyłaś, że program z rysunku 10.6. stosuje inny sposób – zamiast badania warunku $i < n$ bada $i * i \leq n$. To właśnie przykład zastosowania innej metody sprawdzania liczby.

ZADANIA DO ROZWIĄZANIA

1. Zbadaj, czy program z rysunku 10.5. można zmodyfikować tak, by był badany warunek $i \leq n$ zamiast $i * i \leq n$ i nie zmieniło to poprawności działania programu, a przy tym efekt działania programu pozostał bez zmian. Przetestuj rozwiązanie na kilku przykładach. Porównaj wyniki.
2. Zmień komunikaty wyświetlane przez program z zachowaniem poprawności treści, które niosą. Zmodyfikuj sposób wyświetlania, posługując się poleceniem `endl`.
3. Zmodyfikuj algorytm, dodając badanie, czy wprowadzona liczba n jest większa od 1. Jeśli tak nie będzie, program powinien podać odpowiedni komunikat i poprosić o ponowne wpisanie liczby.
4. Ułóż program na podstawie algorytmu z zadania 3. Sformułuj odpowiednie komunikaty i zorganizuj odpowiednie ich wyświetlanie w konsoli.
- 5*. Zmodyfikuj algorytm i ułóż na jego podstawie program, w którym zostaną wyeliminowane wszelkie błędy wprowadzania danych, czyli liczby, które nie są naturalne oraz mniejsze od 1, a także litery i inne znaki.

PODSUMOWANIE LEKCJI

- s. 75 • Na podstawie jednego algorytmu można ułożyć wiele różniących się programów, używając do tego różnych poleceń.
- s. 77 • O kierunku **strumienia danych** świadczą znaki `<<i` i `>>`.
- s. 78 • **Algorytm i program można modyfikować** oraz dodawać im większej funkcjonalności.
- s. 79 • **Pętle** występujące w algorytmie można realizować w programie za pomocą różnych instrukcji.
- s. 79 • Efekty działania programu można obserwować w konsoli **w środowisku Eclipse**, w którym układa się i uruchamia program.

11. Największy i najmniejszy, czyli jak znaleźć NWD i NWW

NA TEJ LEKCJI:

- poznasz algorytm optymalny i nieoptymalny znajdowania NWD;
- dowiesz się, jak ułożyć program, który oblicza NWD i NWW.

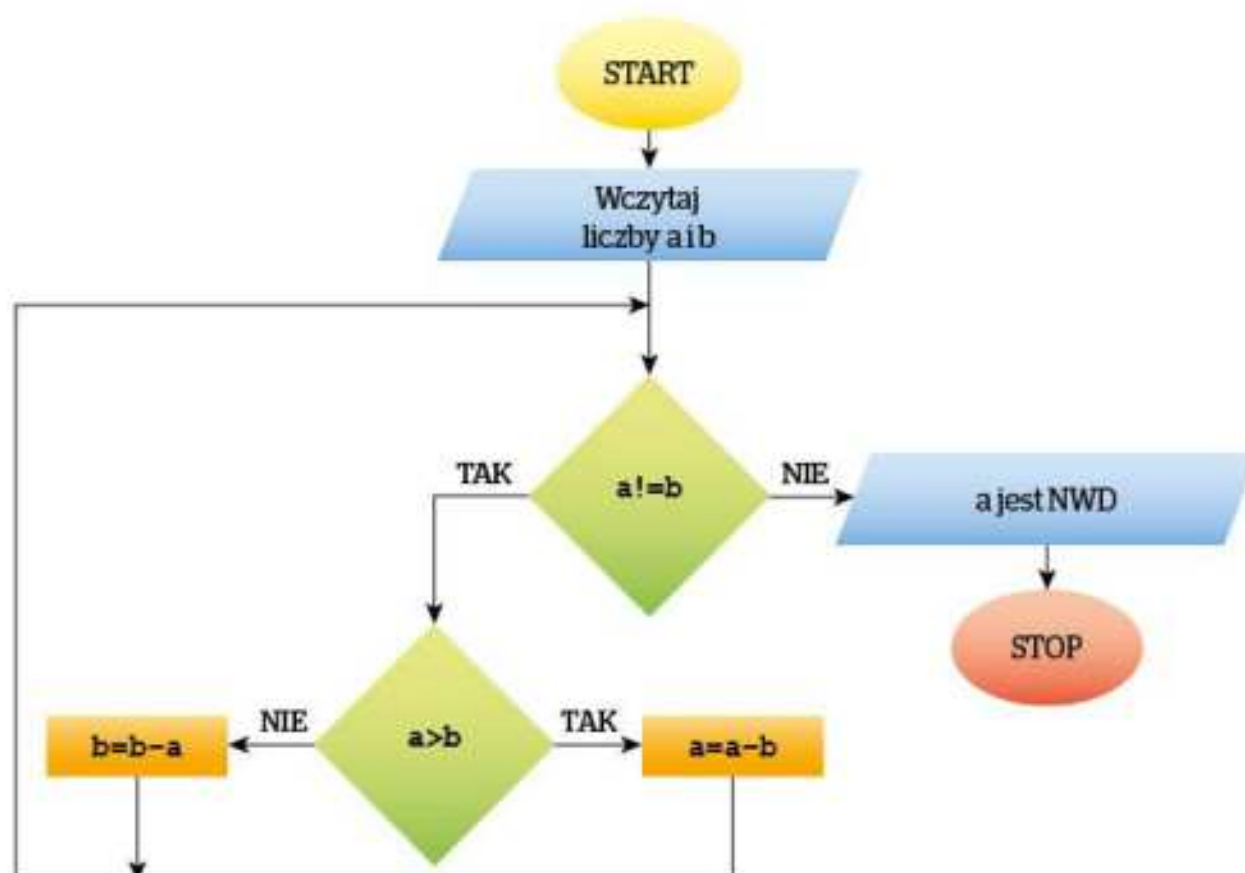
Pamiętasz naukę dodawania i odejmowania ułamków? Należało znaleźć NWW, czyli najmniejszą wspólną wielokrotność mianownika. Czas nauczyć tego twój komputer. Najpierw jednak zmusisz go do obliczenia NWD (największego wspólnego dzielnika), a potem zmodyfikujesz algorytm tak, by znajdował NWW.

11.1. Nieoptymalizowane i zoptymalizowane, czyli algorytmy Euklidesa

Metodę znajdowania NWD przypisuje się greckiemu uczonemu Euklidesowi z Aten. Jest ona skuteczna, ale przy pewnym układzie liczb może być mało wydajna. Oznacza to, że liczba kroków potrzebnych algorytmowi do zakończenia obliczeń może być znaczna i bliska większej z liczb. Dlatego tę metodę nazywamy nieoptymalnym algorytmem Euklidesa lub inaczej metodą odejmowania (rys. 11.1.).

metoda
znajdowania
NWD

algorytm
nieoptymalny



Rys. 11.1. Algorytm nieoptymalizowanej metody Euklidesa

```

length; c.unique = b.length;
c}); } return b.length;
, b), b = b.replace(
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

Prześledźmy działanie algorytmu dla przykładowych liczb. Sprawdź, czy umiesz odczytywać i testować algorytmy. Zakryj kartką wiersze tabeli (tab. 11.1.). Śledź algorytm dla liczb $a = 18$ i $b = 48$. Pierwszy wiersz tabeli jest prosty do odgadnięcia. Wykonuj kolejne kroki algorytmu i sprawdzaj, czy prawidłowo odczytujesz algorytm, odkrywając kolejne wiersze poniższej tabeli.

Tab. 11.1. Realizacja niezoptymalizowanego algorytmu Euklidesa dla liczb 18 i 48

Krok	Działanie	Wynik
1.	Wprowadzenie danych do algorytmu	$a = 18, b = 48$
2.	Sprawdzenie warunku końca algorytmu ($a \neq b$)	$18 \neq 48$ TAK
3.	Porównanie liczb a i b ($a > b$)	$18 > 48$ NIE
4.	Wykonanie odejmowania $b - a$, wynik podstaw pod b	$b = 48 - 18 = 30$
5.	Sprawdzenie warunku końca algorytmu ($a \neq b$)	$18 \neq 30$ TAK
6.	Porównanie liczb a i b ($a > b$)	$18 > 30$ NIE
7.	Wykonanie odejmowania $b - a$, wynik podstaw pod b	$b = 30 - 18 = 12$
8.	Sprawdzenie warunku końca algorytmu ($a \neq b$)	$18 \neq 12$ TAK
9.	Porównanie liczb a i b ($a > b$)	$18 > 12$ TAK
10.	Wykonanie odejmowania $a - b$, wynik podstaw pod a	$a = 18 - 12 = 6$
11.	Sprawdzenie warunku końca algorytmu ($a \neq b$)	$6 \neq 12$ TAK
12.	Porównanie liczb a i b ($a > b$)	$6 > 12$ NIE
13.	Wykonanie odejmowania $b - a$, wynik podstaw pod b	$b = 12 - 6 = 6$
14.	Sprawdzenie warunku końca algorytmu ($a \neq b$)	$6 \neq 6$ NIE
15.	Koniec działania algorytmu	NWD = 6

Algorytm obliczył NWD. Czas, by zrobił to twój komputer. Podobnie jak w poprzednim rozdziale użyjemy środowiska Eclipse i języka C++.

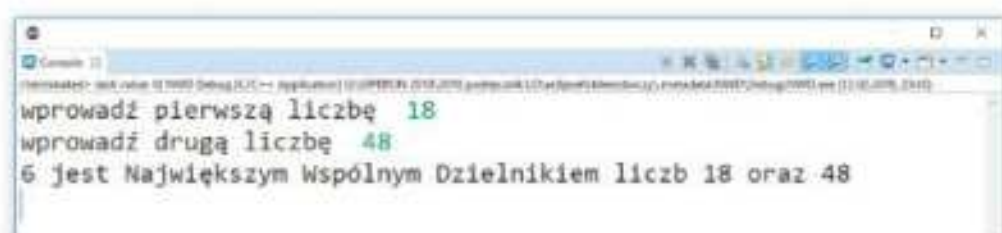
```

1 #include <iostream>
2 using namespace std;
3 int a,b,c,d;
4 int main()
5 {
6     cout << "wprowadź pierwszą liczbę ";
7     cin >> a;
8     c=a;
9     cout << "wprowadź drugą liczbę ";
10    cin >> b;
11    d=b;
12    while(a != b)
13        if(a > b) a = a - b;
14        else b = b - a;
15    cout << a << " jest Największym Wspólnym Dzielnikiem liczb " << c << " oraz " << d << endl;
16    return 0;
17 }

```

Rys. 11.2. Przykładowy program realizujący niezoptymalizowany algorytm Euklidesa

Jeśli uważnie przeczytałeś/przeczytałaś rozdział 10., linie od 1 do 11 powinny być dla ciebie zrozumiałe. W linii 12 zaczyna się realizacja zasadniczej części algorytmu. Zgodnie z nim obliczenia będą prowadzone do stanu, w którym $a = b$. Dlatego w linii 12 została użyta pętla `while (a!=b)`, czyli realizowana tak długo, gdy $a \neq b$. Wewnątrz pętli (od linii 12 do 16) zgodnie z algorytmem jest badany warunek, czy $a > b$ i w zależności od wyniku następują odpowiednie działania arytmetyczne (odejmowanie). Jeśli a i b będą sobie równe, warunek pętli nie zostanie spełniony i wykona się polecenie z linii 15, czyli zostanie wysłany strumień danych do konsoli i wyświetli się wynik obliczeń (rys. 11.3.). Zapewne dostrzegasz, że oprócz zmiennych `a` i `b`, do których wpisuje się na początku dane liczby, zadeklarowano także zmienne całkowite `c` i `d`. Nie ma to nic wspólnego z realizacją algorytmu, ale pozwala na wyświetlenie pełnego komunikatu końcowego. Dlaczego? Przeanalizuj jeszcze raz program. Zauważ, że wartości `a` i `b` zmieniają się w trakcie jego wykonywania, a to oznacza, że gubimy dane wejściowe i ich wyświetlenie w komunikacie jest niemożliwe. Stąd zapamiętanie ich tuż po wprowadzeniu pod inną nazwą zmiennej.



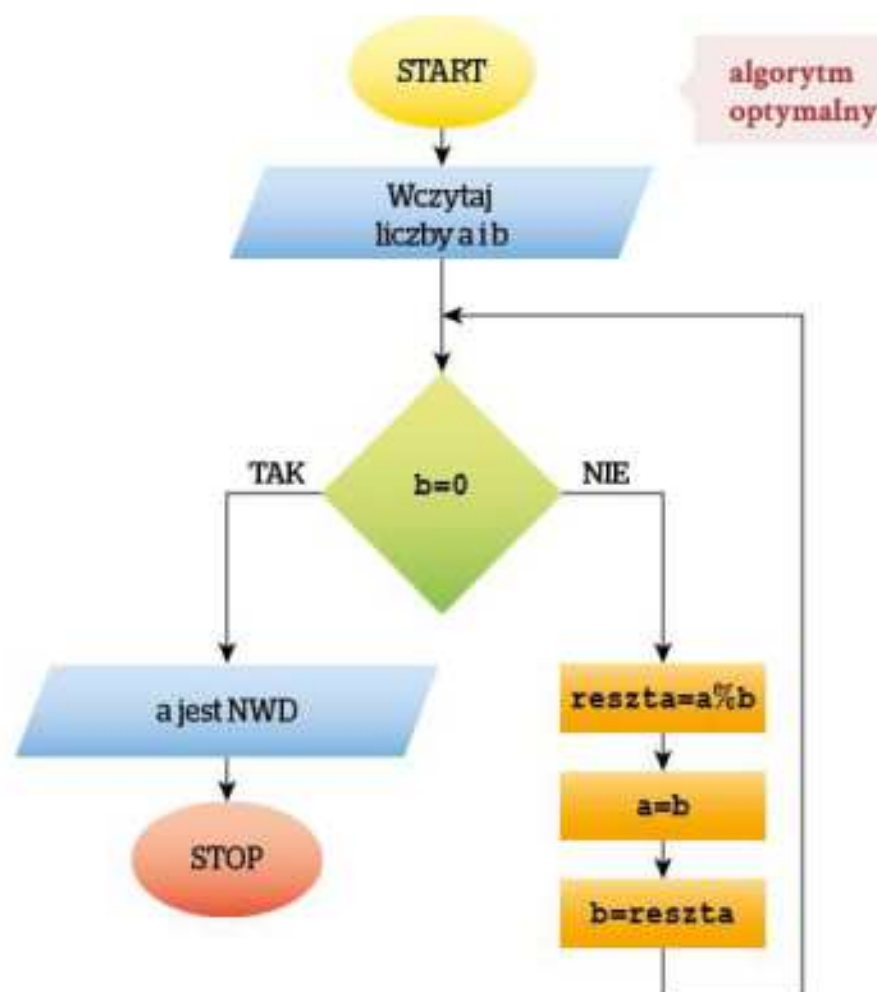
Rys. 11.3. Wynik działania programu dla liczb 18 i 48

Skoro istnieje algorytm niezoptymalizowany, musi istnieć zoptymalizowany, czyli wykonujący obliczenia w mniejszej liczbie kroków. Wykorzystano w nim obliczanie reszty z dzielenia, a więc to samo działanie, które wykorzystaliśmy do badania liczb pierwszych. Metodę zoptymalizowaną algorytmu Euklidesa poznasz, analizując algorytm (rys. 11.4.).

Przypomnienie

Operator % użyty w przykładzie w języku C++ oblicza resztę z dzielenia.

Prześledź działanie algorytmu dla liczb $a = 48$ i $b = 18$ (tab. 11.2.).



Rys. 11.4. Przykładowy algorytm zoptymalizowanej metody Euklidesa

```

a.replace(
length; c.unique = b.le
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

Tab. 11.2. Realizacja zoptymalizowanego algorytmu Euklidesa dla liczb 18 i 48

Krok	Działanie	Wynik
1.	Wprowadzenie danych do algorytmu	$a = 48, b = 18$
4.	Sprawdzenie, czy $b = 0$	$b = 18$ NIE
5.	Obliczanie reszty z dzielenia $a \% b$	$reszta = 48 \% 18 = 12$
6.	Podstawienie $a = b$	$a = 18$
7.	Podstawienie $b = reszta$	$b = 12$
8.	Sprawdzenie, czy $b = 0$	$b = 12$ NIE
9.	Obliczanie reszty z dzielenia $a \% b$	$reszta = 18 \% 12 = 6$
10.	Podstawienie $a = b$	$a = 12$
11.	Podstawienie $b = reszta$	$b = 6$
12.	Sprawdzenie, czy $b = 0$	$b = 6$ NIE
13.	Obliczanie reszty z dzielenia $a \% b$	$reszta = 12 \% 6 = 0$
14.	Podstawienie $a = b$	$a = 6$
15.	Podstawienie $b = reszta$	$b = 0$
16.	Sprawdzenie, czy $b = 0$	$b = 0$ TAK
17.	Komunikat końcowy	$a = 6$ jest NWD

Porównując tabele dla obu wersji algorytmu Euklidesa, szybko dojdiesz do wniosku, dlaczego pierwszy nazwano nieoptymalnym. Dla tych samych liczb w wersji nieoptymalnej warunki są badane 9 razy. W drugim przypadku – 4. Algorytm zoptymalizowanej metody zdecydowanie zmniejsza liczbę wykonywania pętli w wypadku, gdy mamy do czynienia z parą liczb pierwszych lub kiedy ich NWD jest mały. Czas na realizację programową w języku C++.

pętla while

W pętli głównej użyjemy instrukcji `while`.

```

1 #include <iostream>
2 using namespace std;
3 int a,b,c,d,reszta;
4 int main()
5 {
6     cout <<"wprowadź pierwszą liczbę ";
7     cin >> a;
8     c=a;
9     cout <<"wprowadź drugą liczbę ";
10    cin >> b;
11    d=b;
12    while (b!=0)
13    {
14        reszta=a % b;
15        a=b;
16        b=reszta;
17    }
18    cout << a <<" jest Największym Wspólnym Dzielnikiem liczb "<<<<" oraz "<<d <<endl;
19    return 0;
20 }

```

Rys. 11.5. Przykładowa realizacja algorytmu zoptymalizowanej metody Euklidesa

Jeśli przeanalizowałeś/przeanalizowałaś algorytm (rys. 11.4.), program przedstawiony na rysunku 11.5. nie wymaga omówienia. Podobnie jak w poprzednim i tu użyto dodatkowych zmiennych przechowujących dane wejściowe, by było możliwe wyświetlenie pełnego komunikatu (rys. 11.6.).



Rys. 11.6. Komunikaty wyświetlane przez przykładowy program obliczający NWD metodą zoptymalizowaną Euklidesa

Pozostaje nam już tylko ułożenie algorytmu obliczającego najmniejszą wspólną wielokrotność NWW dwóch liczb. Potrzebny będzie tylko jeden wzór:

$$\text{NWW}(a, b) = (a * b) \text{ div NWD}(a, b).$$

czyli podzielenie iloczynu obu liczb przez ich NWD. Spotkałeś/spotkałaś się już z tą metodą przy skracaniu ułamków. Zmodyfikujemy więc algorytm NWD w taki sposób, by w wyniku podawał NWW (rys. 11.7.).

Program po modyfikacji może obliczać zarówno NWD, jak i NWW (rys. 11.8.)

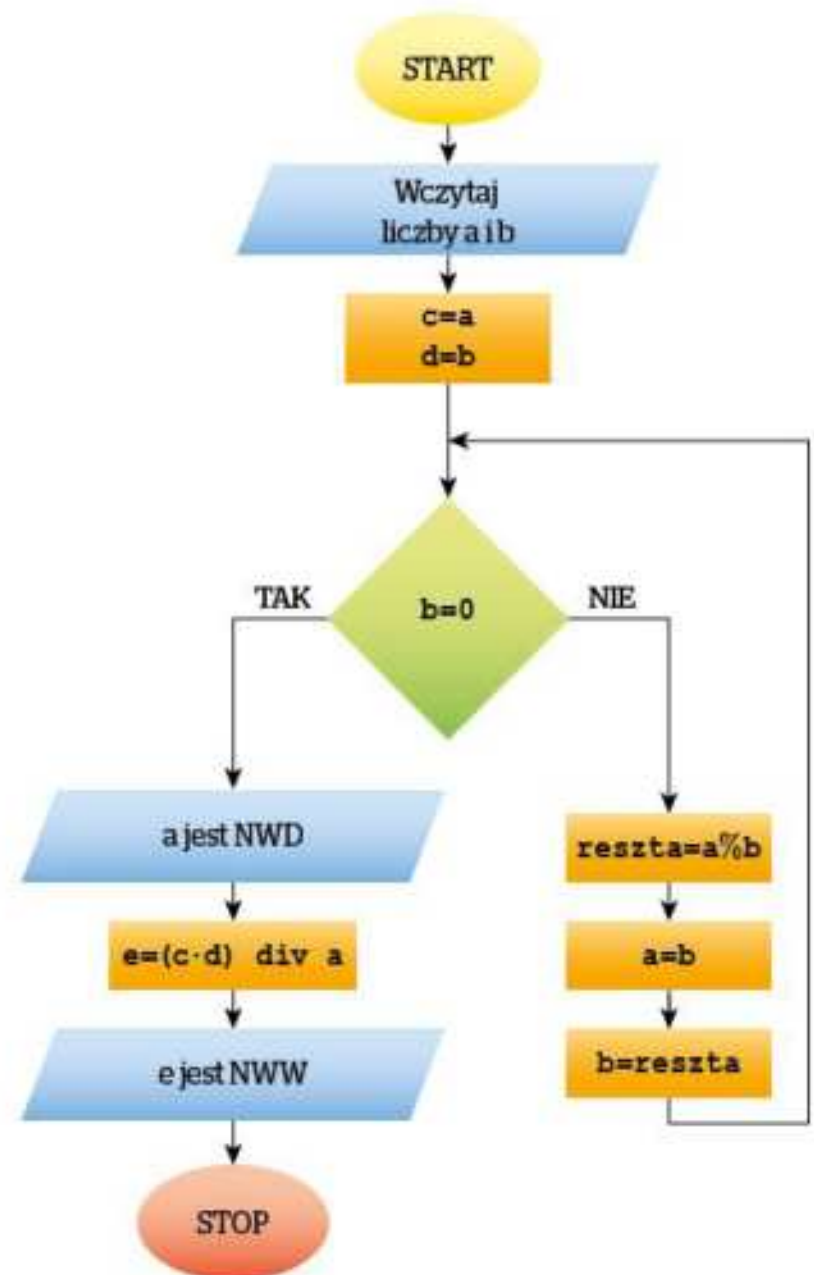
Zauważ, że NWW nie jest obliczane w osobnej linii programu, lecz bezpośrednio w poleceniu strumieniowania do konsoli `cout`. Dzięki temu nie ma potrzeby używania dodatkowej zmiennej. Byłaby potrzebna, gdyby NWW była używana w dalszej części programu.

```

1 #include <iostream>
2 using namespace std;
3 int a,b,c,d,reszta;
4
5 int main()
6 {
7     cout <<"wprowadź pierwszą liczbę ";
8     cin >> a;
9     c=a;
10    cout <<"wprowadź drugą liczbę ";
11    cin >> b;
12    d=b;
13    while (b!=0)
14    {
15        reszta=a % b;
16        a=b;
17        b=reszta;
18    }
19    cout << a <<" jest Największym Wspólnym Dzielnikiem liczb " << c <<" oraz " << d << endl;
20    cout << c*d/a <<" jest Najmniejszą Wspólną Wielokrotnością liczb " << c <<" oraz " << d << endl;
21    return 0;
22 }

```

Rys. 11.8. Program obliczający NWW



Rys. 11.7. Algorytm obliczania NWW

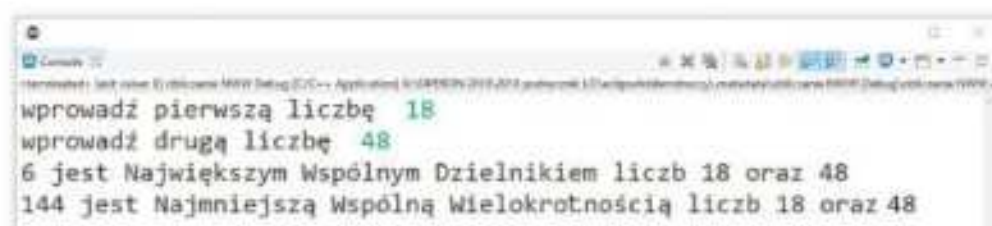
```

a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

Efekt działania tego programu jest wyświetlenie zarówno NWD, jak i NWW dla tej samej pary liczb.



```

wprowadź pierwszą liczbę 18
wprowadź drugą liczbę 48
6 jest Największym Wspólnym Dzielnikiem liczb 18 oraz 48
144 jest Najmniejszą Wspólną Wielokrotnością liczb 18 oraz 48

```

Rys. 11.9. Efekt działania programu obliczającego NWD i NWW dla tych samych liczb

ZADANIA DO ROZWIĄZANIA

1. Wyszukaj w internecie lub literaturze artykuły na temat metody Euklidesa. Porównaj proponowane rozwiązania z podanymi w rozdziale. Wskaż, na czym polegają różnice.
2. Zmodyfikuj program NWW tak, by umożliwiał obliczanie NWD dla dowolnej ilości par liczb. Inspiracji szukaj w poprzednim rozdziale. Dobierz odpowiednie komunikaty dla użytkownika.
3. Ułóż program obliczający NWW z zastosowaniem algorytmu nieoptymalnego Euklidesa znajdującego NWD.
- 4*. Ułóż program wskazujący taką parę liczb spośród 5 wprowadzonych, która będzie miała największy NWD. Opracuj odpowiednie komunikaty i zadбай o wygodne wprowadzanie danych do programu. Skorzystaj z doświadczenia tworzenia programu według optymalnej metody Euklidesa.

PODSUMOWANIE LEKCJI

s. 81

- Autorstwo metod obliczania NWD jest przypisywane Euklidesowi, a algorytmy zostały nazwane jego imieniem.

s. 81

s. 83

- Istnieje **algorytm nieoptymalny** i **optymalny**. Różnica w ilości powtórzeń pętli jest tym większa, im mniejszy względem najmniejszej z badanych liczb jest NWD.

s. 84

- Do realizacji pętli algorytmów NWD i NWW najwygodniej jest użyć **pętli while**.

12. Dodawanie ułamków, czyli jak wykorzystać NWW i NWD w programie komputerowym

NA TEJ LEKCJI:

- dowiesz się, jak ułożyć algorytm dodawania ułamków;
- poznasz funkcje.

NWW i NWD stosujesz od pierwszych lat nauki matematyki. Bez nich niemożliwe byłoby na przykład dodawanie ułamków. Umiesz to robić, ale czy wiesz, jak ułożyć algorytm i program wykonujący to działanie? Czy wiesz, jak wykorzystać w nich ułożone wcześniej programy obliczające NWW? Zmierz się z tym zadaniem.

12.1. Łączymy algorytmy, czyli nie wyważamy otwartych drzwi

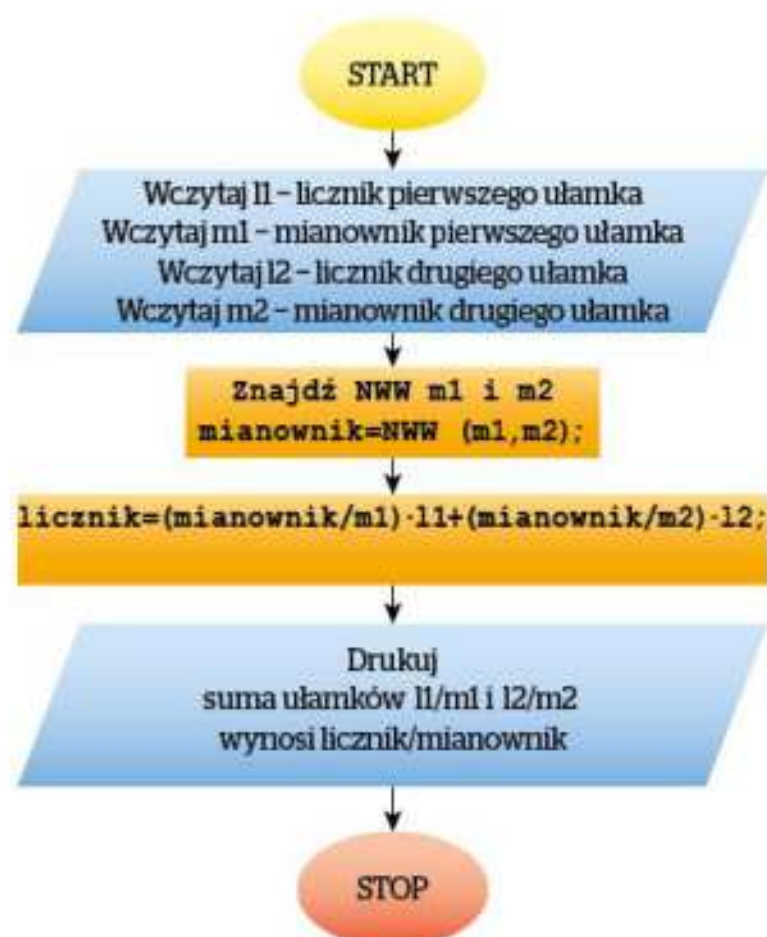
Budowanie algorytmu zaczniemy od opisu ogólnie znanej metody dodawania ułamków na prostym przykładzie.

$$\frac{3}{16} + \frac{5}{12} = \frac{3 \cdot 3}{48} + \frac{4 \cdot 5}{48} = \frac{9}{48} + \frac{20}{48} = \frac{29}{48}$$

Jak widać pierwszym krokiem było znalezienie NWW mianowników ułamków. Dla 16 i 12 to 48. Następnie należy podzielić NWW przez mianownik danego ułamka i wymnożyć wynik przez jego licznik. Mając dwa ułamki o tych samych mianownikach, wystarczy już tylko dodać do siebie liczniki i sprawdzić, czy wynik da się uprościć. Na tym właściwie można skończyć układanie algorytmu. Taka jego postać (lista kroków) jest przecież dopuszczalna. Spróbuj jednak utworzyć algorytm w postaci schematu blokowego (rys. 12.1.).

Jeśli przeanalizujesz algorytm z rysunku 12.1., dojdiesz do wniosku, że otrzymasz wynik w postaci nieuproszczonego ułamka. Rozwiązaniem tego problemu zajmiesz się w następnym rozdziale. Teraz skup się na ułożeniu programu w języku C++. Z pozoru nie powinno to sprawić ci trudności, ponieważ w algorytmie nie występują bloki warunkowe. Czy na pewno? Obliczenie NWW nie obędzie się bez warunków, ale ten problem rozwiązaliśmy wspólnie w poprzednim rozdziale.

algorytm
dodawania
ułamków



Rys. 12.1. Przykładowy schemat blokowy algorytmu dodawania ułamków

12.2. Kodujemy dodawanie, czyli układamy program dodający ułamki

Do ułożenia programu potrzeba kodu programu znajdującego NWW dwóch liczb. Z algorytmu wynika, że w programie wystąpią cztery zmienne wejściowe – licznik i mianownik każdej z liczb oraz dwie zmienne – licznik i mianownik wyniku.

```

1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int nww, a, b, i, l1, l2, m1, m2, licznik, mianownik;
7
8     cout<<"Podaj licznik pierwszego ułamka ";
9     cin >> l1;
10    cout << "Podaj mianownik pierwszego ułamka ";
11    cin >> m1;
12
13    cout<<"Podaj licznik drugiego ułamka ";
14    cin >> l2;
15    cout << "Podaj mianownik drugiego ułamka ";
16    cin >> m2;
17
18    a=m2;
19    b=m1;
20
21    while(b!=0)           //obliczanie NWD
22    {
23        i = b;
24        b = a%b;
25        a = i;
26    }                     // a = NWD
27
28    nww=m1*m2/a;
29
30    mianownik = nww;
31    licznik = mianownik/m1*l1 + mianownik/m2*l2;
32
33    cout<<l1<<"/"<<m1<<" + "<<l2<<"/"<<m2<<" = "<<licznik<<"/"<<mianownik;
34
35    return 0;
36 }

```

Rys. 12.2. Przykładowy program w języku C++ obliczający sumę dwóch ułamków

Główne obliczenia są wykonane w liniach 30 i 31. Korzystają one jednak z wcześniej obliczonego NWD. Największa wspólna wielokrotność zostaje z kolei obliczona na podstawie NWD. Tak więc te fragmenty kodu można było skopiować z programów z poprzedniego rozdziału. W programie z rysunku 12.2. nie jest konieczne używanie zmiennej `nww`. Użyto jej dla ułatwienia analizy programu. Zamiast linii 28 i 30 można by napisać: `mianownik = m1*m2/a;`.

Zwróć uwagę, jak program organizuje wprowadzenie danych. Dzięki odpowiednim komunikatom użytkownik programu wie, które wartości ma wprowadzić. Linia 33 to wydruk wyniku. Dzięki temu, że zmienne `l1`, `l2`, `m1` oraz `m2` nie zmieniają swojej wartości od momentu ich wpisania, można było podać wynik wraz z danymi wejściowymi (rys. 12.3.).

```

Podaj licznik pierwszego ułamka 3
Podaj mianownik pierwszego ułamka 16
Podaj licznik drugiego ułamka 5
Podaj mianownik drugiego ułamka 12
3/16 + 5/12 = 29/48

```

Rys. 12.3. Wynik działania przykładowego algorytmu dodawania ułamków

12.3. NWD i NWW jako funkcje, czyli modyfikujemy program

Ułożony program działa prawidłowo. Jednak gdyby nie komentarze, trudno byłoby niedoświadczonemu programiście domyślić się, jakie zadanie realizuje. A gdyby tak utworzyć oddzielne funkcje obliczania NWW i NWD? Wtedy, gdy w funkcji głównej należałoby użyć tych wartości, nie trzeba by ich obliczać. Obliczenia wykonałyby funkcje. W mało skomplikowanym programie zysk nie jest tak widoczny. Jeśli układa się program, w którym obliczanie pewnej wartości dla różnych danych występuje wielokrotnie, takie rozwiązanie jest wskazane. Kod staje się bardziej zrozumiały i uporządkowany, a także nie powtarzają się w nim w wielu miejscach te same fragmenty.

Przypomnij sobie, czym jest funkcja. To fragment programu, któremu nadano nazwę. Dzięki temu może być wykonywany zawsze, gdy zostanie ona użyta w kodzie. Funkcja może przetwarzać argumenty, które są zmiennymi wejściowymi. Istnieją funkcje bez argumentów.

funkcja

Zapiszmy funkcję obliczającą NWD (rys. 12.4.).

```

int NWD(int a, int b)    //Funkcja obliczająca NWD
{
    int i;
    while (b!=0)
    {
        i = b;
        b = a%b;
        a = i;
    }
    return a;
}

```

Rys. 12.4. Przykładowa funkcja obliczająca NWD dwóch liczb

Łatwo zauważyć, że kod funkcji zawiera te same instrukcje, co fragment programu z rys. 12.2. To oczywiste, gdyż sposób obliczania NWD się nie zmienił. Pierwsza linia `int NWD(int a, int b)` nadaje funkcji, której kod znajduje się poniżej, nazwę NWD oraz deklaruje jej argumenty. W tym przypadku to dwie zmienne o typie danych `int`, czyli liczb całkowitych.

Jeśli w pewnym miejscu kodu głównego programu (w funkcji `main()`) należy obliczyć NWW, a do tego potrzebne jest NWD dwóch liczb, można użyć funkcji (rys. 12.4.), podając w nawiasie obie liczby oddzielone przecinkiem. Będą one argumentami funkcji.

Przykładowe użycie funkcji NWD w celu obliczenia NWW dla argumentów `a` i `b`:

```
NWW=a/NWD(a,b)*b;
```

```

a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

Czas na użycie funkcji w programie dodającym dwa ułamki (rys. 12.5.).

```

1 #include<iostream>
2 using namespace std;
3
4 int NWD(int a, int b)          //Funkcja obliczająca NWD liczb całkowitych a i b
5 {
6     int i;
7     while(b!=0)
8     {
9         i = b;
10        b = a%b;
11        a = i;
12    }
13    return a;
14 }
15
16 int NWW(int a, int b)          //Funkcja obliczająca NWW liczb a i b
17 {                               //wykorzystująca funkcję NWD
18     return a/NWD(a,b)*b;
19 }
20
21 int main()                     //Główna funkcja programu
22 {
23     int l1, l2, m1, m2, licznik, mianownik;
24
25     cout<<"Podaj licznik pierwszego ułamka ";
26     cin >> l1;
27     cout << "Podaj mianownik pierwszego ułamka ";
28     cin >> m1;
29
30     cout<<"Podaj licznik drugiego ułamka ";
31     cin >> l2;
32     cout << "Podaj mianownik drugiego ułamka ";
33     cin >> m2;
34
35     mianownik = NWW(m1, m2);    //mianownik wyniku jest równy NWW mianowników dodawanych ułamków
36     licznik = (mianownik/m1)*l1 + (mianownik/m2)*l2; //Obliczanie sumy rozszerzeń liczników
37
38     cout<<l1<<"/"<<m1<< " + "<<l2<<"/"<<m2<< " = "<<licznik<<"/"<<mianownik;
39
40     return 0;
41 }

```

Rys. 12.5. Przykładowy program z zastosowaniem funkcji

Po dołączeniu standardowej biblioteki (linia 1) i określeniu przestrzeni zmiennych (linia 2) następuje zdefiniowanie funkcji obliczającej NWD, a następnie funkcji obliczającej NWW. Jak widać w linii 19, wewnątrz funkcji została użyta funkcja NWD zdefiniowana wcześniej.

W linii 37 i 38 następuje obliczenie sumy ułamków. Do otrzymania wyniku jest używana funkcja NWW.

Porównaj kody obu programów. W drugim został wprowadzony ład. Nie ma już problemu z analizą programu. Przykładowo w pierwszym kodzie to zmienna `a` przyjmuje wartość NWD i dalej jest pod tą nazwą używana. Mało doświadczonemu programiście może być trudno odgadnąć jej znaczenie poza pętlą obliczania NWD. W drugim NWD jest obliczane przez funkcję wywoływaną w tym samym miejscu, w którym stało `a`. Ponieważ jej przeznaczenie jest znane, analiza programu staje się łatwiejsza.

Zastosowanie funkcji ułatwia modyfikację lub poprawianie kodu. Dzięki użyciu funkcji ewentualne zmiany w jej kodzie nie pociągają za sobą konieczności zmian w części kodu, w którym ma ona być wykonana.

ZADANIA DO ROZWIĄZANIA

1. Zmodyfikuj algorytm dodawania ułamków w taki sposób, by obliczał sumę trzech ułamków. Ułóż go tak, by ułatwić programiście użycie funkcji obliczających NWW i NWD.
2. Zaproponuj inny sposób na wprowadzenie zmiennych **m1**, **m2**, **l1** i **l2** do programu dodającego ułamki. Może ona polegać na użyciu innych komunikatów, zmodyfikowanym wyświetlaniu wprowadzonych danych itp.
3. Sprawdź działanie programu dodającego ułamki dla liczb ujemnych. Dokonaj obliczeń samodzielnie, a następnie porównaj je z wynikami działania programu. Wyjaśnij, dlaczego program działa także dla liczb ujemnych. Czy można powiedzieć, że ten program odejmuje ułamki?
- 4*. Ułóż algorytm dodający dowolną, ustaloną liczbę ułamków.
- 5*. Ułóż program dodający dowolną liczbę ułamków. Zdefiniuj w tym celu odpowiednią funkcję.

PODSUMOWANIE LEKCJI

- **Algorytm dodawania ułamków** wykorzystuje algorytm znajdowania NWW i NWD. Jeśli znana jest NWW, dodawanie ułamków nieuwzględniające wyłączenia części całkowitej nie wymaga badania warunków. s. 87
- **Funkcja** to fragment programu, któremu nadano nazwę. Dzięki temu można ją wielokrotnie wykorzystać w kodzie programu, bez konieczności każdorazowego wpisywania odpowiadającego jej kodu. s. 89
Użycie funkcji w programie ułatwia jego analizę, późniejszą modyfikację i ogranicza wielkość kodu.

```
a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b
```

13. Skracamy i wyłączamy, czyli dodawania ułamków ciąg dalszy

NA TEJ LEKCJI:

- ułożysz program upraszczający ułamki;
- ułożysz program wyłączający z ułamka całości.

Twój program dodający ułamki już działa. Niestety wyniki podaje w postaci nieuproszczonej. Niektóre przekraczają także 1, czyli da się z nich wyłączyć całości. Nie zawsze jest to wskazane w dalszych obliczeniach, jednak wynik wyświetlany z wyłączeniem całości i w postaci uproszczonego ułamka będzie dla użytkownika programu atrakcyjniejszy. Zmodyfikujmy więc program.

13.1. Skracanie ułamków, czyli zmiana postaci wyniku działania programu dodawania

Program dodający dwa ułamki oblicza licznik i mianownik sumy. Przy czym mianownik jest najmniejszą wspólną wielokrotnością mianowników liczb dodawanych. Po wykonaniu obliczeń może się okazać, że wynik (ułamek) można uprościć. Umiesz już to robić. Jak zmodyfikować program, by dawał wynik w postaci uproszczonego ułamka?

Aby skrócić postać ułamka, należy znaleźć największy wspólny dzielnik licznika i mianownika. Kolejnym ruchem jest wykonanie dzielenia licznika i mianownika przez ich NWD.

W drugiej wersji programu dodającego ułamki pojawiły się funkcje. Tym razem funkcja obliczająca NWD będzie wykorzystana wielokrotnie – do wykonania podstawowego zadania, czyli dodania ułamków, oraz uproszczenia postaci wyniku.

modyfikacja
programu a funkcje

```

1 #include<iostream>
2 using namespace std;
3
4 int NWD(int a, int b)      //Funkcja obliczająca NWD
5 {
6     int i;
7     while(b!=0)
8     {
9         i = b;
10        b = a%b;
11        a = i;
12    }
13    return (a);
14 }
15
16
17 int NWW(int a, int b)      //Funkcja obliczająca NWW
18 {
19     return a/NWD(a,b)*b;
20 }
21
22 int main()                //Główna funkcja programu
23 {
24     int i, l1, l2, m1, m2, licznik, mianownik;
25
26
27     cout<<"Podaj licznik pierwszego ułamka ";
28     cin >> l1;
29     cout << "Podaj mianownik pierwszego ułamka ";
30     cin >> m1;
31
32     cout<<"Podaj licznik drugiego ułamka ";
33     cin >> l2;
34     cout << "Podaj mianownik drugiego ułamka ";
35     cin >> m2;
36
37     mianownik = NWW(m1, m2);
38     i = licznik = mianownik/m1*l1 + mianownik/m2*l2;
39     licznik = licznik / NWD(i,mianownik);
40     mianownik = mianownik / NWD(i,mianownik);
41
42     cout<<l1<<"/"<<m1<< " + "<<l2<<"/"<<m2<< " = "<<licznik<<"/"<<mianownik;
43
44     return 0;
45 }

```

Rys. 13.1. Przykładowy program dodający ułamki i podający uproszczony wynik

Do testu poprawności działania użyjemy tych samych liczb.

```

Podaj licznik pierwszego ułamka 12
Podaj mianownik pierwszego ułamka 18
Podaj licznik drugiego ułamka 7
Podaj mianownik drugiego ułamka 12
12/18 + 7/12 = 45/36

```

```

Podaj licznik pierwszego ułamka 12
Podaj mianownik pierwszego ułamka 18
Podaj licznik drugiego ułamka 7
Podaj mianownik drugiego ułamka 12
12/18 + 7/12 = 5/4

```

Rys. 13.2. Porównanie wyników działania programów dodających ułamki

Do programu dodającego ułamki dodano dwie linie (39 i 40). Są one zgodne z założeniami poczynionymi powyżej, jednak nie dokładnie. Dlaczego?

```

length; c.unique = b.length;
c)); } return b.length;
, b), b = b.replace(
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

zmienna
pomocnicza

W działaniach użyto zmiennej `i`. Przeanalizuj jej rolę w programie. Zauważ, że przechowuje ona licznik pierwotnego wyniku. Co by się stało, gdyby nie została zastosowana (rys. 13.2.)?

```

37 mianownik = NWW(m1, m2);
38 licznik = mianownik/m1*l1 + mianownik/m2*l2;
39 licznik = licznik / NWD(licznik,mianownik);
40 mianownik = mianownik / NWD(licznik,mianownik);
41
42 cout<<l1<<"/"<<m1<< + "<<l2<<"/"<<m2<< = "<<licznik<<"/"<<mianownik;

```



Podaj licznik pierwszego ułamka 12
 Podaj mianownik pierwszego ułamka 18
 Podaj licznik drugiego ułamka 7
 Podaj mianownik drugiego ułamka 12
 12/18 + 7/12 = 5/36

Rys. 13.3. Nieprawidłowy wynik działania programu bez zmiennej pomocniczej `i`

Analiza

W liniach 37 i 38 (rys. 13.3.) następuje dodanie danych ułamków. Linia 39 daje uproszczenie licznika. Od tej chwili zmienna `licznik` ma nową wartość. Obliczanie na jej podstawie NWD dla uproszczenia mianownika jest więc błędem.

Czy można inaczej ułożyć fragment programu odpowiedzialny za uproszczenie postaci wyniku? W programie z rysunku 13.1. funkcja NWD we fragmencie upraszczającym wynik jest wywoływana dwukrotnie. Można użyć jej tylko jeden raz, podstawiając wynik jej działania pod zmienną `i` (rys. 13.4.).

```

37 mianownik = NWW(m1, m2);
38 licznik = mianownik/m1*l1 + mianownik/m2*l2;
39 i=NWD(licznik, mianownik);
40 licznik = licznik / i;
41 mianownik = mianownik / i;

```

Rys. 13.4. Fragment programu z jednokrotnym wywołaniem funkcji NWD

Po takim działaniu linie 40 i 41 nie zawierają już wywołania funkcji, a jedynie używają zmiennej `i`. Z punktu widzenia czasu wykonywania programu jest to rozwiązanie korzystniejsze. Tracimy jednak na czytelności kodu.

13.2. Wyłączamy całości, czyli wynik w ostatecznej postaci

Wynik w postaci ułamka może być przydatny, gdy używamy go do dalszych obliczeń. Jeśli jest on ostatecznym wynikiem wyświetlanym użytkownikowi, ciekawiej byłoby wyłączyć z niego całości. Jak to zrobić bez użycia komputera?

1. Podziel licznik przez mianownik.
2. Zbadaj, czy część całkowita wyniku jest różna od 0. Jeśli tak, zapisz ją jako całkowitą część badanego ułamka.
3. Licznik ułamka zmniejsz o iloczyn części całkowitej i mianownika.

Zrealizujemy ten algorytm, dołączając go do kodu programu dodającego ułamki (rys. 13.5.).

```

21 int main() //Główna funkcja programu
22 {
23     int c, i, l1, l2, m1, m2, licznik, mianownik;
24
25     cout<<"Podaj licznik pierwszego ułamka ";
26     cin >> l1;
27     cout << "Podaj mianownik pierwszego ułamka ";
28     cin >> m1;
29
30     cout<<"Podaj licznik drugiego ułamka ";
31     cin >> l2;
32     cout << "Podaj mianownik drugiego ułamka ";
33     cin >> m2;
34
35     mianownik = NWW(m1, m2);
36     licznik = mianownik/m1*l1 + mianownik/m2*l2;
37     i=NWD(licznik, mianownik);
38     licznik = licznik / i;
39     mianownik = mianownik / i;
40
41     if (licznik/mianownik != 0)
42     {
43         c=licznik/mianownik;
44         licznik=licznik-c*mianownik;
45         cout<<l1<<"/"<<m1<< " + "<<l2<<"/"<<m2<< " = " <<c << " i "<<licznik<<"/"<<mianownik;
46     }
47     else
48         cout<<l1<<"/"<<m1<< " + "<<l2<<"/"<<m2<< " = "<<licznik<<"/"<<mianownik;
49
50     return 0;
51 }

```

Rys. 13.5. Funkcja główna programu zawierającego wyłączenie z wyniku dodawania ułamków części całkowitej

Instrukcja `if` została użyta jedynie w celu zapewnienia właściwego wydruku wyniku. Bez badania tego warunku program w miejscu części całkowitej mógłby wydrukować 0 dla ułamków mniejszych od 1. Nie ma potrzeby szukania części całkowitej z dzielenia licznika przez mianownik, ponieważ zmienna `c` jest zadeklarowana jako całkowita, więc wynik dzielenia nie będzie zawierał części ułamkowej. Dla wyników mniejszych od 1 `c=0`. Sprawdźmy poprawność działania dla naszych danych testowych (rys. 13.6.).

**zmienna
zadeklarowana
jako całkowita**



Rys. 13.6. Wynik działania programu z wyłączeniem części całkowitej z wyniku dodawania ułamków

```
a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b
```

ZADANIA DO ROZWIĄZANIA

1. Na podstawie analizy programu z podręcznika wyjaśnij znaczenie i celowość tworzenia funkcji.
2. Zmodyfikuj program dodawania ułamków w taki sposób, by podawał wynik w postaci równości, w której po lewej stronie zostanie wyświetlony ułamek przed uproszczeniem, a po prawej – skrócony.
3. Po wykonaniu zadania 2. dodaj do wydruku wynik z wyłączoną częścią całkowitą.
4. Zbadaj działanie programu z rozdziału w wersji ostatecznej dla różnych ułamków. Sprawdź, jak jest wyświetlany ujemny wynik. Zmodyfikuj wyświetlanie w taki sposób, by znak „-” został wyświetlony przed ostatecznym wynikiem, np. -1 i $3/4$.
- 5*. Zmodyfikuj program z rysunku 13.5. w taki sposób, by można było dodawać trzy ułamki. Przedstaw sposób takiej modyfikacji dla dowolnej liczby ułamków.

PODSUMOWANIE LEKCJI

s. 92

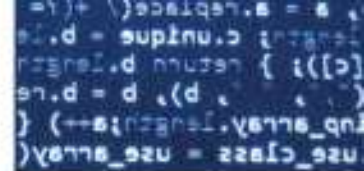
- Łatwa **modyfikacja programu** jest możliwa dzięki użyciu funkcji.

s. 94

- **Zmienna pomocnicza** pozwoliła uniknąć błędnego obliczenia wyniku.

s. 95

- Wynik dzielenia zapisany pod **zmienną** zadeklarowaną jako **całkowita** zawiera jedynie część całkowitą tego działania.



14. Szyfrowanie, czyli poznajemy szyfr Cezara

NA TEJ LEKCJI:

- poznasz prosty szyfr przesuwający, czyli szyfr Cezara;
- poznasz szyfr wieloalfabetowy;
- dowiesz się, jak ważne jest szyfrowanie informacji.

Nie wiemy, kto pierwszy zaszyfrował informację. Wiemy za to, w jakim celu to zrobił – nie chciał, by została ona odczytana przez osoby do tego nieuprawnione. Z różnymi sposobami szyfrowania można się spotkać w powieściach lub filmach. Każdy może wymyślić własny szyfr. Może on być bardzo skomplikowany, gdy użyjesz nietypowych, choć prostych rozwiązań. Ale zanim zaczniesz szyfrować własnym szyfrem, poznaj szyfr, z którego według niektórych źródeł jako pierwszy skorzystał Juliusz Cezar. Stąd nazwa – szyfr Cezara.

14.1. Prosty szyfr przesuwający, czyli Cezar szyfruje informacje

Wyobraź sobie, że przechwytujesz tajną informację, na przykładą taką:

`vcbiufhcdud`

Zdajesz sobie sprawę, że do jej rozszyfrowania potrzebujesz klucza. Klucz może być wskazówką, wzorem lub opisem metody rozszyfrowania tego tekstu. Komunikat składa się z liter alfabetu łacińskiego, ale jego znaczenie jest zagadką. A może rozwiązanie tego problemu wcale nie jest tak trudne, na jakie wygląda? Wcielmy się w rolę kryptologa, przed którym stanęło zadanie rozszyfrowania informacji. Zaczniemy od wypisania do tabeli pierwszych kilku liter szyfru (tab. 14.1.). W sąsiednich kolumnach wpiszmy litery występujące w alfabecie w sąsiedztwie znaków szyfru. Dla ułatwienia wypisz obok wszystkie litery alfabetu łacińskiego:

`a b c d e f g h i j k l m n o p q r s t u v w x y z`

Co zrobić, gdy w jednym lub drugim kierunku zostanie litera? Zapętl alfabet, zakładając, że po literze `z` wystąpi `a` oraz przed `a` wystąpi `z`.

szyfr Cezara

```

length; c.unique = b.length;
c)); } return b.length;
, b), b = b.replace(
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

Tab. 14.1. Tabela pomocna w odszyfrowaniu szyfru przesuwającego

3 litery w lewo	2 litery w lewo	1 litera w lewo	Zaszyfrowana informacja	1 litera w prawo	2 litery w prawo	3 litery w prawo
s	t	u	v	w	x	y
z	a	b	c	d	e	f
y	z	a	b	c	d	e
f	g	h	i	j	k	l
r	s	t	u	v	w	x
c	d	e	f	g	h	i
e	f	g	h	i	j	k
z	a	b	c	d	e	f
a	b	c	d	e	f	g
r	s	t	u	v	w	x
a	b	c	d	e	f	g

Teraz pozostaje analiza otrzymanych wyników. Szybko okaże się, że sensowne wyrazy otrzymasz już w pierwszej kolumnie, w której każdą z liter kodu zastąpiono literą poprzedzającą ją o **trzy pozycje** w alfabecie. Tak właśnie postąpiono w czasach Juliusza Cezara.

Jak opiszesz klucz potrzebny do **zaszyfrowania informacji**? Pomoże w tym diagram (rys. 14.1.).



Każdy znak przesuwamy o 3 znaki w prawo:

A → D
 B → E
 C → F
 ...
 Z → C

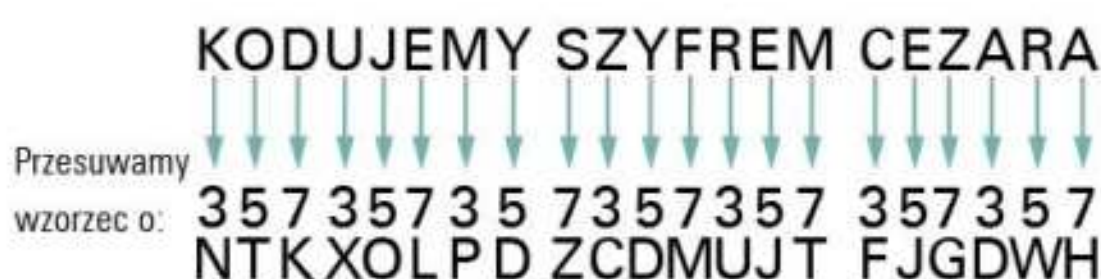
Rys. 14.1. Diagram ilustrujący metodę szyfrowania szyfrem Cezara

Wniosek: każdą z liter oryginalnej wiadomości należy zastąpić trzecią literą następującą po niej w alfabecie. Proces deszyfracji przebiega odwrotnie. Jeśli każdą literę zastąpimy literą, której pozycja w alfabecie jest oddalona o pewną stałą liczbę (np. o 3), to tę liczbę można nazwać kluczem szyfrowania.

Czy istnieją jeszcze inne szyfry oparte na metodzie przesuwania znaków? Oczywiście, że tak. Ty także możesz taką opracować, zmieniając liczbę lub kierunek podstawień. Ponieważ ta metoda jest dość znana, mogłaby być nieskuteczna. Dlatego można ją udoskonalić, tworząc **szyfr przesuwający wieloalfabetowy**. W jaki sposób? Czy liczba podstawień musi zawsze być taka sama? Oczywiście, że nie. Mogą to być na przykład liczby pierwsze 3, 5 i 7. W tym przypadku odnalezienie 1 litery odkodowanej wiadomości przebiegałoby jak w szyfrze Cezara. Następną znaleźć można nie na trzeciej, a na piątej pozycji, kolejną na siódmej, a czwartą – ponownie na trzeciej (rys. 14.2.).

szyfr
wieloalfabetowy

ABCDEFGHIJKLMNOPQRSTUVWXYZ



Zmiana szyfru następuje wraz ze zmianą pozycji znaku:

- pierwszy znak szyfrujemy, przesuwając wzorzec o **3 znaki** w prawo;
- drugi znak szyfrujemy, przesuwając wzorzec o **5 znaków** w prawo;
- trzeci znak szyfrujemy, przesuwając wzorzec o **7 znaków** w prawo;
- czwarty znak szyfrujemy, przesuwając wzorzec o **3 znaki** w prawo.

Rys. 14.2. Szyfrowanie przykładowym szyfrem wieloalfabetowym

W powyższych przykładach nie stosuje się przerw między wyrazami, ponieważ to ułatwiłoby deszyfrację. Nie korzysta się także z polskich znaków, a ponadto nie rozróżnia się dużych i małych liter.

14.2. Szyfr przesuwający na schemacie, czyli tworzymy algorytm

Algorytm szczegółowy tej metody ułożony z myślą o realizacji w języku programowania musiałby opisywać sposób zapisywania liter w zmiennej tablicowej o rozmiarze zgodnym z długością tekstu. Musisz nabyć więcej umiejętności, by sprostać temu zadaniu. Nic nie stoi na przeszkodzie byś poznał/poznała istotę szyfru Cezara (rys. 14.3.).

```

a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length;
, b), b = b.rep
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY



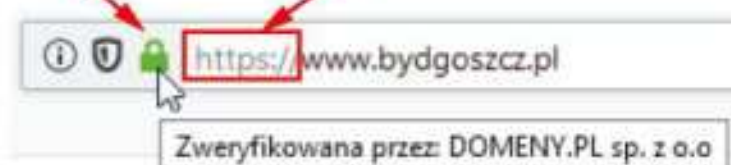
Rys. 14.3. Algorytm opisujący metodę szyfrowania szyfrem Cezara

szyfrowanie informacji

Szyfrowanie informacji ma wielkie znaczenie dla użytkowników sieci. Oczywiście nie korzysta się z tak prymitywnych metod jak szyfr Cezara. Szyfrowanie, choć ważne w zastosowaniach indywidualnych, ma ogromne znaczenie dla instytucji odpowiedzialnych za bezpieczeństwo danych. Szyfrowane są nie tylko informacje, lecz także transmisje sieciowe. Masz z tym do czynienia za każdym razem, gdy logujesz się do swojej poczty lub na konto dowolnej usługi. Skąd o tym wiadomo? Zwróć uwagę na symbole, które pokazują się obok adresu takiej usługi (rys. 14.4.).

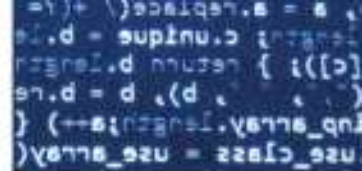
Kłódka oznacza, że strona została zweryfikowana pod kątem bezpieczeństwa.

HTTPS (Hypertext Transfer Protocol Secure) oznacza, że transmisja danych w protokole HTTP jest szyfrowana.



HTTPS

Rys. 14.4. Oznaczenie połączenia szyfrowanego dla strony internetowej



ZADANIA DO ROZWIĄZANIA

1. Opracuj własny szyfr podstawieniowy oparty na kluczu wybierającym kolejne litery. Zszyfruj nim tekst „Szyfrszefa”. Narysuj w zeszycie schemat blokowy opisujący twoją metodę szyfrowania.
2. Zastosuj dwukrotnie szyfr Cezara do zaszyfrowania tekstu „czytaniejestsuper”. Oznacza to, że będziesz musiał/musiała po jednokrotnym szyfrowaniu powtórzyć czynności dla już zakodowanej informacji. Sprawdź doświadczalnie, czy da to taki sam efekt, jak szyfrowanie ze skokiem 6 zamiast 3, jak w szyfrze Cezara. Wyciągnij wnioski i je uzasadnij. Postaraj się uogólnić opis tego przypadku na inne, podobne.
3. Czy algorytm z rysunku 14.3. może służyć do opisanie dowolnej metody opartej na stosowanej w szyfrze Cezara? Jeśli nie, to co należałoby zmienić?
4. W szyfrze Cezara zastosowano skok o 3 litery. Dla jakiej liczby skoków szyfrowanie nie miałoby sensu? Do jakiej liczby skoków szyfrowanie tą metodą ma sens? Jakie liczby skoków dałyby takie same efekty szyfrowania?
- 5**. Ułóż algorytm do napisania programu szyfrującego metodą szyfru Cezara. Dla ułatwienia zastosuj stałą długość kodowanego tekstu.

PODSUMOWANIE LEKCJI

- Jedną z metod podstawieniowych szyfrowania informacji jest **szyfr Cezara**, w którym każdą literę w informacji zastępuje się literą znajdującą się w alfabecie łacińskim o trzy pozycje dalej. Przykładowo: a → d. s. 97
- Istnieje nieskończona ilość tworzenia metod szyfrowania **metodą podstawieniową**, ponieważ można korzystać z różnych sposobów wybierania kolejnej litery, np. co czwartą lub w odległościach opisanych kluczem, np. 3, 5, 7. Mamy wtedy do czynienia z **szyframi wieloalfabetowymi**. s. 99
- **Szyfrowanie skomplikowanymi metodami** ma wielkie znaczenie dla bezpieczeństwa danych. s. 100
- **https://** przed adresem oznacza szyfrowanie transmisji dla tej strony, np. podczas logowania się do konta pocztowego, konta usługi. s. 100

15. Konspiracja, czyli tworzymy własne szyfry

NA TEJ LEKCJI:

- zrozumiesz, jaką rolę odgrywają klucze w szyfrowaniu informacji;
- dowiesz się, czym są kryptografia symetryczna i asymetryczna;
- ułożysz własny szyfr z kluczem generowanym losowo.

Kryptografia to jedna z ciekawszych dziedzin nauki. Zajmują się nią naukowcy na najśłynniejszych uniwersytetach świata. Skuteczne szyfrowanie informacji stało się jednym z najważniejszych zagadnień współczesnej informatyki. Domowy użytkownik sieci korzysta z szyfrowanych połączeń na przykład z bankiem. Rządziej szyfruje swoje listy. Jednak szyfrowanie korespondencji i przesyłania danych pomiędzy firmami, bankami, politykami itp. jest już normą – im skuteczniejsze sposoby, tym bezpieczniejsze dane. Czy zatem uczeń może stworzyć własny, niepowtarzalny kod ze zmiennym kluczem? Zmierzymy się z tym problemem.

15.1. Zamknij – otwórz, czyli jaką rolę odgrywają klucze w szyfrowaniu informacji

Do otwarcia każdego zamka potrzebny jest klucz. Nie ma znaczenia, czy to zamek mechaniczny, elektroniczny czy softwarowy. Klucz jest niezbędny. Dopóki znają go wyłącznie upoważnione osoby, informacja jest bezpieczna.

Klucz w szyfrowaniu stanowi informację, dzięki której można zaszyfrować lub odszyfrować informację lub dane.

Algorytmy symetryczne do szyfrowania i odszyfrowywania wykorzystują klucze znane nadawcy i odbiorcy. Mówimy wtedy, że mamy do czynienia z **kryptografią symetryczną** (rys. 15.1.). Zazwyczaj dla zwiększenia bezpieczeństwa dla każdego połączenia generuje się nowy klucz. Należy więc znaleźć sposób na przekazanie go odbiorcy.

Do tego można użyć metody **asymetrycznej**, która jest bezpieczniejsza, ale wymaga skomplikowanych obliczeń. Dlatego stosuje się ją najczęściej do szyfrowania krótkich informacji.

klucz
w szyfrowaniu

kryptografia
symetryczna

kryptografia
asymetryczna



Rys. 15.1. Schemat szyfrowania symetrycznego

W szyfrowaniu asymetrycznym używa się **kluczy publicznych i prywatnych**.

Obu rodzajów używa się w **szyfrowaniu asymetrycznym**. Klucz prywatny pełni funkcję swego rodzaju podpisu nadawcy i powinien być znany tylko jemu. Klucz prywatny i publiczny stanowią unikatową parę (rys. 15.2.).

szyfrowanie
asymetryczne



Rys. 15.2. Schemat szyfrowania asymetrycznego

Funkcje kluczy mogą być łączone, na przykład informacja najpierw jest podpisywana kluczem prywatnym nadawcy, a potem szyfrowana kluczem publicznym odbiorcy. Po jej odebraniu program deszyfrujący odwraca kolejność działań. Odszyfrowuje informację kluczem prywatnym odbiorcy, a następnie sprawdza zgodność sygnatury z kluczem publicznym autora. Klucze prywatne i publiczne są tworzone na różne sposoby – im bardziej wyrafinowany, tym klucz jest trudniejszy do odgadnięcia. Jeśli zainteresował cię temat kryptografii, możesz dowiedzieć się więcej z dostępnych publikacji.

Czy zatem każdy jest w stanie utworzyć własny sposób szyfrowania informacji? Zawsze można opracować sposób oparty na kryptografii symetrycznej z jednym kluczem do szyfrowania i deszyfrowania, lecz zmieniającym się na przykład losowo. Taki klucz będzie musiał znać także odbiorca. Do dzieła.

15.2. Klucz w tablicy, czyli szyfrujemy skuteczniej od Cezara

W przypadku szyfru Cezara kluczem była liczba wyznaczająca znak do podstawienia za oryginalną literę. Wykorzystajmy ten pomysł, jednak zamiast stałego skoku użyjemy zmiennego.

```
a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b
```

III. PROGRAMOWANIE I ALGORYTMY

tablica kodów ASCII

Ponadto zamiast liter z tekstu wykorzystamy wszystkie znaki z 256-znakowej tablicy kodów ASCII (informacje o kodach ASCII przyporządkowanych poszczególnym znakom i literom powinien(e)s/powinnaś znać ze szkoły podstawowej; jeśli nie pamiętasz, skorzystaj z wiadomości w internecie). Oznacza to, że litery w tablicy zawierającej tekst do szyfrowania będą traktowane jako odpowiadające im kody, czyli liczby. Do nich zostanie dodany klucz i z powrotem wyświetlony jako tekst (tab. 15.1.). Taka operacja prowadzi do zaszyfrowania tekstu. Deszyfracja polega na odejmowaniu wartości klucza (tab. 15.2.).

Zaszyfrujmy tym sposobem wyraz „Dom”.

Tab. 15.1. Przykład szyfrowania

Litera tekstu	ASCII	Klucz	ASCII po dodaniu klucza	Zaszyfrowany wyraz
D	68	12	80	P
o	111	13	124	l
m	109	5	114	r

Tab. 15.2. Przykład deszyfracji

Litera tekstu	ASCII	Klucz	ASCII po dodaniu klucza	Zaszyfrowany wyraz
P	80	12	68	D
l	124	13	111	o
r	114	5	109	m

Do zaszyfrowania i odszyfrowania danej litery użyjemy tego samego klucza. Dla każdego kolejnego znaku będzie on inny i zgodny z tablicą kluczy.

W pierwszym projekcie użyjemy stałego zestawu kluczy w postaci 20-elementowej tablicy z liczbami (rys. 15.3.).

```

1 #include <iostream>
2 #include <ctime>
3 using namespace std;
4
5 void szyfrowanie(char *tekst, int przesuniecie[])
6 {
7     int skok = 0;
8     for (int i=0; tekst[i] != '\0'; i++)
9     {
10         tekst[i] = ( tekst[i] + przesuniecie[skok] );
11         skok ++;
12         if(skok == 19) skok = 0;
13     }
14 }
15
16 void odszyfrowanie(char *tekst, int przesuniecie[])
17 {
18     int skok = 0;
19     for (int i=0; tekst[i] != '\0'; i++)
20     {
21         tekst[i] = ( tekst[i] - przesuniecie[skok] );
22         skok ++;
23         if(skok == 19) skok = 0;
24     }
25 }
26
27 int main()
28 {
29     int klucz[]={12,13,5,3,4,6,12,19,11,10,9,8,3,7,1,14,12,17,6,2};
30
31     cout<<endl;
32     char tekstdoszyfrowania[100];
33     cin.get ( tekstdoszyfrowania, 100);
34     cout << tekstdoszyfrowania <<endl;
35     szyfrowanie(tekstdoszyfrowania, klucz);
36     cout << tekstdoszyfrowania << endl;
37     odszyfrowanie(tekstdoszyfrowania, klucz);
38     cout << tekstdoszyfrowania;
39
40     return 0;
41 }

```

Funkcja szyfrująca

Funkcja deszyfrująca

Rys. 15.3. Przykładowy program szyfrujący i deszyfrujący ze stałym kluczem w postaci tablicy liczb

W przykładowym programie zbudowano dwie funkcje – jedna szyfruje, a druga odszyfrowuje informacje tekstowe. Ta konstrukcja pozwala na prześledzenie procesu kodowania i sprawdzenie poprawności działania. Oczywiście jeśli ta metoda miałaby być wykorzystana, musiałyby istnieć dwa oddzielne programy wykorzystujące ten sam klucz. Założono także, że ciąg znaków, jaki tworzy informację, nie będzie dłuższy od 100. Oczywiście łatwo to zmienić w **liniach 32 i 33**.

Główną funkcję programu zaczyna utworzenie 20-elementowej tablicy z wpisanym kluczem. Są to przypadkowo ułożone liczby z zakresu od 0 do 19. Zauważ, że niektóre się powtarzają. Nie ma to znaczenia dla skuteczności szyfrowania.

Linia 32 to deklaracja tablicy dla zmiennej wejściowej zawierającej tekst do zaszyfrowania. Kolejna linia to strumieniowanie znaków z klawiatury, czyli wprowadzanie tekstu, który ma być szyfrowany. Funkcja `cin.get` służy do wprowadzania znaków do zmiennej tablicowej. Wymaga podania nazwy zmiennej i maksymalnej liczby znaków. Użycie `cin` spowodowałoby wstawienie w miejscu spacji wartości 0, co zakończyłoby pętlę szyfrowania.

Linia 34 to kontrolne wyświetlanie wprowadzonego tekstu. Oczywiście nie ma wpływu na szyfrowanie, ale w naszym przykładzie posłuży do porównania z tekstem po odszyfrowaniu.

```

        b.replace(0, b.length(), c);
        return b;
    }
    use_class = use_array(b);
}

```

III. PROGRAMOWANIE I ALGORYTMY

Linia 35 to wywołanie funkcji szyfrującej z przekazaniem wartości zmiennych. Pierwszą jest tablica z tekstem do szyfrowania, drugą – klucz szyfrujący.

Linia 36 wysyła strumień z zakodowanym tekstem. W tym momencie zmienna tablicowa uprzednio zawierająca tekst do szyfrowania została zmieniona przez funkcję szyfrującą z zakodowaną informacją.

Linia 37 wywołuje funkcję deszyfrującą. Program przekazuje jej tablicę z zaszyfrowanym tekstem i tablicę z kluczem.

Linia 38 strumieniuje odszyfrowany tekst (zmienioną przez funkcję deszyfrującą zmienną tablicową) na ekran konsoli. Teraz można porównać tekst przed szyfrowaniem i po nim (rys. 15.4.).



Rys. 15.4. Wynik działania przykładowego programu szyfrującego i deszyfrującego ze stałym kluczem

zmienna tablicowa

A jak działa sam mechanizm szyfrowania? Wewnątrz funkcji zmienna `tekst[]` jest wypełniona wartościami kodów ASCII znaków z szyfrowanego tekstu, a tablica `przesuniecie[]` przyjmuje wartości zmiennej tablicowej `klucz[]`. Pętla `for` wykonuje się do momentu, gdy zostanie wykryty koniec tekstu:

```
for (int i=0; tekst[i] != '\0'; i++)
```

Kluczowe jest tu użycie warunku `tekst[i] != '\0'`, czyli badanie, czy kolejny element tablicy nie jest pusty. A co w pętli? Do kolejnych kodów liter tekstu szyfrowanego zostaną dodane kolejne liczby z tablicy kluczy. Użyty warunek pozwala na wielokrotne, cykliczne użycie tablicy z kluczami w przypadku kodowania tekstu dłuższego niż 20 znaków. Funkcja zmienia elementy tablicy, w której wcześniej był tekst do szyfrowania. Teraz jej zawartość stanowi zaszyfrowany tekst.

Funkcja deszyfrująca jest bardzo podobna do szyfrującej. Ma przywrócić pierwotne wartości tablicy z tekstem, dlatego musi od aktualnych kodów ASCII odjąć te same klucze (liczby), które służyły do ich zakodowania.

15.3. Generujemy losowo, czyli dla każdego szyfrowania inny klucz

Skomplikujmy nieco życie złodziejom informacji i zamiast stałej, zaszytej w kodzie programu tablicy kluczy użyjmy takiej, która przy każdorazowym uruchomieniu szyfrowania będzie zawierała inne klucze.

Czy to możliwe? Jeśli użyjemy generatora liczb pseudolosowych, nie będzie to trudne. Zmodyfikujmy więc nasz program, dodając kod wypełniający przypadkowymi liczbami tablicę kluczy (rys. 15.5.).

```

1 #include <iostream>
2 #include <ctime>
3 using namespace std;
4
5 void szyfrowanie(char *tekst, int przesuniecie[])
6 {
7     int skok = 0;
8     for (int i=0; tekst[i] != '\0'; i++)
9     {
10         tekst[i] = ( tekst[i] + przesuniecie[skok] );
11         skok ++;
12         if(skok == 19) skok = 0;
13     }
14 }
15 void odszyfrowanie(char *tekst, int przesuniecie[])
16 {
17     int skok = 0;
18     for (int i=0; tekst[i] != '\0'; i++)
19     {
20         tekst[i] = ( tekst[i] - przesuniecie[skok] );
21         skok ++;
22         if(skok == 19) skok = 0;
23     }
24 }
25
26 int main()
27 {
28     int klucz[20], i;
29     srand( time( NULL ) );
30     for (i=0; i<20; i++)
31     {
32         klucz[i]=((rand() % 19 ) + 0 );
33         cout << klucz[i] << " ";
34     }
35
36     cout<<endl;
37     char tekstdoszyfrowania[100];
38     cin.get ( tekstdoszyfrowania, 100);
39     cout << tekstdoszyfrowania <<endl;
40     szyfrowanie(tekstdoszyfrowania, klucz);
41     cout << tekstdoszyfrowania << endl;
42     odszyfrowanie(tekstdoszyfrowania, klucz);
43     cout << tekstdoszyfrowania;
44     return 0;
45 }

```

Rys. 15.5. Przykładowy program z losowo wypełnianą tablicą kluczy

Do generowania liczb losowych przydatnych w naszym zadaniu będzie potrzebna funkcja `time()`. Wymaga to dołączenia biblioteki `ctime`.

W miejscu deklaracji wypełnionej kluczami tablicy `klucz[]`, został dodany kod wypełniający tablicę o tej samej nazwie (linie od 28 do 35) losowymi liczbami. Dzięki zachowaniu nazw zmiennych nie trzeba modyfikować reszty kodu programu.

Jak działa wypełnianie tablicy losowo wybranymi liczbami?

```

int klucz[20], i;
srand( time( NULL ) );
for (i=0; i<20; i++)
{
    klucz[i]=((rand() % 19 ) + 0 );
    cout << klucz[i] << " ";
}

```

```

a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length;
, b), b = b.rep
np_array.length; a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY

funkcja rand()

Wielkość tablicy kluczy pozostaje bez zmian, dlatego została zadeklarowana jako `klucz[20]`. Zmienna `i` jest potrzebna do sterowania pętlą, która wykonuje się 20 razy, za każdym razem wpisując do kolejnej pozycji tablicy (zależnej od `i`) losową wartość z przedziału od 0 do 19. Skąd jednak liczby losowe?

Liczby całkowite generuje funkcja `rand()`.

Aby uzyskać każdorazowo inną wartość, należy powiązać funkcję z czasem rzeczywistym wskazywanym przez zegar komputera. Stąd konieczność dołączenia biblioteki `ctime`. Dzięki niej będzie można użyć funkcji `time`, która zwraca czas zegara komputera jako liczbę. Użyta w funkcji `srand` ustawiającej wartość początkową liczby zapewni różne wartości dla każdej kompilacji programu. Zakres liczb (od 0 do 19) zapewnia `%19 + 0`.

W przykładowym programie dodano także strumieniowanie do konsoli wygenerowanej tablicy kluczy. Pozwoli to zaobserwować losowość jej wypełniania i weryfikację poprawności szyfrowania i deszyfrowania.

```

9 4 2 4 12 0 13 13 4 1 14 12 1 0 5 9 11 12 11 17
uczę się programowania w C++
uczę się programowania w C++
~g|i,sv+$q€{hrfvz?lwmc$? P8/
uczę się programowania w C++

```

```

13 13 5 5 0 2 9 4 18 10 15 2 0 10 16 8 14 1 9 13
uczę się programowania w C++
uczę się programowania w C++
,płd ur12z?qg|qu}xj{vf%w"L/=
uczę się programowania w C++

```

```

6 16 0 16 9 3 16 12 0 4 16 18 1 7 6 16 3 15 8 5
uczę się programowania w C++
uczę się programowania w C++
{szú)vyö t,?hyg}r†itya0€#57+
uczę się programowania w C++

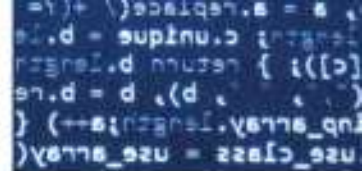
```

Rys. 15.6. Przykłady szyfrowania tekstu programem z zastosowaniem losowych kluczy

Dzięki losowości generowania tablicy kluczy odszyfrowanie tekstu bez jej znajomości jest bardzo trudne.

Obie metody zazwyczaj dają różne wartości dla tych samych znaków szyfrowanych. Wyjątek stanowi rzadki przypadek kodowania tej samej liczby tym samym kluczem.

Przykładowe programy, po szyfrowaniu, nie zmieniają liczby znaków.



ZADANIA DO ROZWIĄZANIA

1. Sprawdź dla dowolnego tekstu poprawność metody szyfrowania użytej w przykładach. Wyniki przedstaw w tabelach wzorowanych na tabelach 15.1. i 15.2. Użyj tablicy kodów ASCII.
2. Zmodyfikuj program ze stałymi kluczami w taki sposób, by wyświetlał w konsoli kody ASCII zaszyfrowanych tekstów. Na podstawie tabeli kodów sprawdź, czy funkcja szyfrująca i deszyfrująca działają prawidłowo. Sprawdzenia dokonaj na trzech dowolnych znakach tekstu.
3. Zmodyfikuj program ze stałymi kluczami w taki sposób, by ograniczyć liczbę kluczy w tablicy do 5. Obserwuj wyniki szyfrowania i porównaj z wynikami dla tablicy 20-elementowej. Czy szyfrowanie stało się mniej bezpieczne?
4. W programie z losowo wypełnianą tablicą kluczy zmień sposób wyświetlania wyników w konsoli w taki sposób, by informacja była pełna. Na ich podstawie użytkownik powinien bez wątpliwości wskazywać i nazywać wyświetlane dane.
5. Zmodyfikuj program z losowo generowanymi kluczami w taki sposób, by szyfrowanie i deszyfrowanie odbywało się dwukrotnie przy użyciu tych samych kluczy. Czy zwiększy to bezpieczeństwo szyfrowanych danych?

PODSUMOWANIE LEKCJI

- **Klucz w szyfrowaniu** jest informacją, dzięki której można zaszyfrować lub odszyfrować informację. s. 102
- **Kryptografia symetryczna** używa tego samego klucza do szyfrowania i deszyfrowania. Muszą go mieć zarówno nadawca, jak i odbiorca. s. 102
- **Kryptografia asymetryczna** używa kluczy publicznych i prywatnych. Klucz prywatny zna jedynie odbiorca. s. 102
- **Szyfrowanie asymetryczne** często jest stosowane do szyfrowania kluczy kryptografii symetrycznej. s. 103
- Teksty są przechowywane w postaci **kodów ASCII**. Teksty można szyfrować, dodając lub odejmując klucz w postaci liczby całkowitej do kodów ASCII poszczególnych znaków. s. 104
- Program może przekazać do funkcji także **zmienną tablicową**. s. 106
- Funkcja **rand()** generuje całkowite liczby losowe. s. 108



16. Automat wydaje, czyli jak dynamicznie wydawać resztę

NA TEJ LEKCJI:

- dowiesz się, na czym polega metoda zachłanna;
- użyjesz w programie zmiennych tablicowych.

Wrzucając pieniądze do automatu z napojami, nie musisz odliczać dokładnej kwoty. Urządzenie samo sumuje nominały, wydaje towar i resztę. Jak komputer w automacie oblicza resztę? Czy algorytm jest skomplikowany? Opracujmy go wspólnie. Nie będzie to bardzo trudne.

16.1. Zachłannienie, czyli wydajemy resztę z użyciem najmniejszej liczby nominałów

Przypomnij sobie, jak sprzedawca w sklepie wydaje resztę. Oczywiście może to zrobić, używając dowolnych nominałów. Robi to na tyle racjonalnie, by w kasie nie zabrakło żadnego z nich. Automat musi kierować się innymi kryteriami. Nie tylko kwota musi się zgadzać, lecz także wydawanie musi być optymalne, czyli z użyciem jak najmniejszej liczby wydawanych monet. Załóżmy, że ma zostać wydana kwota 13 zł i 60 groszy z użyciem najmniejszej liczby monet (rys. 16.1.).



Rys. 16.1. Optymalne wydanie reszty w kwocie 13,60 zł

Zastanów się teraz nad kwotami, które można wydawać za pomocą zestawu monet z rysunku 16.1. Jeśli jest ich dostatecznie dużo, możesz wydać dowolną kwotę pod warunkiem, że nie będzie ona zawierała pojedynczych groszy. Z tego wniosek, że dzięki odpowiednio dobranym w systemie monetarnym nominałom, można złożyć dowolną kwotę.

metoda zachłanna

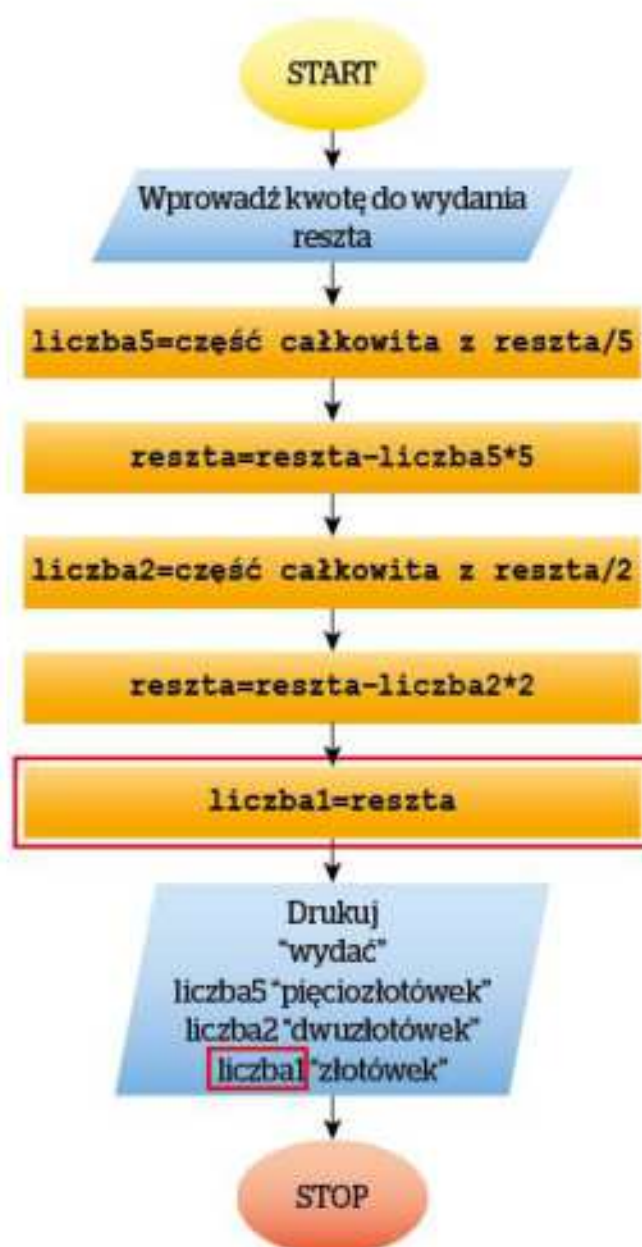
Wydawanie reszty (jak na rysunku 16.1.), czyli zaczynając od największej, możliwej liczby najwyższych nominałów, nazywamy metodą zachłanną. Jeśli przeanalizujesz tę metodę, zapewne dojdziesz do wniosku, że nie sprawdzi się ona dla nietypowego zestawu nominałów. Gdybyśmy chcieli kwotę 6 złotych wydać za pomocą monet dwu- i pięciozłotowych, algorytm zachłanny wydałby najpierw

monetę pięciozłotową, bo jej wartość mieści się w 6. Kasjerka w sklepie wydałaby trzy monety dwuzłotowe. Ale ona nie stosuje algorytmu zachłannego.

Przystępując do opracowania algorytmu wydawania reszty metodą zachłanną, musimy założyć, że do dyspozycji są wszystkie dostępne nominały w odpowiedniej ilości. Dla uproszczenia w przykładzie wykorzystamy jedynie nominały 1, 2 i 5 złotych oraz założymy, że wydajemy resztę niezawierającą groszy (rys. 16.2.). W przykładowym algorytmie użyto następujących zmiennych:

reszta – zmienna zawierająca kwotę do wydania. W trakcie procesu wydawania zmienia wartość na równą kwocie, która pozostała do wydania.

`liczba5`, `liczba2`, `liczba1` to zmienne, w których ostatecznie zostanie zapisana liczba poszczególnych nominatów do wydania.



Rys. 16.2. Przykład algorytmu wydającego resztę metodą zachłanną dla reszty wyrażonej liczbą całkowitą

Algorytm nie wymaga omówienia. Zastanów się jednak, czy blok zaznaczony czerwoną ramką jest konieczny do otrzymania prawidłowego wyniku. Od tego miejsca wartość zmiennej `reszta` nie zmienia się do końca działania algorytmu. Może więc być wyświetlona jako liczba wydawanych złotych, czyli



indeks 0

zmienna `liczba1` staje się niepotrzebna. Właśnie dokonaliśmy weryfikacji optymalności kodu programu.

Modyfikacja wiąże się także ze zmianą w bloku wyjściowym – jako liczbę wydawanych złotych należy wpisać wartość zmiennej `reszta`. Może ona mieć wartość 0 dla parzystych kwot do wydania lub 1 dla nieparzystych. Na tym przykładzie widać, że każdy algorytm powinien być weryfikowany nie tylko pod kątem funkcjonalności (daje prawidłowe wyniki czy nie), lecz także pod kątem jego budowy i wykorzystania zasobów komputera. W tym przypadku dodatkowa zmienna spowoduje zarezerwowanie większego obszaru pamięci.

Program realizujący algorytm z rysunku 16.2. wykorzystuje fakt, że zmienna zadeklarowana jako `int` przechowująca wynik dzielenia będzie zawierała jedynie jego część całkowitą. Nie ma więc potrzeby korzystania z odpowiedniej funkcji (`div`) podającej wartość całkowitą liczby (rys. 16.3.).

```

1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int liczba5=0, liczba2=0, liczba1=0, reszta;
7 int main ()
8 {
9     cout << "Podaj kwotę do wydania: ";
10    cin >> reszta;
11
12    liczba5=reszta/5;
13    reszta=reszta-liczba5*5;
14    liczba2=reszta/2;
15    reszta=reszta-liczba2*2;
16
17    cout << "do wydania" << endl;
18    cout << liczba5 << " szt. pięciozłotówek" << endl;
19    cout << liczba2 << " szt. dwuzłotówek" << endl;
20    cout << reszta << " szt. złotych" << endl;
21    return 0;
22 }

```

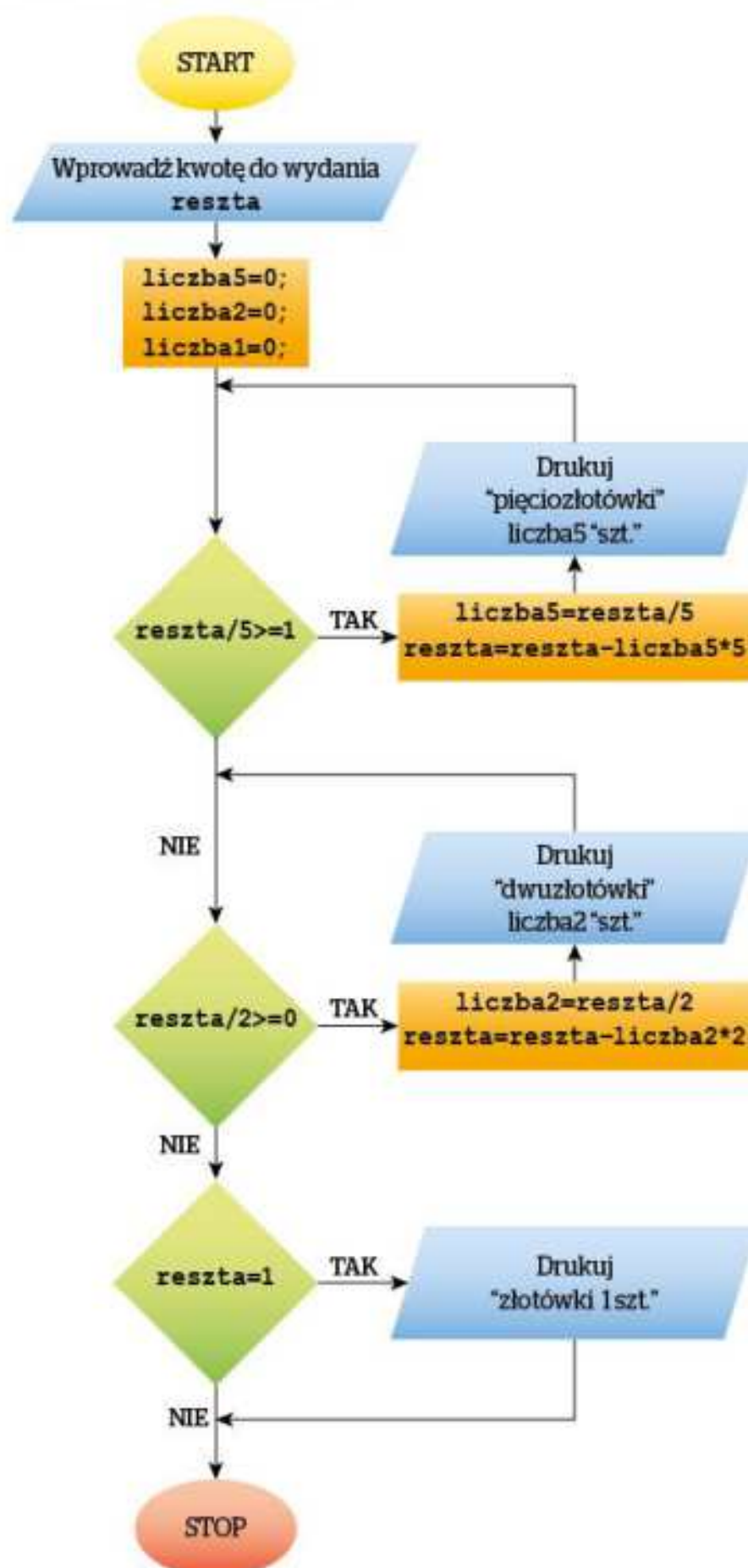
Rys. 16.3. Program wydający resztę według przykładowego algorytmu zachłannego

Podaj kwotę do wydania: 23 do wydania	Podaj kwotę do wydania: 10 do wydania
4 szt. pięciozłotówek	2 szt. pięciozłotówek
1 szt. dwuzłotówek	0 szt. dwuzłotówek
1 szt. złotych	0 szt. złotych

Rys. 16.4. Wyniki działania programu wydającego resztę metodą zachłanną dla kwoty 23 złote i 10 złotych

Program (rys. 16.3.) będzie zawsze wyświetlał liczbę wszystkich nominałów, nawet gdy nie będą używane do wydania reszty (rys. 16.4.). Zmodyfikujmy algorytm i program.

Dzięki zastosowaniu badania warunków można opuścić drukowanie i obliczenia dla nominałów większych od pozostałej do wydania kwoty (rys. 16.5.). W programie ułożonym według tego algorytmu zastosowano badanie warunków za pomocą instrukcji `if` (rys. 16.6.).



Rys. 16.5. Inne rozwiązanie algorytmu dla metody zachłannej wydawania reszty



```

1 #include <iostream>
2 #include <stdlib.h>
3
4 using namespace std;
5
6 int liczba5=0, liczba2=0, liczba1=0, reszta;
7 int main ()
8 {
9     cout << "Wprowadź kwotę do wydania: ";
10    cin >> reszta;
11    cout << endl;
12    if (reszta/5>=1)
13    {
14        liczba5=reszta/5;
15        reszta=reszta-liczba5*5;
16        cout << "pięciozłotówki - " << liczba5 << " szt."<< endl;
17    }
18
19    if (reszta/2>=1)
20    {
21        liczba2=reszta/2;
22        reszta=reszta-liczba2*2;
23        cout << "dwuzłotówki - " << liczba2 << " szt." << endl;
24    }
25
26    if (reszta==1)
27        cout << "złotówki - 1 szt.";
28
29    return 0;
30 }

```

Rys. 16.6. Przykładowy program realizujący algorytm z rysunku 16.5.

Wprowadź kwotę do wydania: 11

pięciozłotówki - 2 szt.
złotówki - 1 szt.

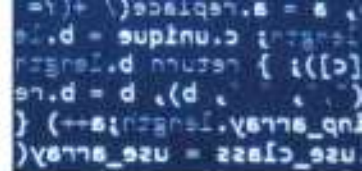


Wydanie 11 złotych
metodą zachłanną nie
przewiduje dwuzłotówek.

Rys. 16.7. Wynik działania programu z rysunku 16.6.

16.2. Nominały w tablicach, czyli jedna zmienna, wiele wartości

W obiegu znajdują się zarówno monety, jak i banknoty. Program komputerowy może odpowiedzieć sprzedawcy, ile i jakich nominałów ma użyć do wydania reszty. W Polsce w obiegu są następujące nominały banknotów: 500 zł, 200 zł, 100 zł, 50 zł, 20 zł, 10 zł oraz monet: 5 zł, 2 zł, 1 zł, 50 gr, 20 gr, 10 gr, 5 gr, 2 gr i 1 gr. Jedynie banknotem pięćsetzłotowym nie da się wydawać reszty. Gdyby dla wszystkich pozostałych 14 nominałów budować algorytm podobny do tego z rysunku 16.3., należałoby rozbudować go o kolejnych 11 dzieleń lub warunków.



Można jednak posłużyć się tablicami, w których zostaną zapisane wszystkie nominały i użyć bloku warunkowego (rys. 16.8.). Spowoduje to, że algorytm po odpowiednich zmianach w tablicy `nominaly` będzie wydawał także resztę dla nominałów używanych na przykład w Stanach Zjednoczonych, w których używa się monety 25 centów, a nie ma monet o nominałach 2 i 20 centów.

Przypomnienie

Odśwież swoją wiedzę na temat zmiennych tablicowych. Tablica jest zmienną, która może przechowywać wiele wartości. Każda pozycja tablicy jest indeksowana (od 0 do n), to oznacza, że ma swój numer. W związku z tym podczas deklaracji należy podać jej rozmiar. W C++, jeśli podczas deklaracji tablicy zostanie ona wypełniona danymi, podawanie rozmiaru nie jest konieczne. Podobnie jak inne zmienne tablice muszą mieć nazwę i określony typ danych, jakie będą przechowywały.

Przykład:

```
int tablica[]={20000,10000,5000,2000,1000,
500,200,100,50,20,10,5,2,1};
```

lub inaczej

```
int tablica[13];
```

W pierwszym przypadku tablica została wypełniona 14 danymi. W drugim – ta sama tablica musi być wypełniana danymi w czasie realizacji programu. Oczywiście w obu przypadkach program może wpisywać do nich nowe wartości. W przykładzie `tablica[0]` będzie miała wartość 20000, `tablica[4]` 1000, a `tablica[8]` to 50. Jak widać, indeks wskazuje bezpośrednio pozycję w tablicy.

Można zadeklarować tablice o większej liczbie wymiarów.

Zmienne tablicowe są nazywane także macierzami.

zmienne tablicowe

Wróćmy do naszego zadania. Algorytm (rys. 16.8.) w zasadniczej części nadal realizuje metodę zachłanną wydawania reszty. Tym razem jednak używa do tego wszystkich możliwych nominałów. W przykładowym algorytmie obliczona liczba poszczególnych nominałów jest drukowana tuż po obliczeniu i nie jest przechowywana.

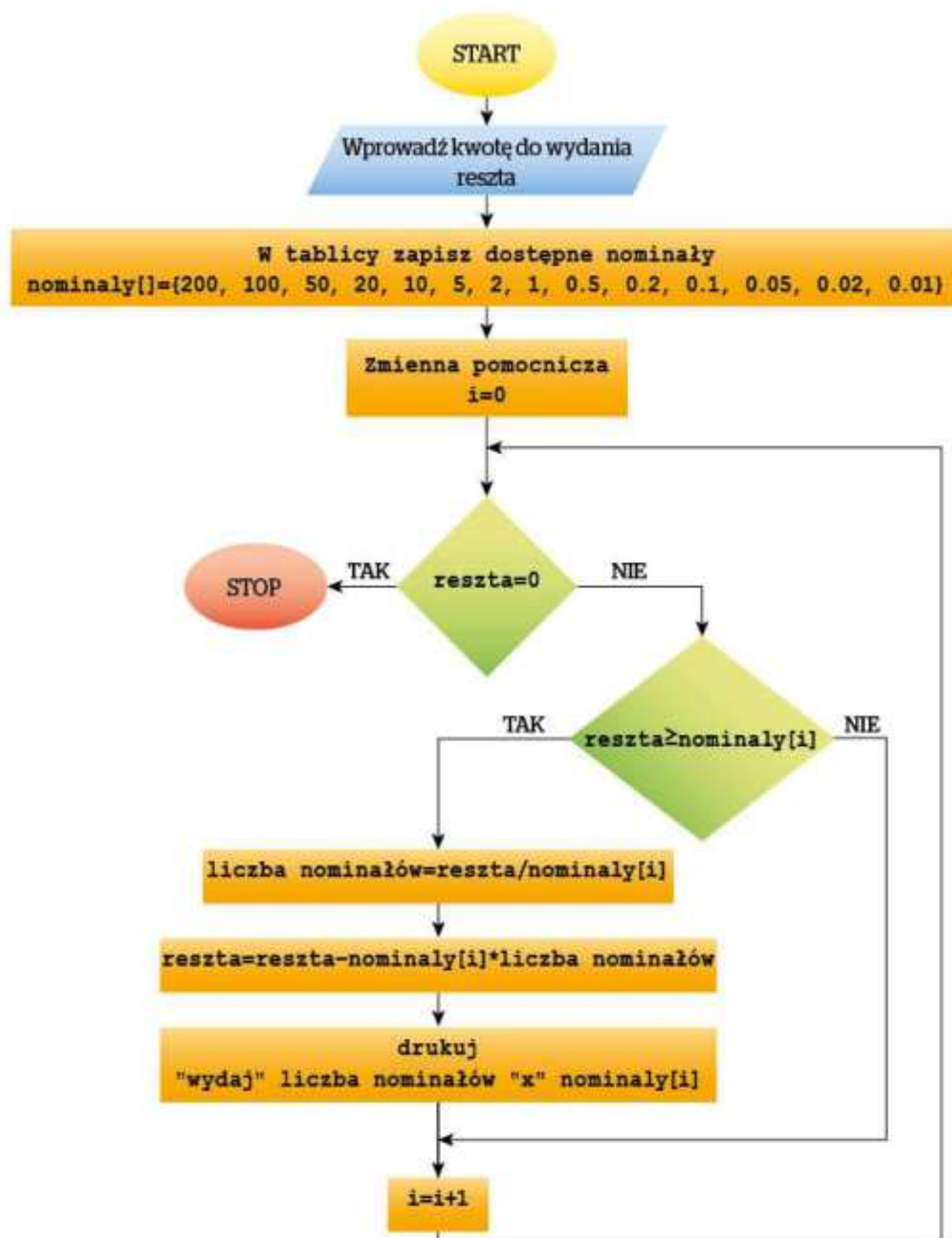
Prześledźmy działanie algorytmu na przykładzie. Załóżmy, że kwota do wydania to 40,10 zł. Taką też wartość zapiszmy pod zmienną `reszta`.

```

a.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b

```

III. PROGRAMOWANIE I ALGORYTMY



Rys. 16.8. Przykładowy algorytm wydawania reszty metodą zachłanną używający 14 nominałów



Tab. 16.1. Analiza działania algorytmu wydawania reszty metodą zachłanną dla kwoty 40.10 zł

Krok	Działanie	Wartości
1.	Wprowadzenie kwoty do wydania	reszta = 40.10
2.	Wypełnienie tablicy nominałami	nominały[13]={200, 100, 50, 20, 10, 5, 2, 1, 0.5, 0.2, 0.1, 0.05, 0.02, 0.01}
3.	Indeks tablicy – zmienna i – ustawiony na pierwszej danej tablicy	i = 0
4.	Czy reszta jest już wydana	reszta=40.1 ≠ 0 kierunek – NIE
5.	Czy kwota do wydania zawiera w sobie przynajmniej jedną wartość nominału 200	nominały[0]=200 40.1 < nominały[0] kierunek – NIE
6.	Indeks tablicy ustawia się na następny nominał	i=0+1=1
7.	Czy reszta jest już wydana	reszta=40.1 ≠ 0 kierunek – NIE
Kroki 4, 5, 6, 7 są wykonywane do momentu, gdy i = 3		
8.	Reszta do wydania jest większa od 20	reszta ≥ nominały[3]=20 kierunek TAK
9.	Obliczanie liczby dwudziestozłotówek	liczbanominałów=40.1/20=2 Nie ma części ułamkowej, ponieważ zmienna jest typu int
10.	Obliczanie pozostałej kwoty po wydaniu 20 zł	reszta=40.1-(20*2)=0.1 zł
11.	Wydruk liczby dwudziestozłotówek	Wydaj 2 x 20
12.	Indeks następnego nominału	i=3+1=4
13.	Reszta jeszcze nie wydana	reszta=0.1 ≠ 0 kierunek – NIE
Dla nominałów od 4 do 9 wartość warunku reszta ≥ nominały[i] – NIE		
14.	i=10	0.1 ≥ nominały[10]=0.1 kierunek TAK
15.	Obliczanie liczby dziesięciogroszówek	liczbanominałów=0.1/0.1=1
16.	Obliczanie pozostałej kwoty po wydaniu 2x20 zł i 10 gr	reszta=0.1-0.1*1=0
17.	Wydruk liczby dziesięciogroszówek	Wydaj 1 x 0.1
18.	Cała reszta wydana	Warunek reszta=0 spełniony Koniec obliczeń



Realizacja algorytmu (rys. 16.8.) w języku C++ napotyka na pewne trudności. Otóż część nominałów jest mniejsza od 1. Z tej przyczyny tablicę powinno się zadeklarować jako zmienną typu `float`. Pozostała kwota do wydania (zmienna `reszta`) także musi być liczbą rzeczywistą. Gdy do wydania zostaje 0,01 złotego, wartość zmiennej `reszta` powinna wynosić 0,01. Niestety komputer oblicza wartość zmiennej `reszta` z pewnym przybliżeniem, co spowoduje błędny wynik dla kwot do wydania o nieparzystej liczbie groszy. W programie zastosujemy więc zaokrąglenie do setnych części złotego. W programie (rys. 16.9.) została użyta funkcja `round` powodująca zaokrąglenie wyniku do części całkowitej w sposób, który znasz z matematyki, na przykład dla liczby 0,6 da wynik 1, a dla 0,4 da 0. Jej użycie jest możliwe po dołączeniu biblioteki `math.h`. Zaokrąglenie wyniku obliczania reszty pozostałej do wydania do dwóch miejsc po przecinku realizuje linia 20.

```
reszta=round((reszta-(nominaly[i]*liczbanominalow))*100)/100;
```

Tablica nominałów będzie miała postać:

```
float nominaly[]={200,100,50,20,10,5,2,1,0.5,0.2,0.1,0.05,0.02,0.01}
```

Nie ma potrzeby podawania rozmiaru tablicy w `[ ]`, skoro podczas deklaracji wpisujemy do niej konkretne dane.

```
1 #include <iostream>
2 #include <stdlib.h>
3 #include <math.h>
4 using namespace std;
5
6 int liczbanominalow, i=0;
7 float nominaly[]={200,100,50,20,10,5,2,1,0.5,0.2,0.1,0.05,0.02,0.01}, reszta;
8
9 int main()
10 {
11     cout << "Podaj kwotę do wypłacenia: ";
12     cin >> reszta;
13
14     cout << endl << "wydaj następujące nominały:" << endl << endl;
15     while (reszta!=0)
16     {
17         if (reszta >= nominaly[i])
18         {
19             liczbanominalow=reszta/nominaly[i];
20             reszta=round((reszta-(nominaly[i]*liczbanominalow))*100)/100;
21
22             cout << nominaly[i] << " zł - " << liczbanominalow << " szt." << endl;
23         }
24         i++;
25     }
26     return 0;
27 }
```

Rys. 16.9. Przykładowy program wydający resztę metodą zachłanną korzystający z wszystkich nominałów



Podaj kwotę do wypłacenia: 444.31

wydać następujące nominały:

200 zł - 2 szt.

20 zł - 2 szt.

2 zł - 2 szt.

0.2 zł - 1 szt.

0.1 zł - 1 szt.

0.01 zł - 1 szt.

Rys. 16.10. Wynik działania programu wydającego resztę metodą zachłanną wykorzystującego wszystkie nominały

ZADANIA DO ROZWIĄZANIA

1. Jaką wartość może przybierać liczba wydawanych najniższych nominałów, np. 1 zł 1 gr? Uzasadnij swoją odpowiedź i podaj kilka przykładów.
2. Zmodyfikuj program z rysunku 16.3. w taki sposób, by wydawanie reszty realizowała funkcja. Zdefiniuj ją. Efekt działania programu (wyświetlanie komunikatów) powinien pozostać bez zmian.
3. Automaty, w których można dokonać zakupów, operują najczęściej nominałami od 5 zł do 10 groszy. Zmodyfikuj algorytm i program wydawania reszty w taki sposób, by mógł być zastosowany w takich urządzeniach.
4. Zmodyfikuj program z rysunku 16.9. w taki sposób, by można było wprowadzać kwotę reszty do wydania w złotych, np. 34,51 zł. Sprawdź, jak dokładność wykonywania obliczeń wpływa na działanie programu dla różnych kwot w tym 34,51 zł.
5. Zmodyfikuj program wydawania reszty z wykorzystaniem wszystkich dostępnych nominałów w USA.

PODSUMOWANIE LEKCJI

- **Metoda zachłanna** zakłada wydawanie reszty nominałami największymi z możliwych. Można opracować wiele algorytmów wykorzystujących metodę zachłanną, w tym także uwzględniających różne zakresy dostępnych nominałów. s. 110
- **Zmienne tablicowe** są zmiennymi, w których można przechowywać wiele wartości. Wszystkie są tego samego typu, deklarowanego podczas deklaracji tablicy. Jeśli podczas deklaracji zmiennej tablicowej jednowymiarowej wypełnimy ją danymi, nie ma potrzeby podawania jej rozmiaru. s. 115
- Dane zapisane w zmiennej tablicowej są indeksowane numerem miejsca w tablicy. Dla tablicy jednowymiarowej pierwsza ma **indeks 0**. s. 112

```

s.replace(/ +(?=
length; c.unique = b.len
c)); } return b.length;
, b), b = b.rep
np_array.length;a++) {
use_class = use_array(b

```

PODSUMOWANIE PRZED SPRAWDZIANEM

9. Przez 1 i siebie, czyli jak zbadać, czy dana liczba jest liczbą pierwszą

- s. 70 • **Metoda Sito Eratostenesa** powstała w starożytności i wyszukuje liczby pierwsze spośród liczb z zakresu od 2 do n .
- s. 71 • By zbadać, czy dana **liczba całkowita jest pierwsza**, można użyć metody dzielenia jej przez liczby całkowite i sprawdzania reszty.
- s. 71 • **Wielkie liczby pierwsze** odgrywają dużą rolę w kryptologii i szyfrowaniu danych komputerowych. Są pomocne przy tworzeniu szyfrów z kluczem publicznym.

10. Która pierwsza, czyli jak program bada liczby

- s. 79 • Program dający odpowiedź na pytanie, **czy liczba jest pierwsza**, zawiera **pętle**.
- s. 79 • **Pętle for i while** mogą być stosowane w różnych realizacjach tego samego algorytmu.
- s. 79 • **Modyfikacje** programów mogą zwiększyć szybkość ich działania.

11. Największy i najmniejszy, czyli jak znaleźć NWD i NWW

- s. 81 • Autorstwo metod obliczania NWD jest przypisywane **Euklidesowi**, a **algorytmy** nazwane jego imieniem.
- s. 81, 83 • Istnieje algorytm **niezoptymalizowany** i **zoptymalizowany**. Różnica w ilości powtórzeń pętli jest tym większa, im mniejszy względem najmniejszej z badanych liczb jest NWD.
- s. 84 • W programie szukającym NWD i NWW można użyć pętli **while**.

12. Dodawanie ułamków, czyli jak wykorzystać NWW i NWD w programie komputerowym

- s. 87 • **Algorytm dodawania ułamków** wykorzystuje algorytmy znajdowania NWW i NWD.
- s. 89 • **Kod znajdowania NWW i NWD** można zapisać w **postaci funkcji**. Dzięki temu możliwe będzie wielokrotne ich wykorzystanie bez konieczności umieszczania kodu obliczeń w każdym wymagającym obliczeń NWW i NWD miejscu programu.

13. Skracamy i wyłączamy, czyli dodawania ułamków ciąg dalszy

- s. 92 • Funkcja obliczająca NWD w programie skracającym postać ułamka będzie wykorzystana wielokrotnie.
- s. 94 • Wartości mianowników są **przekazywane do funkcji** obliczających NWW i NWD.
- s. 95 • W języku C++ wynik dzielenia zapisany pod zmienną zadeklarowaną jako **int** zawiera jedynie **część całkowitą** tego działania.

14. Szyfrowanie, czyli poznajemy szyfr Cezara

- s. 97 • **Szyfr Cezara** jest szyfrem przesuwającym wykorzystującym alfabet łaciński.
- s. 100 • **https** przed adresem internetowym oznacza szyfrowanie transmisji danych.

15. Konspiracja, czyli tworzymy własne szyfry

- s. 102 • Z **kryptografią symetryczną** mamy do czynienia wtedy, gdy wykorzystuje się **klucz** znany nadawcy i odbiorcy.
- s. 102 • **Metoda asymetryczna** polega na stosowaniu **kluczy publicznych i prywatnych**.
- s. 103 • Do przesyłania tablicy kluczy można użyć szyfrowania asymetrycznego.
- s. 106 • Użycie **tablicy kluczy** do szyfru przesuwającego utrudnia deszyfrację, ponieważ dla każdej pozycji szyfrowanego tekstu może występować inny klucz.
- s. 108 • Tablice kluczy mogą być tworzone **generatorem liczb losowych**.

16. Automat wydaje, czyli jak dynamicznie wydawać resztę

- s. 110 • Wydawanie reszty w opisany sposób, czyli zaczynając od największej możliwej liczby najwyższych nominałów nazywamy **metodą zachłanną**.
- s. 115 • Do wydawania reszty wszystkimi nominałami można zastosować **tablicę**.





IV. Grafika i druk

- ▶ Druk 3D
- ▶ Projektowanie 3D
- ▶ Wypukłe napisy w edytorze GIMP
- ▶ Działanie skanera
- ▶ Ekspozycja w fotografii
- ▶ Przygotowanie do sesji zdjęciowej
- ▶ Kadrowanie
- ▶ Edytowanie zdjęcia w warstwach
- ▶ Filmowanie
- ▶ Strumieniowanie danych

17. Modele w przestrzeni, czyli podstawy druku 3D

NA TEJ LEKCJI:

- poznasz zasadę działania drukarki 3D;
- dowiesz się, po co potrzebne są modele 3D.

Drukarki 3D przestały być wyłącznie wyposażeniem laboratoriów i biur konstrukcyjnych. Znalazły zastosowanie w wielu dziedzinach życia, a coraz częściej można je spotkać także w domach. Są w stanie wydrukować praktycznie każdy przedmiot od małej figurki do domu mieszkalnego lub mostu. Planuje się wykorzystać tę technologię w budowie baz na Księżycu i planetach. O wiele łatwiej dowieźć tam materiały budowlane niż całe, gotowe moduły. To jednak odległa przyszłość. Na szczęście nawet amator drukowania 3D bez problemu znajdzie odpowiedni program i nauczy się nim posługiwać. Wielu z nich można używać za darmo. Jakie powinny spełniać wymagania i skąd brać inspiracje do projektów? Jak działają drukarki 3D? Tego dowiesz się w rozdziale.

17.1. Warstwa na warstwie, czyli jak działa drukarka 3D

Wyobraź sobie, że zbudowałeś/zbudowałaś z klocków jakiś obiekt, na przykład pałac (rys. 17.1.). A teraz pomyśl, jak rozwiązać problem zbudowania takiego samego obiektu przez twoją koleżankę lub kolegę w tym samym czasie, lecz w oddalonym miejscu – przykładowo w ich mieszkaniu. Czy umiesz ułożyć algorytm opisujący kolejność czynności i gwarantujący zbudowanie identycznych pałaców?



Rys. 17.1. Pałac z klocków z kolorowymi warstwami elementów

Przyjmijmy, że budowniczcy będą dysponować takim samym zestawem klocków, a algorytm zapiszemy w postaci listy kroków.

Założenia algorytmu:

1. Kolejne czynności opisz jako układanie warstw klocków.
2. Warstwy klocków będą układane od lewej strony do prawej.
3. Przesunięcie rozpoczęcia warstwy względem niższej liczone jest liczbą klocków konstrukcyjnych.

Ułóż samodzielnie taki algorytm, uwzględniając założenia i porównaj z poniższym.

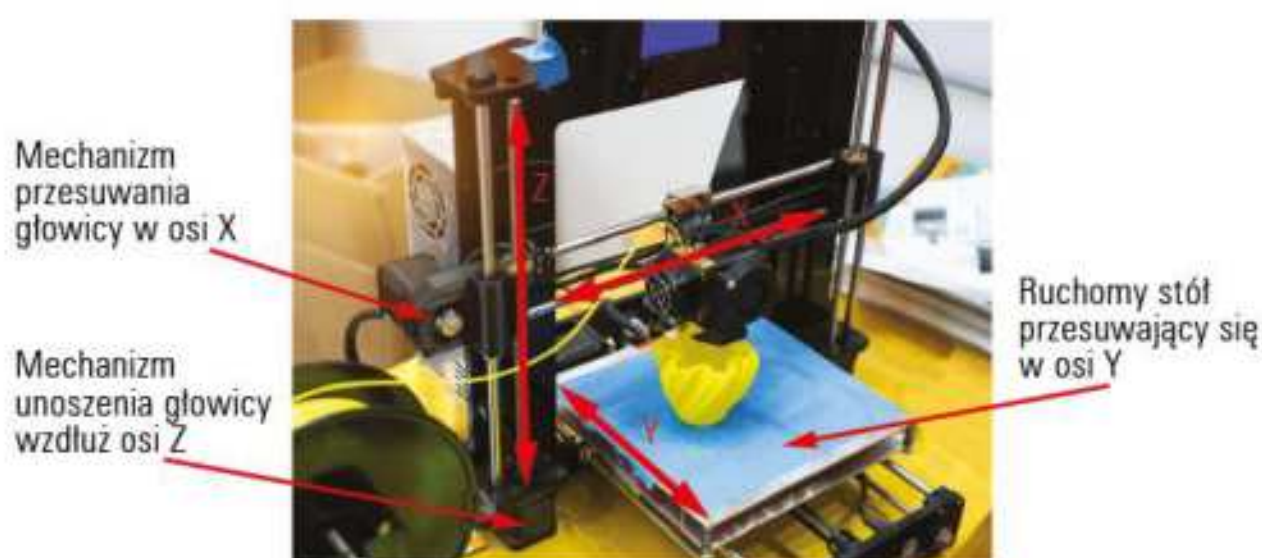


Algorytm budowania pałacu z rysunku 17.1.

1. Z sześciu czerwonych klocków utwórz pierwszą warstwę (od lewej do prawej), układając je jeden obok drugiego.
2. Z żółtych klocków ułóż drugą warstwę, zaczynając z przesunięciem o jeden rząd klocków konstrukcyjnych w prawo i kończąc o jeden przed końcem pierwszej warstwy.
3. Z zielonych klocków ułóż trzecią warstwę, zaczynając z przesunięciem o jeden rząd klocków konstrukcyjnych w prawo i kończąc o jeden przed końcem trzeciej warstwy.
4. Z czerwonych klocków ułóż czwartą warstwę, zaczynając z przesunięciem o dwa rzędy klocków konstrukcyjnych w prawo i kończąc o dwa przed końcem pierwszej warstwy.
5. Na środku czwartej warstwy ułóż zielony klocek tworzący piątą warstwę.
6. Ułóż klocek żółty na zielonym (szósta warstwa).
7. Siódmą warstwę utwórz z klocka niebieskiego zakończonego kopułką.

Wspólną cechą wszystkich punktów listy kroków jest układanie warstw. Tę własność wykorzystano także podczas tworzenia technologii druku 3D. Tam również układa się warstwę na warstwie według określonego algorytmu. Ponieważ druk ma być przestrzenny, warstwy nakłada się wzdłuż oś XYZ (rys. 17.2.).

Program komputerowy zamienia projekt (model) tworzony w edytorze grafiki 3D na zbiór warstw, które muszą być nałożone na siebie, by powstał obiekt. Podobnie wygląda wydruk na kartce, w którym obraz składa się z kolejnych linii na płaszczyźnie.



Rys. 17.2. Mechanizmy drukarki 3D

Dzięki złożeniu ruchów głowicy i podłoża można umieścić głowicę w dowolnym punkcie przestrzeni drukowania z uwzględnieniem najmniejszego możliwego skoku mechanizmów.

Uwaga!

Istnieją konstrukcje drukarek 3D, w których porusza się wyłącznie głowica, a podłoże pozostaje nieruchome.

filament
w drukarkach

Wiesz już, jak głowica osiąga pożądane położenie w przestrzeni drukowania. Pozostaje jeszcze rozwikłać tajemnicę jej działania.

Głowica drukarki 3D rozgrzewa filament (rys. 17.3.). Topi go i nakłada warstwę na warstwę. Filament szybko twardnieje i tworzy drukowany obiekt. Wadą takiego rozwiązania jest długi czas druku, a zaletą – możliwość stosowania filamentów z różnych materiałów, od tworzyw sztucznych aż po metale (rys. 17.4.).

głowica
w drukarkach 3D



Rys. 17.3. Filament i głowica drukarki 3D



Rys. 17.4. Różnokolorowe filamenty do druku 3D

17.2. Edytory i skanowanie, czyli skąd się biorą modele 3D

Drukarka, choćby najlepsza, niczego nie wydrukuje, jeśli nie będzie miała cyfrowego modelu wytworzonego za pomocą komputera.

model 3D

Ten może powstać w edytorze grafiki 3D lub z zastosowaniem skanera 3D. Profesjonalne występują w wersji stołowej (rys. 17.5.) i ręcznej, w której obiekt pozostaje nieruchomy. Za pomocą ręcznych skanerów skanuje się obiekty o dużych gabarytach oraz takie, które nie mogą być przenoszone, jak eksponaty muzealne.



Najtańszym rozwiązaniem dla amatorów i pasjonatów modelowania 3D są aplikacje na smartfony. Jakość powstałych modeli nie jest wysoka, ale po ich edycji i korekcie można uzyskać zadowalające efekty. Informacje o tych aplikacjach bez trudu znajdziesz w sieci.



Rys. 17.5. Skaner i edytor grafiki 3D

Skanery 3D działają nie tylko w paśmie promieniowania widzialnego. Istnieją takie, które potrafią wiernie skopiować kształt organów ludzkiego ciała. Modele są następnie drukowane i wykorzystywane przykładowo do ćwiczeń przed skomplikowaną operacją.

Skany 3D można także bez trudu zamówić w wyspecjalizowanych firmach.

A co, jeśli nie chcemy drukować modeli rzeczywistych obiektów? Nie wszystkie edytory grafiki 3D nadają się do tworzenia modeli od podstaw. Jakie powinny spełniać warunki? Domyślasz się, że oprócz edycji w trzech wymiarach powinny pozwolić zapisywać pliki wynikowe zgodne z obsługiwanyymi przez drukarki. Jest ich wiele. Im więcej obsługuje program, tym lepiej, jednak najlepiej sprawdzić, z jakimi współpracuje posiadana drukarka. Listę znajdziesz w instrukcji.

STL – w tym formacie geometria obiektu jest zbudowana z trójkątów. Mimo wad to najpowszechniej stosowany format pliku dla drukarek 3D.

OBJ – format przeznaczony dla programów do animacji 3D. Geometria obiektu jest przedstawiana za pomocą wierzchołków wielokątów, przy czym każdy z punktów w zapisie występuje tylko raz. OBJ stał się uniwersalnym sposobem zapisu plików z obiektami 3D. Powszechnie wykorzystuje się go do druku przestrzennego.

VRML – zaawansowany format uwzględniający kolory, fakturę, przezroczystość i inne parametry powierzchni. Ze względu na swoje zalety i darmową licencję również cieszy się popularnością wśród użytkowników drukarek 3D.

3MF – format opracowany na podstawie założeń i rozwiązań firmy Microsoft, wspierany przez wiele wiodących firm zajmujących się techniką projektowania i drukowania. Charakteryzuje się niewielkim rozmiarem.

Który program do edycji modeli 3D wybrać? Wszystko zależy od potrzeb. Początkujący użytkownicy zapewne będą szukać darmowych i łatwych w obsłudze programów, z niewielkimi wymaganiami sprzętowymi. Przykłady zostały zebrane w poniższej tabeli.



Tab. 17.1. Przykłady programów do edycji obiektów 3D

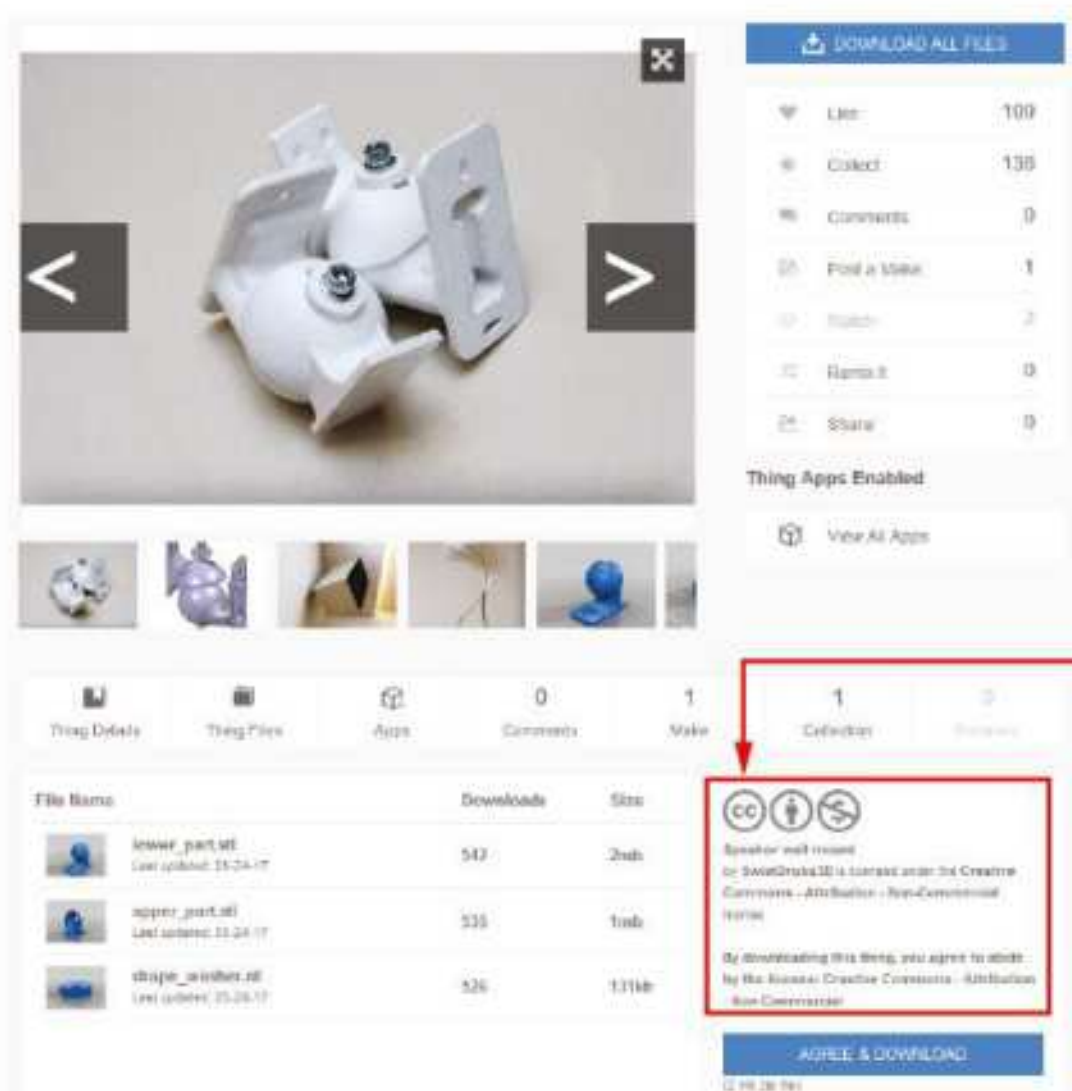
Nazwa programu	Opis	System operacyjny
Tinkercad	To działający w chmurze łatwy w obsłudze program dla początkujących. Choć ma niewielką funkcjonalność, wystarcza do stworzenia prostych projektów. Wymaga założenia konta, które daje dostęp także do innych programów on-line producenta. Program eksportuje do formatów OBJ i STL. Przygotowano edycję w języku polskim.	on-line
DesignSpark	Program udostępniono w wersji całkowicie darmowej. Eksportuje pliki w popularnych formatach. Wymaga darmowej rejestracji. Menu przygotowano w języku polskim. Przygotowano także wersję komercyjną tego programu.	Windows
BRL-CAD	To zaawansowany, darmowy program do projektowania CAD (ang. Computer Aided Design). Jego możliwości są zbliżone do oferty profesjonalnych edytorów. Eksportuje pliki w popularnych formatach.	Windows, OS X, Linux
FreeCAD	Kolejny program klasy CAD o stosunkowo łatwej obsłudze. Jest darmowy. Nie udostępniono polskiej wersji.	Windows, OS X, Linux

17.3. Udostępniane w sieci, czyli nie wszystko musisz robić od podstaw

projekty dla drukarek

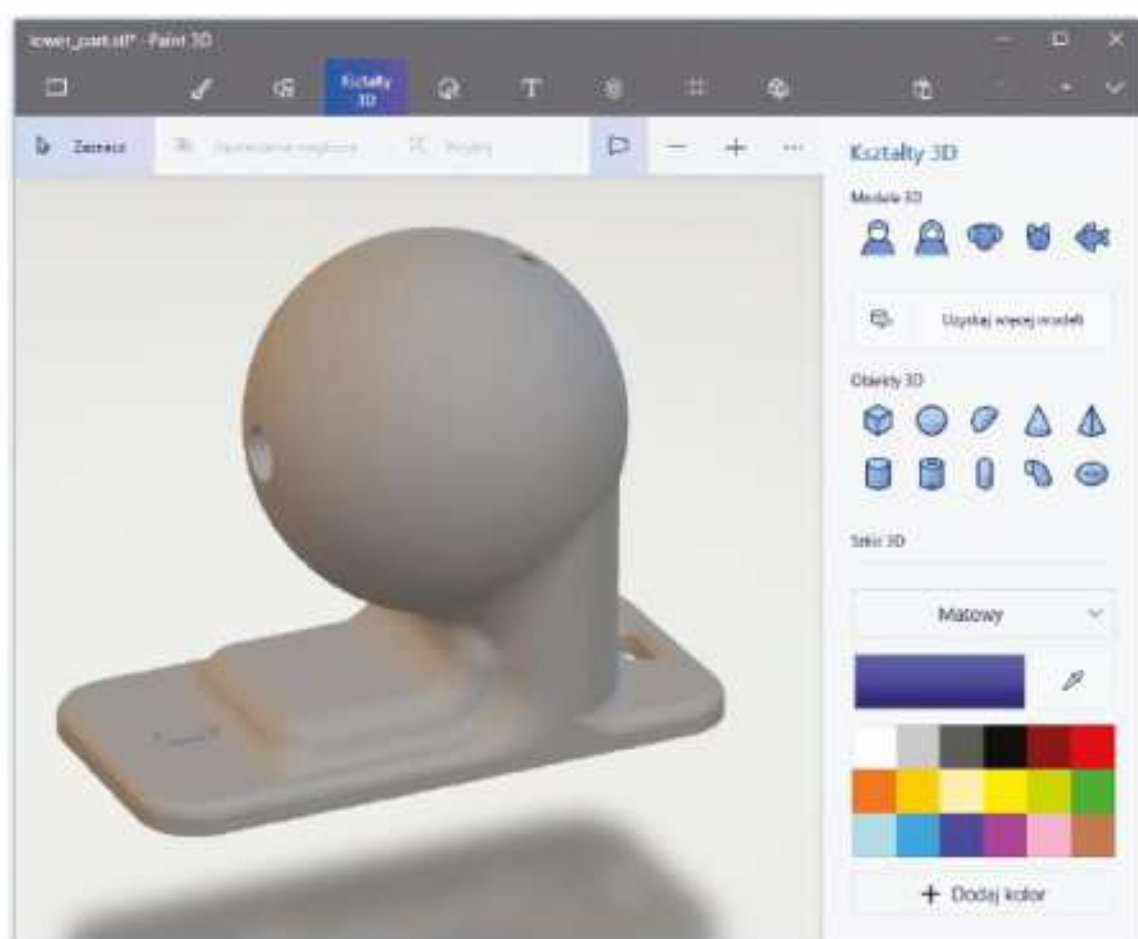
Projekty dla drukarek 3D są zapisywane w plikach o standardowych parametrach. Nie trudno się domyślić, że są one dostępne także w sieci. Niektóre są oferowane za darmo, a inne za opłatą. Jeśli istnieje potrzeba wykonania jakiegoś nietypowego elementu, można go zamówić w specjalistycznej firmie. Dla początkujących najciekawszą ofertę stanowią oczywiście projekty gotowe do natychmiastowego wykorzystania. Możesz wykorzystać projekty darmowe lub komercyjne sprzedawane w sklepach internetowych. Przed ich pobraniem należy zwrócić uwagę na to, czy nie naruszają praw autorskich i praw majątkowych (rys. 17.6.).

Projekty różnią się od ilustracji na stronie, z której zostały pobrane. Jak przeglądać pobrane pliki? Można to zrobić, wczytując projekt do edytora 3D, na przykład użytkownicy Windows skorzystają z nowej wersji programu Paint oznaczonego jako 3D (rys. 17.7.).



CC oznacza licencję Creative Commons. Taki projekt można wykorzystać bez opłat. Pozostałe ikony dotyczą zakazu wykorzystywania projektu do celów komercyjnych.

Rys. 17.6. Przykładowy projekt dla drukarki 3D dostępny w sieci



Rys. 17.7. Przeglądanie pobranego pliku STL w programie Paint 3D

**Uwaga!**

Część oferowanych w sieci modeli do druku 3D jest rozprowadzana nielegalnie. Dotyczy to głównie figurek postaci z filmów i gier. Jeśli masz podejrzenia co do legalności plików, nie pobieraj ich i nie rozpowszechniaj.

ZADANIA DO ROZWIĄZANIA

1. Znajdź w sieci blogi użytkowników drukarek 3D. Zapoznaj się z opisywanymi problemami i sukcesami. Wynotuj parametry, na które musisz zwrócić uwagę podczas wyboru modelu drukarki. Wybierz drukarkę 3D w cenie do 3000 złotych. Uzasadnij swoją decyzję analizą jej parametrów i opiniami umieszczonymi na forach dyskusyjnych.
2. Na podstawie znalezionych w sieci informacji (cena filamentu, projektu itp.) oblicz koszt wydruku dowolnego modelu znalezionego w sieci. Sprawdź, czy w opisie autor podał dodatkowe dane do oszacowania kosztu wydruku.
3. Znajdź w sieci projekty 3D udostępniane na licencji CC. Wybierz jeden z nich. Pobierz i wczytaj do Paint 3D. Sprawdź, w jakich formatach możesz zapisać projekt z tego programu i wskaż, który z nich może służyć do bezpośredniego wydruku na drukarce.
4. Znajdź w sieci i obejrzyj filmy pokazujące działanie ręcznych i stolowych skanerów 3D. Opisz krótko czynności przygotowujące do skanowania i skanowanie obiektów. Wymień zalety jednego i drugiego rozwiązania.
5. Znajdź w sieci informacje i opinie na temat edytorów wymienionych w tabeli 17.1. Wybierz jeden z nich i uzasadnij swoją decyzję.

PODSUMOWANIE LEKCJI

- s. 124 • **Drukarka 3D** nakłada kolejno na siebie warstwy roztopionego filamentu. **Filament** jest podawany do głowicy w postaci elastycznego pręcika.
- s. 124 • W drukarkach 3D **głowica może poruszać się w dwóch kierunkach**. Trzeci kierunek zapewnia ruchoma podstawa, na której powstaje wydruk. Istnieją rozwiązania techniczne drukarek 3D z głowicą poruszającą się w trzech kierunkach i nieruchomą podstawą.
- s. 124 • **Modele (obiekty) do drukowania** w trzech wymiarach tworzy się za pomocą edytora grafiki trójwymiarowej lub poprzez skanowanie 3D.
- s. 126 • **Gotowe projekty można pobrać z sieci za darmo** (na licencji CC) lub kupić w sklepie internetowym.



18. Modelujemy, czyli jak projektować obiekty 3D

NA TEJ LEKCJI:

- poznasz chmurę i aplikację Tinkercad;
- stworzysz swój pierwszy projekt dla drukarki 3D.

Wiesz już, jak działają drukarki 3D i które programy służą do edycji modeli. Umiesz także ich szukać i importować je w odpowiednim formacie. Czas na samodzielną edycję i stworzenie prostego modelu, który będzie można wydrukować na drukarce 3D. Skorzystasz z dwóch darmowych edytorów z polskim menu i możliwością eksportu pliku w najpopularniejszych formatach.

18.1. Bez instalacji, czyli edytor 3D w chmurze

Z pewnością znasz wiele programów on-line. Skoro istnieją edytory rozmaitych dokumentów, to warto poszukać także tych do modelowania 3D. Wśród nich znajdziesz **Tinkercad** ze strony www.tinkercad.com (rys. 18.1.). Na stronie startowej trzeba założyć konto niezbędne do korzystania z edytora. Zanim jednak to zrobisz, możesz zapoznać się z właściwościami programu i tutorialami pokazującymi najważniejsze narzędzia edytora. Możliwe jest też przejrzanie galerii gotowych projektów dostępnych dla każdego z użytkowników.

Tinkercad



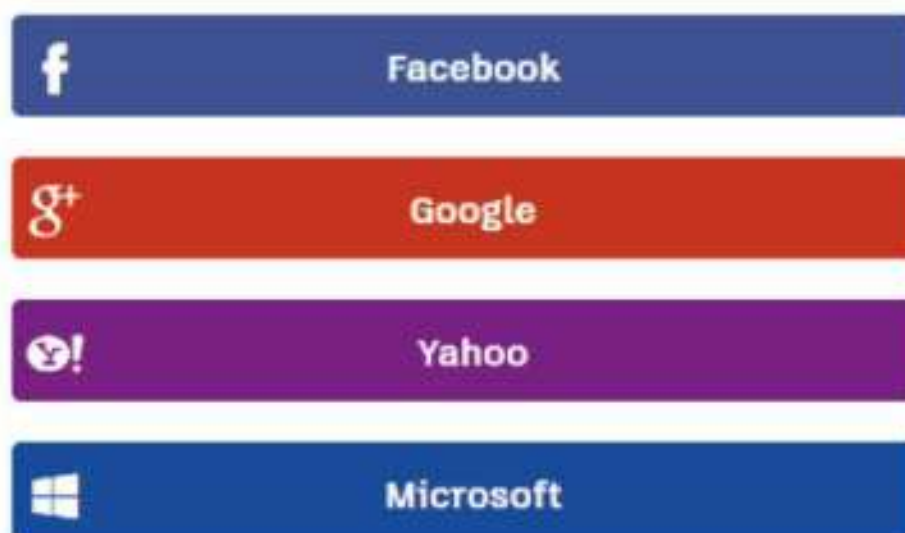
Rys. 18.1. Okno startowe edytora grafiki 3D on-line

Korzystając z Tinkercad, poznasz niektóre mechanizmy i nabędziesz nawyki, które w przyszłości pomogą ci w pracy z bardziej rozbudowanymi edytorami. Edycja w trzech wymiarach znacznie różni się od klasycznej na płaszczyźnie. Zaczniemy od najprostszych czynności na przykładzie modyfikowania gotowego projektu.



Zakładanie konta nie sprawi ci trudności. Jeśli nie chcesz zakładać konta w chmurze, możesz zalogować się poprzez jedną z usług społecznościowych (rys. 18.2.).

< Zaloguj się przy użyciu



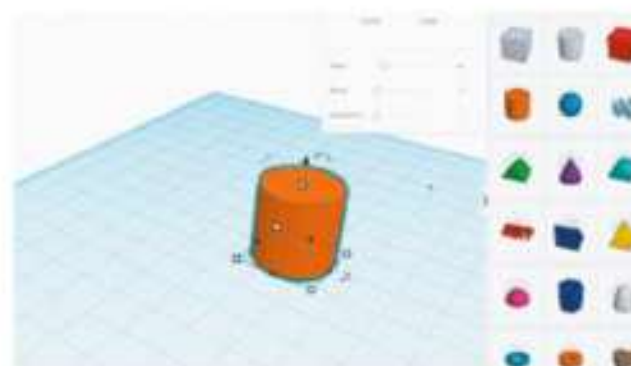
Rys. 18.2. Portale społecznościowe, które mogą służyć do logowania do Tinkercad

Przed przystąpieniem do pracy warto skorzystać z pomocy oferowanej przez chmurę. Dzięki temu szybko opanujesz podstawowe czynności i posługiwanie się narzędziami (rys. 18.3.).

Łatwe pod każdym względem

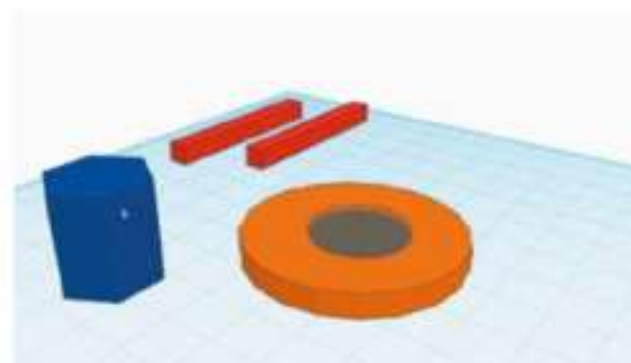
Umieszczanie kształtu

Kształty są blokami konstrukcyjnymi Tinkercad. Można dodawać istniejące kształty lub importować własne



Dopasowywanie obiektów

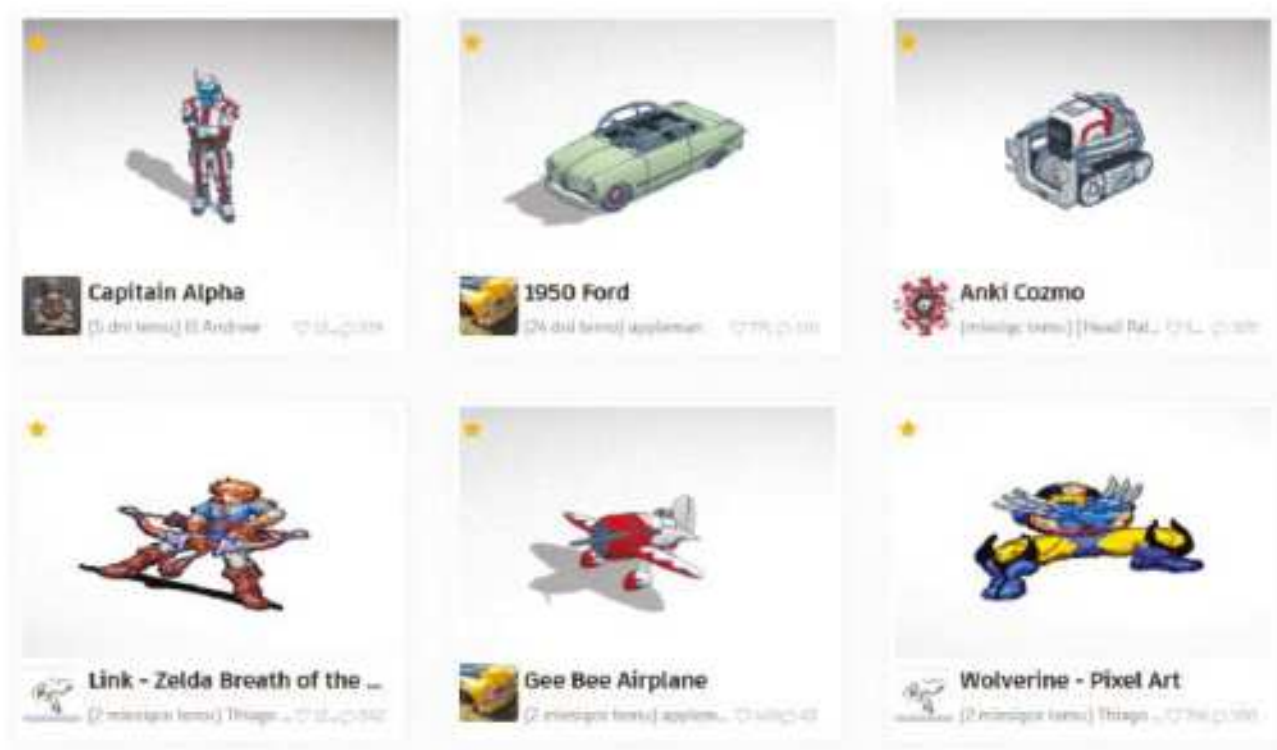
Wystarczy obrócić płaszczyznę roboczą, aby dostosować kształty lub zmienić widoki. Wprowadź dokładne wymiary za pomocą linijki.



Rys. 18.3. Podstawowe opcje pomocy w chmurze Tinkercad



Po wstępnym zapoznaniu się ze środowiskiem programu czas na ćwiczenie. Zacznij od przejrzania projektów w opcji **Galeria** (rys. 18.4.). Po naprowadzeniu kursora na pole projektu możesz obracać obiekt w płaszczyźnie poziomej.



Rys. 18.4. Przykłady projektów dostępnych w chmurze Tinkercad

Do ćwiczenia wybraliśmy projekt buldożera.



Rys. 18.5. Proces pobierania projektu z galerii chmury Tinkercad



edytor Paint 3D

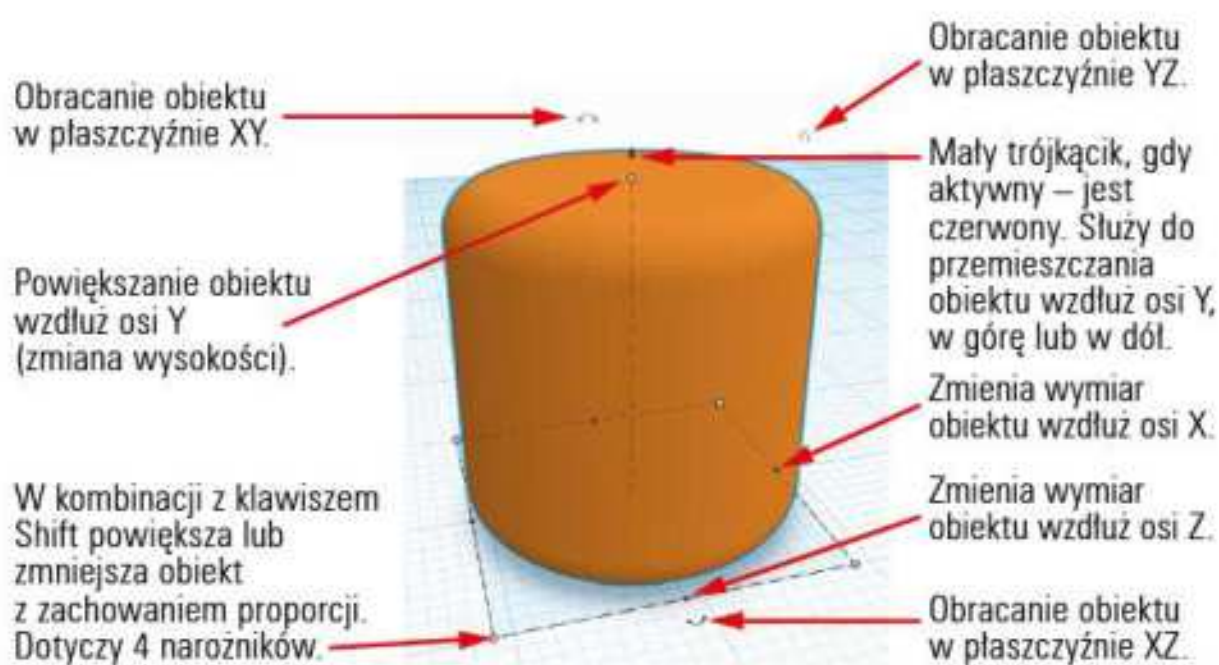
Pobrany projekt możesz obejrzeć w programie Paint 3D (rys. 18.6.). Będzie pozbawiony kolorów, ale dokładnie odwzoruje bryłę do wydruku.



Rys. 18.6. Projekt wyświetlony za pomocą Paint 3D

18.2. Komin i napis, czyli modyfikujemy projekt

Dorysujmy do projektu komin – rurę wydechową silnika i napis na przedniej części karoserii. W trakcie ćwiczenia wykorzystaj umiejętności zdobyte na lekcji na temat chmury oraz odpowiedzi z rysunków 18.7. i 18.8., na których przedstawiono funkcje punktów kontrolnych i przeznaczenie przycisków myszy.



Rys. 18.7. Funkcje punktów kontrolnych w edytorze grafiki 3D



Przybliża i oddala obiekt (powiększanie, pomniejszanie). Nie zmienia rozmiarów modelu.

Użyty z **Ctrl** – daje funkcję identyczną jak prawego przycisku. Umożliwia wybór narzędzi i wykonywanie ich operacji.



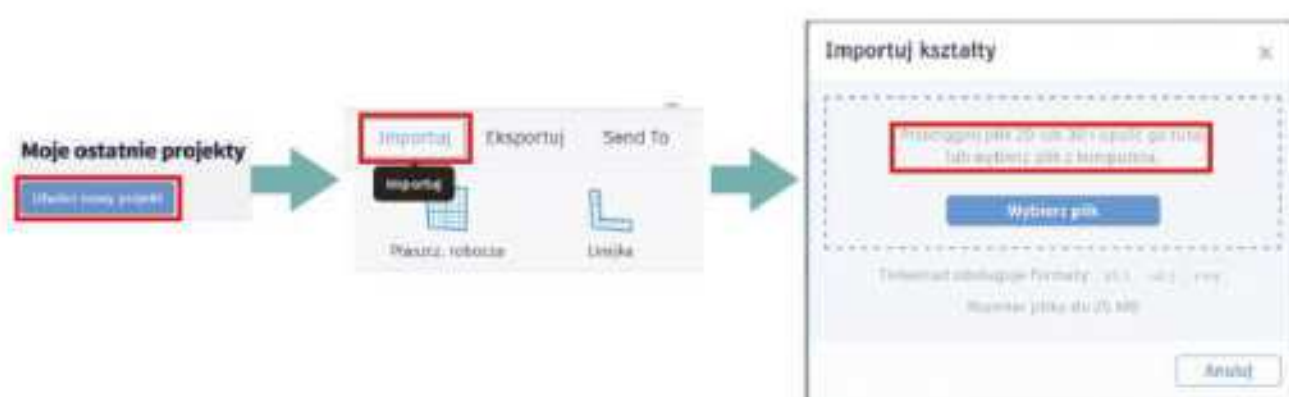
Służy do obracania modelu z zachowaniem jego centralnego położenia. Pozwala na obejrzenie modelu ze wszystkich stron.

Z **Shift** przesuwa płaszczyznę roboczą.

Rys. 18.8. Funkcje przycisków myszy w edycji 3D

Po zalogowaniu się do Tinkercad utwórz nowy projekt i wczytaj pobrany wcześniej plik STL (rys. 18.9.). Poćwicz sposoby podglądu projektu zgodnie z rysunkiem 18.8.

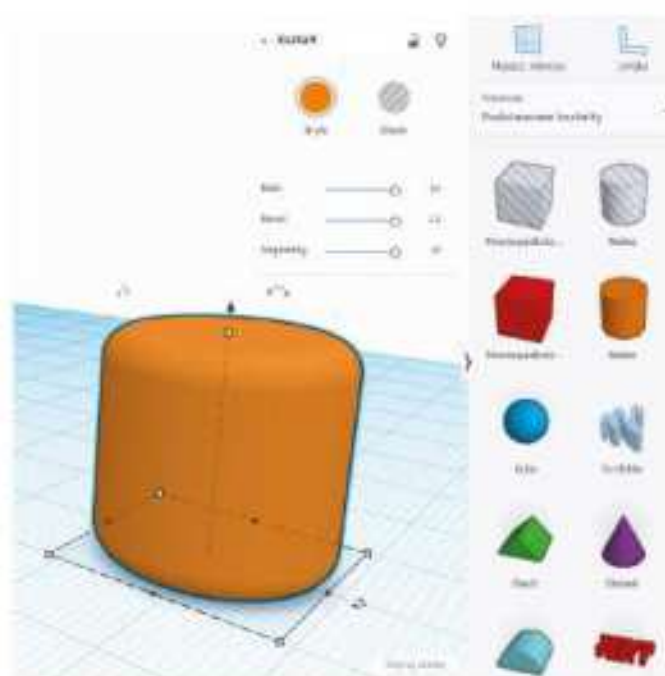
plik STL



Rys. 18.9. Proces wczytywania projektu z dysku do edytora w chmurze Tinkercad

Skoro importowany model jest już gotowy, wprowadź nasze modyfikacje.

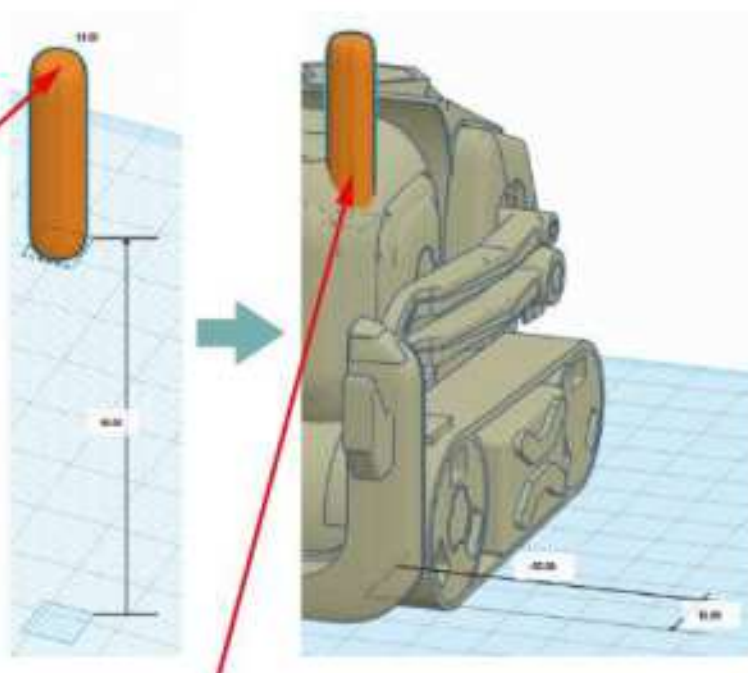
Zapewne wstawiony walec będzie miał niepasujące proporcje i wymiary. Wiesz już, jakie znaczenie mają punkty kontrolne edytora. Ustal więc wysokość walca, a następnie umieść go w odpowiednim miejscu (rys. 18.11.). W trakcie pracy poznasz różnice pomiędzy edycją na płaszczyźnie a pracą z użyciem edytora w trzech wymiarach. Szybko jednak przyzwyczaisz się do techniki operowania obszarem roboczym jako przestrzenią. Takie ćwiczenie przyda się także w poznawaniu zagadnień geometrii przestrzennej na lekcjach matematyki.



Rys. 18.10. Wstawianie elementu modelu 3D



Po uzyskaniu odpowiedniej wielkości elementu chwyć za trójkąt na osi pionowej i unieś element na odpowiednią wysokość (wzdłuż osi Y).



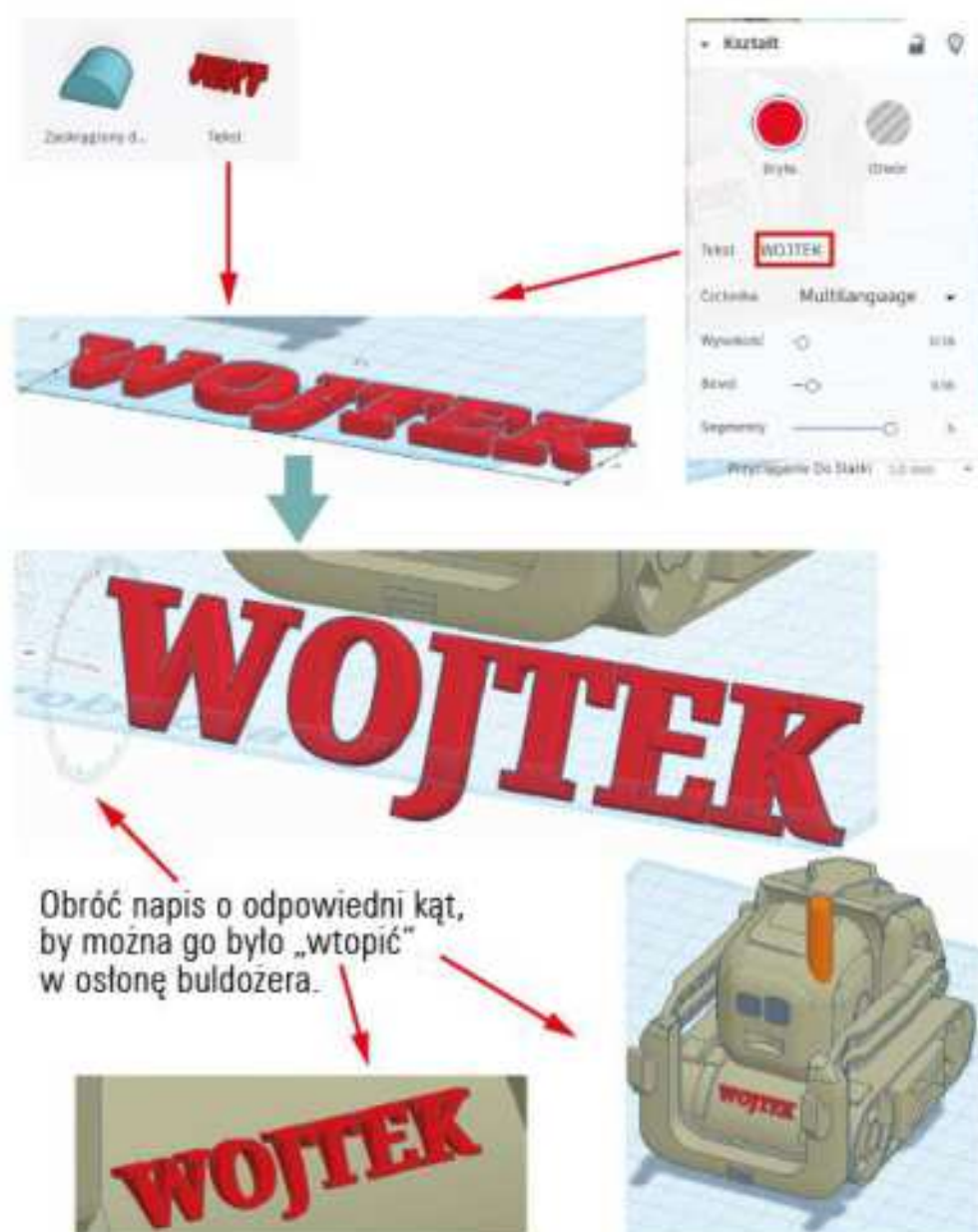
Chwyć element i przenieś w płaszczyźnie XZ w odpowiednie miejsce.

Rys. 18.11. Przemieszczanie obiektu wzdłuż osi XYZ

format STL

Czas na umieszczenie na modelu napisu. Z menu wybierz opcję **Napis** i postępuj według opisu. Zwróć uwagę na to, by grubość liter była wystarczająca do wtopienia ich w karoserię pojazdu. W przeciwnym przypadku będą odstawały od powierzchni buldożera i mogą wystąpić kłopoty z wydrukiem 3D. Użyjesz także narzędzia do obracania wokół osi X. W trakcie pracy sprawdzaj widok modelu z wielu stron, by napis był rzeczywiście dobrze wpasowany w bryłę pojazdu.

Po zakończonej pracy wyeksportuj plik do formatu STL i obejrzyj go w edytorze Paint 3D.



Obróć napis o odpowiedni kąt, by można go było „wtopić” w osłonę buldożera.

Rys. 18.12. Wstawianie napisu do projektu 3D



Rys. 18.13. Model gotowy do wydruku

Uwaga!

Tak jak w innych chmurach w Tinkercad pliki projektów zapisują się automatycznie.

18.3. Tabliczka, czyli tworzymy pierwszy projekt dla drukarki 3D

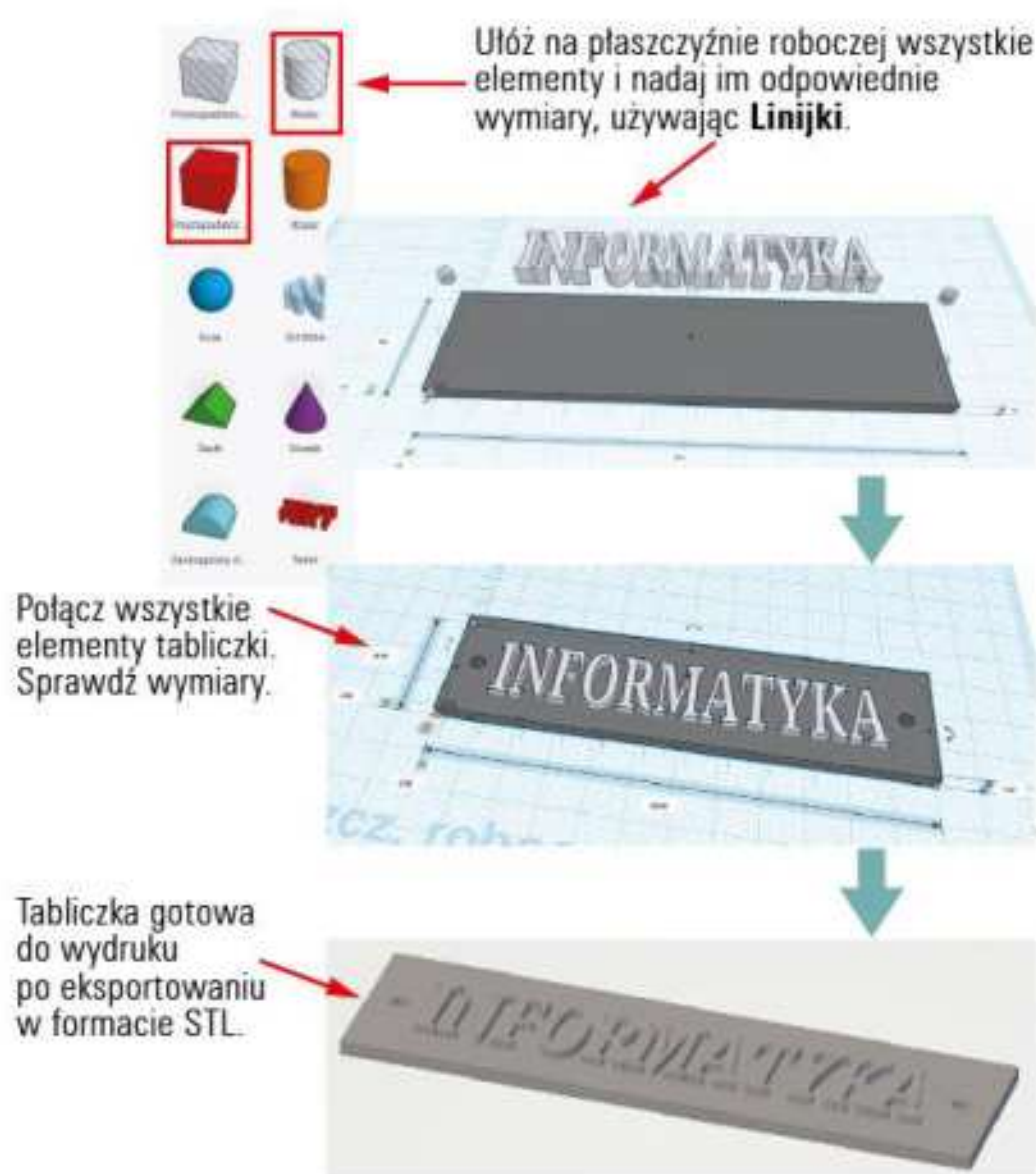
Zaprojektujemy i wykonamy tabliczkę, którą będzie można przymocować do drzwi pracowni informatycznej. Oczywiście pierwszym krokiem jest utworzenie nowego projektu. Postaraj się nadać mu oryginalną nazwę. Ustaw płaszczyznę roboczą w taki sposób, by wygodnie pracować z obiektami.

W tym projekcie **ważne są wymiary** i wzajemne proporcje. Napis musi wtopić się w tabliczkę, a otwory muszą przechodzić na wylot. Osiągniesz to, odpowiednio manipulując punktami kontrolnymi poznanymi wcześniej.

Aby dokładnie dopasować wymiary modelu, wybierz narzędzie **Linijka**. Przyłóż ją w narożniku elementu. W dolnej części okna roboczego możesz zmieniać wymiar jednej małej kratki płaszczyzny roboczej. Domyślnie ustawiono 0,1 mm, co oznacza, że duża kratka podziałki ma boki o długości 1 cm.

narzędzie Linijka

Wysokość walców tworzących otwory powinna być równa grubości tabliczki, a liter – odpowiednio większa, ponieważ muszą wystawać ponad płaszczyznę tabliczki.



Rys. 18.14. Proces tworzenia modelu tabliczki z napisem 3D

Jeśli zainteresowało cię tworzenie modeli trójwymiarowych dla drukarek, skorzystaj z tutoriali udostępnionych w chmurze (rys. 18.15.). Znajdziesz tam wiele ciekawych projektów z dokładnym opisem ich kreślenia. Jeśli język angielski w opisie sprawia ci problem, skopiuj odpowiedzi i użyj tłumacza, na przykład Google.



Rys. 18.15. Przykładowe projekty dostępne w Tinkercad w postaci opisów ich tworzenia



ZADANIA DO ROZWIĄZANIA

1. Korzystając z programu Tinkercad, wykonaj tabliczkę na drzwi z wklęsłym napisem. Zachowaj odpowiednie wymiary i proporcje projektu. Wyeksportuj pracę do formatu STL. Obejrzyj model za pomocą Paint 3D.
2. Znajdź ciekawy model na licencji CC wśród dostępnych w Tinkercad. Edytuj go w tym programie i postaraj się ustalić, z jakich elementów jest zbudowany.
3. Wykonaj projekt breloka do kluczy do sal lekcyjnych w postaci gruszki z wklęsłymi numerami i elementami ozdobnymi. Zadbaj o odpowiedni rozmiar. Zastanów się, jak sprawić, by wewnątrz gruszki było puste. Oszczędzi to bardzo dużo filamentu.
4. Wykonaj projekt zabawki – labiryntu. Załóż, że ścianki mają grubość 2 mm, wysokość 1 cm. Na końcu umieść otwór, do którego może wpaść metalowa kulka przemieszczająca się po labiryncie w momencie przechylenia zabawki. Wybierz kształt zewnętrzny labiryntu, na przykład prostokąt, kwadrat, koło.
5. Zaprojektuj podstawkę pod telefon komórkowy umożliwiającą ustawienie aparatu pod kątem 60 stopni. Może ona służyć do oglądania filmów lub czytania e-booków bez konieczności trzymania smartfona.

PODSUMOWANIE LEKCJI

- Jednym z edytorów 3D przeznaczonych dla początkujących grafików i amatorów druku 3D jest program **Tinkercad** pracujący w chmurze. Aby z niego skorzystać, można założyć własne konto lub zalogować się za pomocą jednego z portali społecznościowych lub Google. Tinkercad oferuje wiele samouczków: od podstawowych do opisów tworzenia skomplikowanych projektów. s. 129
- Gotowy projekt można wyeksportować do formatu **OBJ** lub **STL**. s. 133
- Pliki dla drukarki 3D w formacie STL można przeglądać w **Paint 3D**. s. 134
- Wymiary modeli można sprawdzać za pomocą **narzędzia Linijka**. Jest ona także przydatna przy zmianach wielkości elementów lub całych modeli. s. 135



19. Wizualizacja pomysłów, czyli projektujemy dom w edytorze 3D

NA TEJ LEKCJI:

- dowiesz się, czym jest i do czego służy SketchUp;
- stworzysz projekt 3D domu i poznasz podstawowe narzędzia edycyjne w SketchUp.

Projektowanie 3D to nie tylko tworzenie modeli do wydruku dla specjalistycznych obrabiarek przemysłowych i innych maszyn. Do tego typu projektów używa się programów CAD. To już wiesz. Zdajesz sobie jednak sprawę, że technologia pozwalająca na uzyskiwanie przestrzennych obrazów służy także wielu innym celom. Większość gier komputerowych powstaje dzięki zaawansowanym edytorom 3D. Projekty architektoniczne, wizualizacje budynków i inne również są wykonywane za pomocą takich programów. Obok wyspecjalizowanych pakietów dla inżynierów istnieją edytory mniej zaawansowane, pozbawione modułów ściśle związanych z technologiami budownictwa, ale dające możliwość pracy twórczej nad wizualizacją projektu. Może właśnie dziś wpadniesz na pomysł budowy swojego wymarzonego domu?

19.1. Nowe i stare, czyli społeczność SketchUp prezentuje projekty

SketchUp

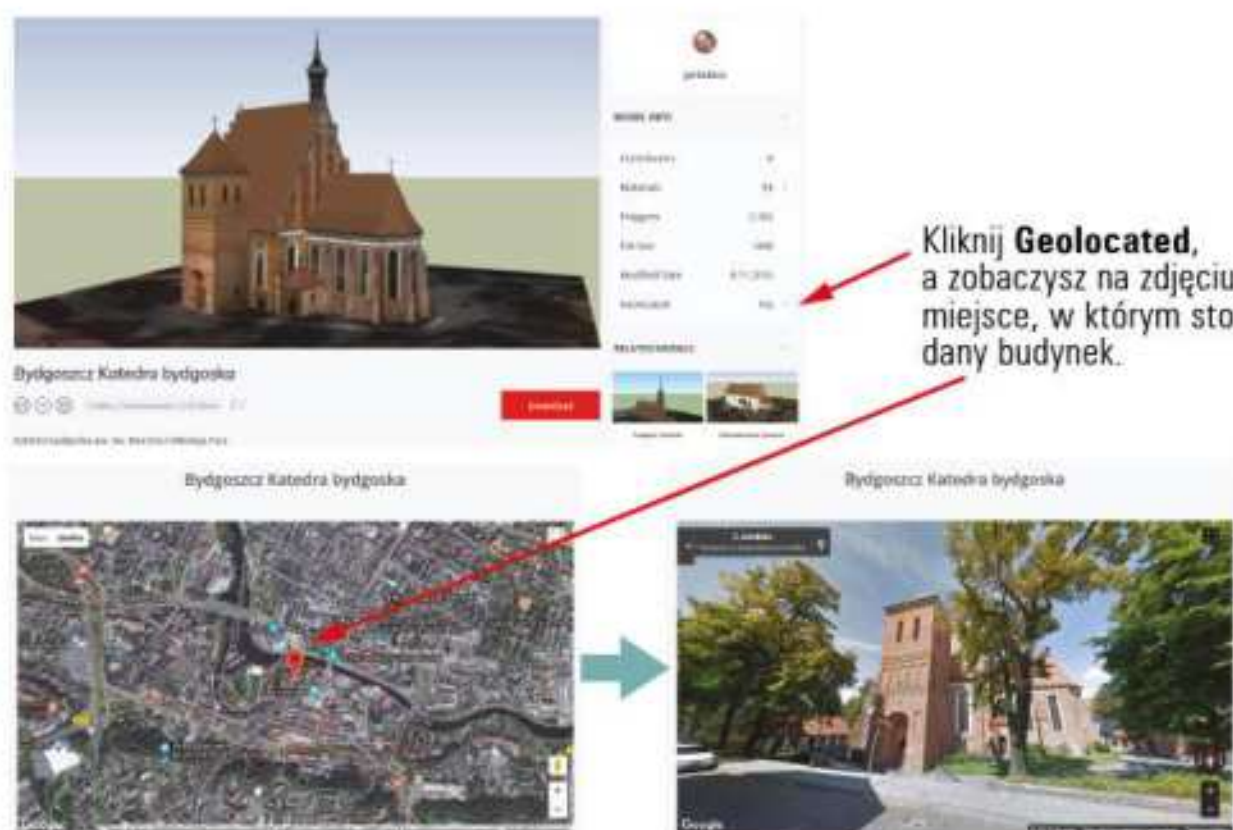
Wśród programów do edycji grafiki 3D znajdują się takie, które dzięki intuicyjnie zaprojektowanym narzędziom i darmowej licencji stały się bardzo popularne wśród miłośników architektury. Wpływ na to ma także udostępnienie darmowej wersji on-line i w postaci aplikacji dla każdego powszechnie używanego systemu operacyjnego. Jednym z najpopularniejszych jest **SketchUp**. Obecnie jego wersja komercyjna jest programem bardzo zaawansowanym i wykorzystywanym przez wielu projektantów. My posłużymy się darmową wersją on-line.

konto na SketchUp

Podobnie jak w przypadku Tinkercad możesz zalogować się do tej chmury kontem Google lub założyć własne konto (twoje projekty będą zapisywane w chmurze). Ponieważ SketchUp podczas rejestracji użytkownika wymaga podania numeru telefonu, imienia i nazwiska, dlatego bezpieczniej posłużyć się wcześniej założonym kontem gmail. Najlepiej ocenić edytor po dziełach w nim tworzonych. Łatwo je znajdziesz. Ciekawym doświadczeniem będzie szukanie wizualizacji różnych budowli, na przykład z twojej okolicy. Okazuje się bowiem, że użytkownicy odwzorowali w SketchUp całe ulice i dzielnice oraz większość polskich zabytków. Przy okazji możesz zdobyć wiele informacji na temat tych obiektów i zwiedzić okolicę za pomocą aplikacji Google Maps i Google Earth. Takie umiejętności przydadzą się na lekcjach historii i podczas przygotowywania tras turystycznych na wycieczkę klasową (rys. 19.1.).



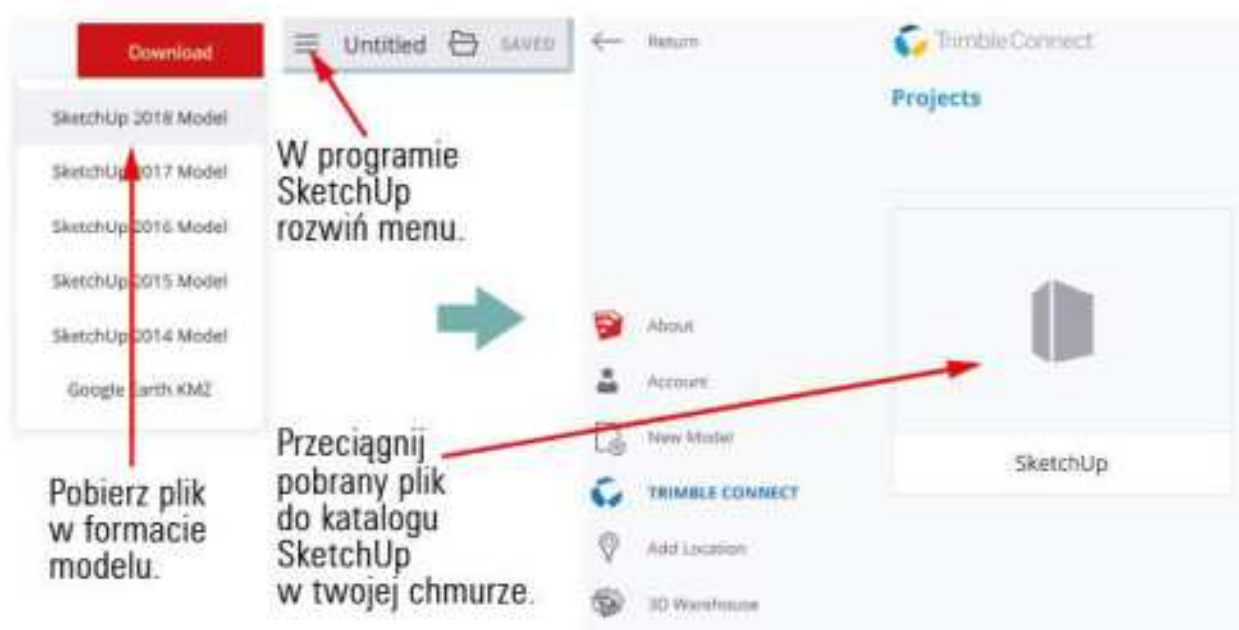
Przy okazji możesz sprawdzić, jak w tych usługach wygląda okolica, w której mieszkasz. Wpisz w wyszukiwarkę frazę: SketchUp Bydgoszcz Fara.



Rys. 19.1. Cyfrowy model budowli i powiązania z mapami Google

Gdy odbędziesz już wirtualną wycieczkę, możesz pobrać (rys. 19.2.) i obejrzeć dany model ze wszystkich stron. Zauważ, że przykładowy budynek jest pokryty w części teksturami wyciętymi z fotografii oryginału (rys. 19.3.).

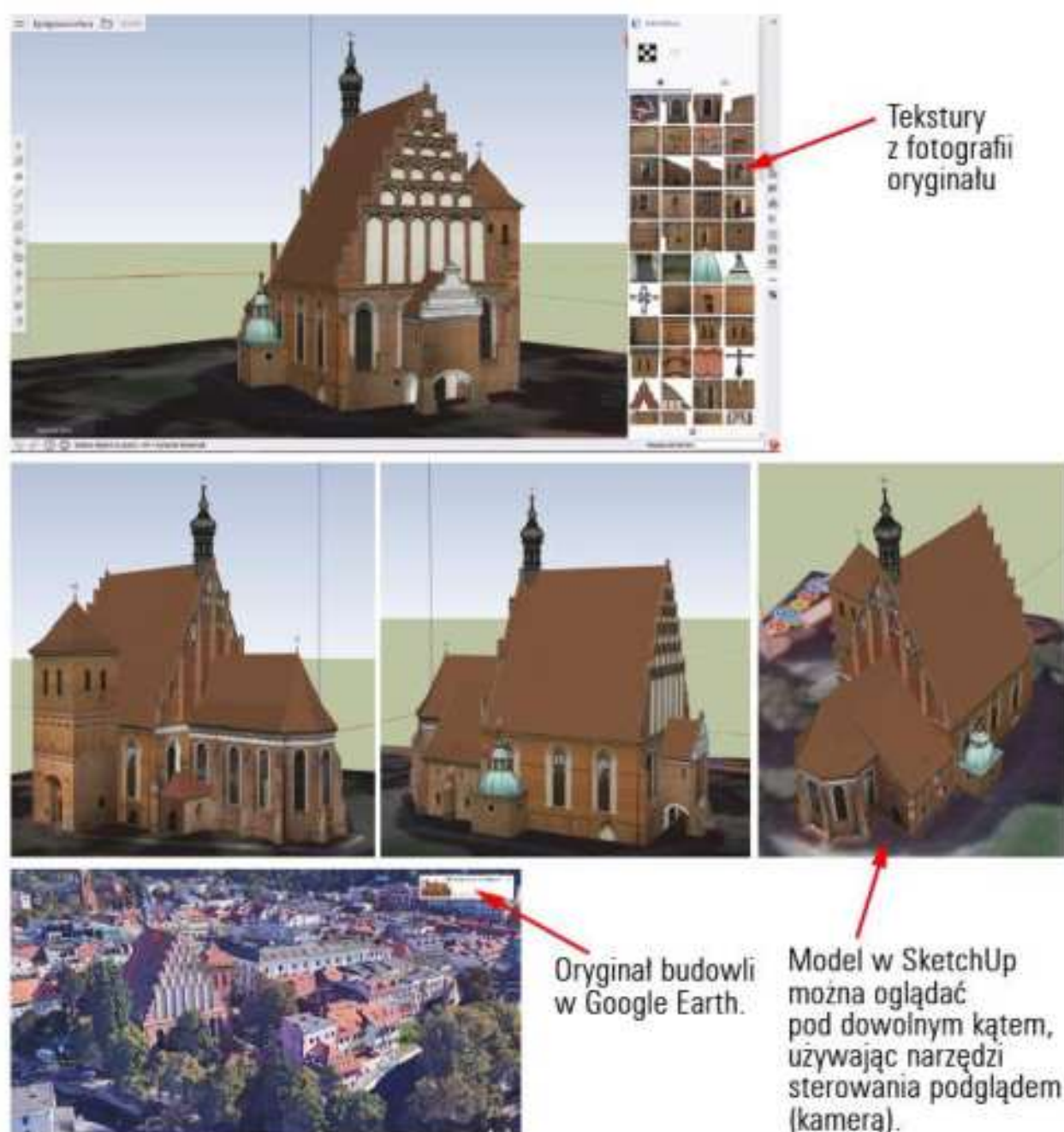
makiety budynków



Rys. 19.2. Proces kopiowania modelu do chmury programu SketchUp



Klikając dwukrotnie ikonę wczytanego do chmury projektu, uruchomisz jego edycję (rys. 19.3.).



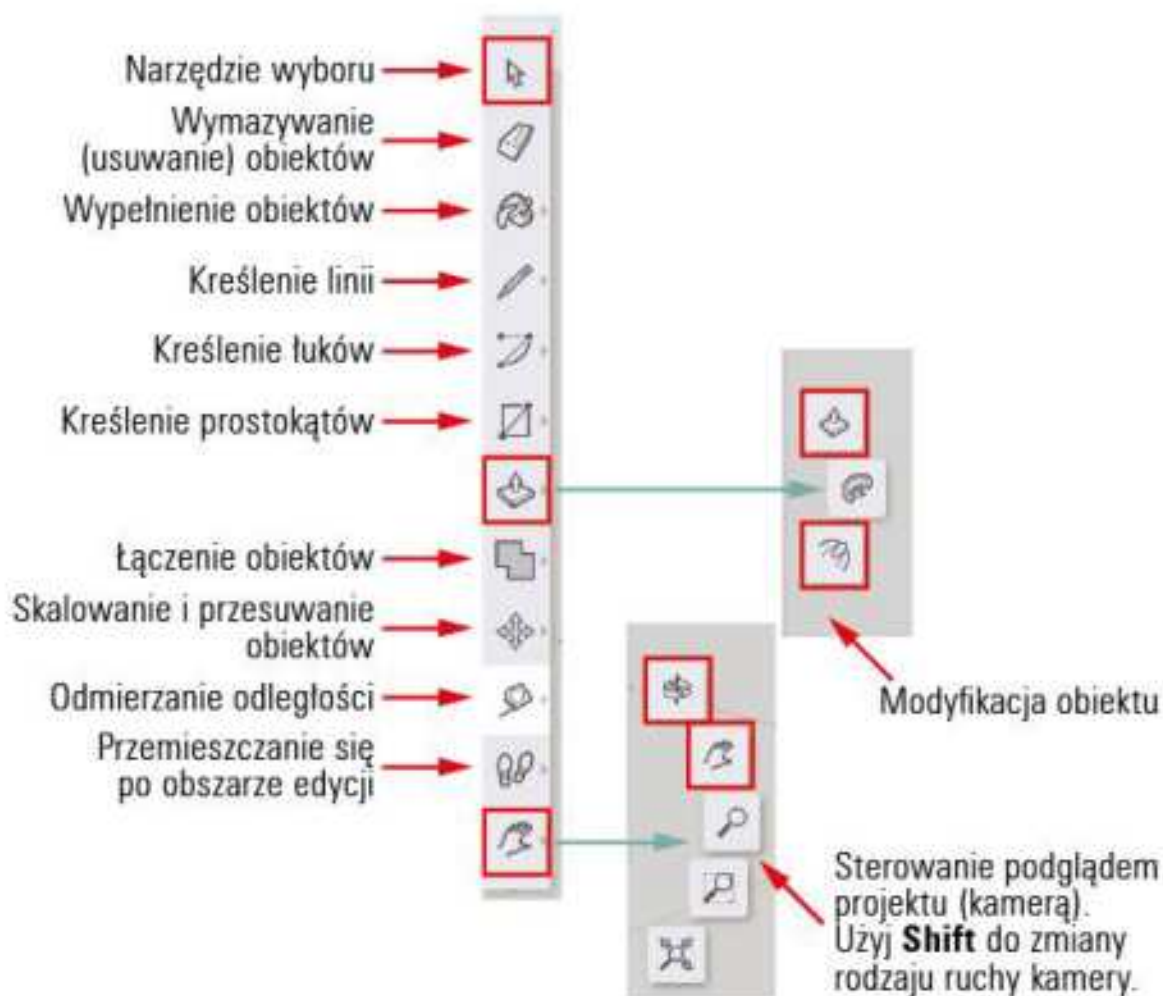
Rys. 19.3. Pobrany model w edycji w programie SketchUp

19.2. Mój wymarzony dom, czyli tworzymy model budynku w edytorze 3D

Czas na uruchomienie wyobraźni i zaprojektowanie nowego obiektu. Zanim zabierzesz się do pracy obejrzyj tutoriale, które dokładnie opisują funkcje narzędzi i metody tworzenia projektów budynków. Oryginalne lekcje znajdziesz pod adresem sketchup.com/learn/videos/. Obejrzenie wspomnianych filmów jest podstawą do sprawnego wykonania ćwiczenia i szybkiego opanowania metody pracy z edytorem SketchUp. Nie wszystko da się opisać, czasami lepiej zobaczyć to na filmie.

Po obejrzeniu lekcji przygotowanych przez autorów edytora uruchom program w chmurze i rozpocznij projektowanie od utworzenia nowego modelu (opcja **New Model**). Wybierz **Prosty szablon (Simple template)** w systemie **metrycznym**. Odległości będą liczone w metrach. Na pustym polu edycyjnym pojawi się postać człowieka. Kliknij ją jeden raz i usuń klawiszem **Del**. Teraz twój edytor jest gotowy do pracy. Przykładowy projekt stworzono, by pokazać użycie podstawowych narzędzi. Twój projekt może wyglądać zupełnie inaczej. Zanim przystąpisz do pracy, uruchom wyobraźnię.

Jeśli obejrzenie instrukcji producenta programu jest niewystarczające, skorzystaj z rysunku 19.4. Czerwoną ramką zaznaczono szczególnie ważne narzędzia.



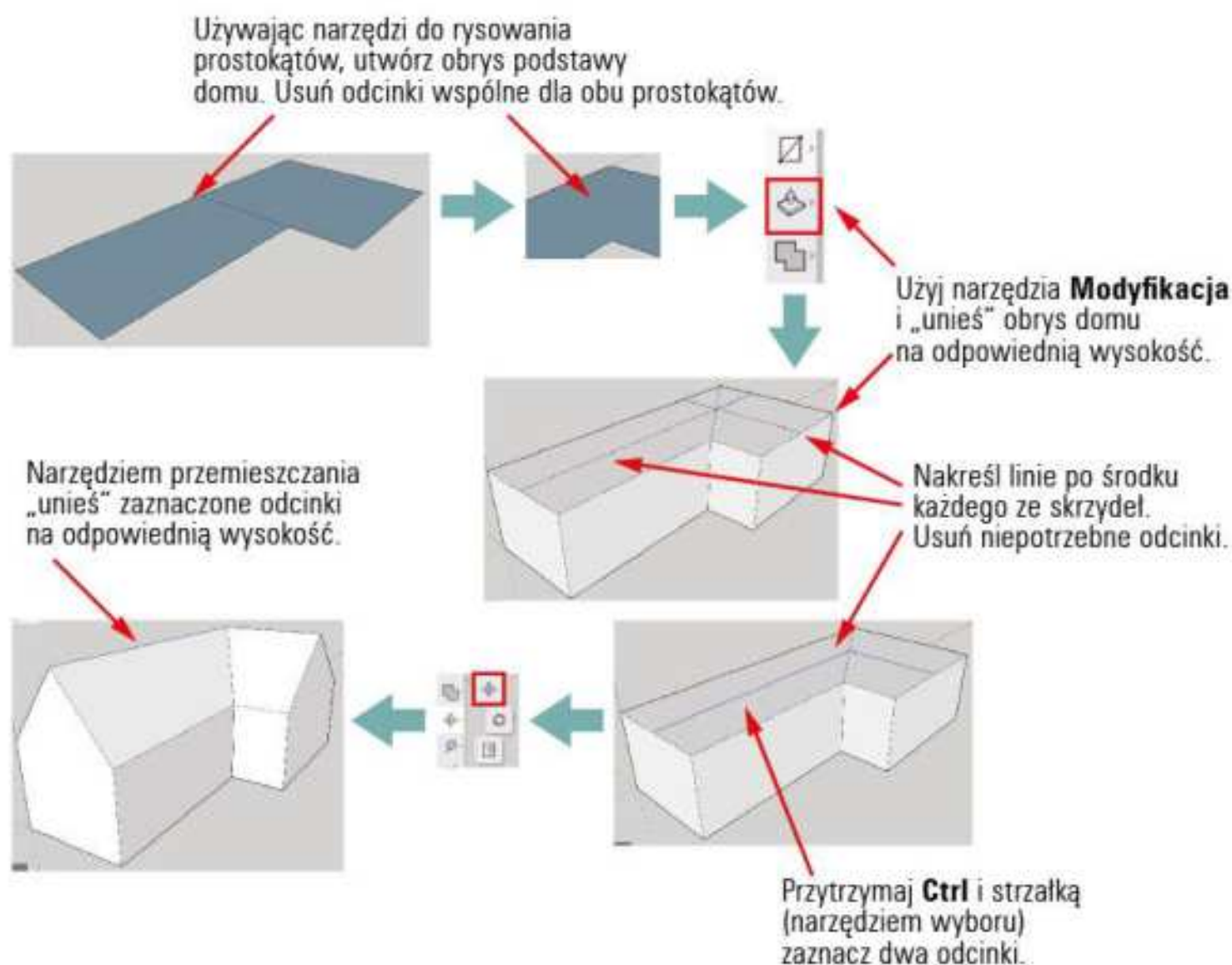
Rys. 19.4. Podstawowe narzędzia edycyjne w **SketchUp**

Pracę rozpoczniemy od narysowania bryły domu (rys. 19.5.). Aby sprawnie wykonać wszystkie czynności, przypomnij sobie metody pracy z narzędziami z obejrzanych filmów producenta.

1. Do usuwania linii i płaszczyzn użyj narzędzia wymazywania (ikona gumki).
2. Linia kończy się i zaczyna w punktach przecięć, a nie w miejscach, w których zaczęto i skończono kreślić. Jest przyciągana do węzłów.
3. Zaznaczanie obiektów, na przykład linii, uzyskasz dzięki narzędziu wyboru (strzałka).



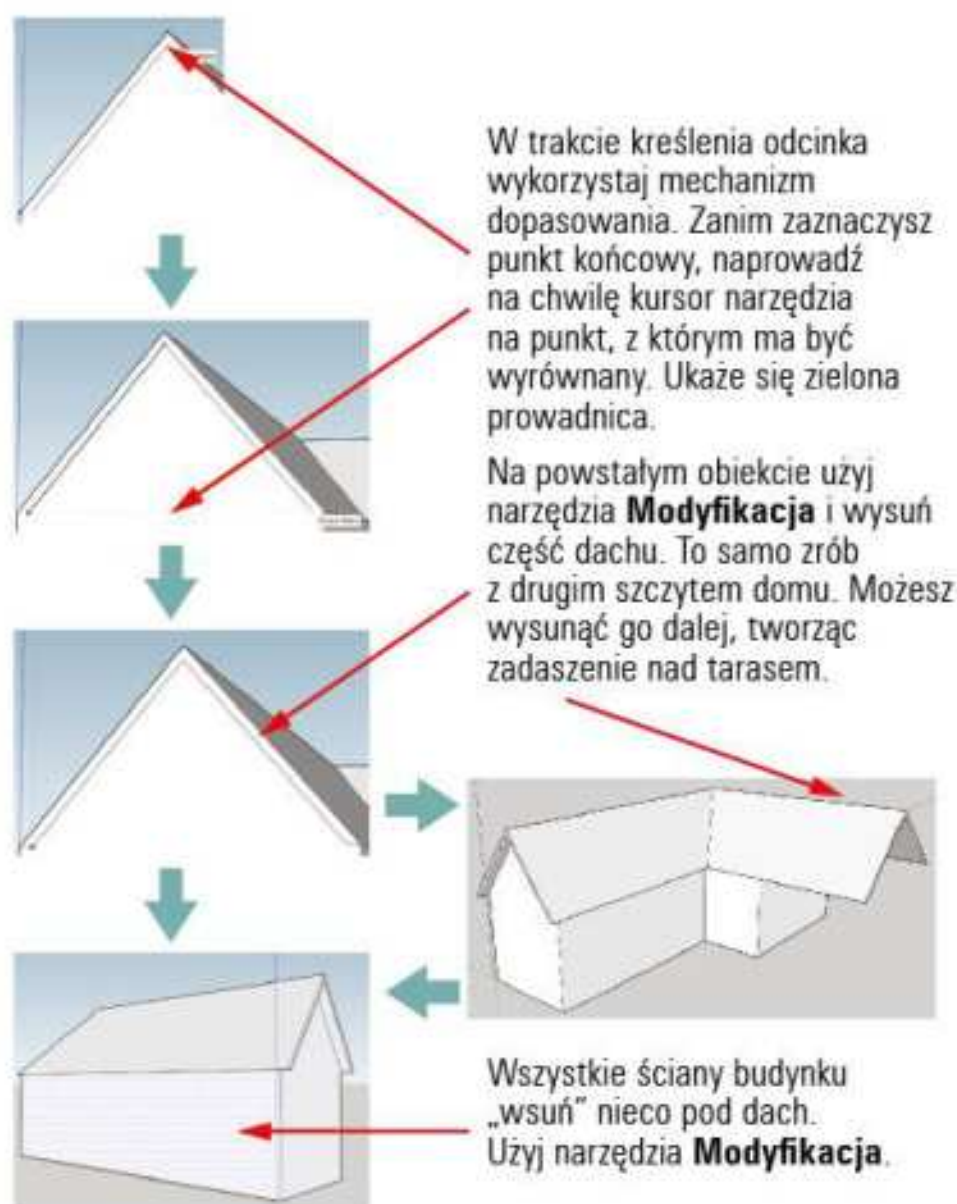
4. W trakcie kreślenia obiektów pokazują się punkty kontrolne pomagające umiejscowić element w odpowiednim miejscu przestrzeni. Ich kolory są zgodne z osią. Punkt pokaże się także pośrodku odcinka, którego ma dotyczyć kreślony obiekt.
5. Przesuwanie itp. dotyczy wskazanego obiektu albo zaznaczonych obiektów.



Rys. 19.5. Tworzenie bryły modelu domu

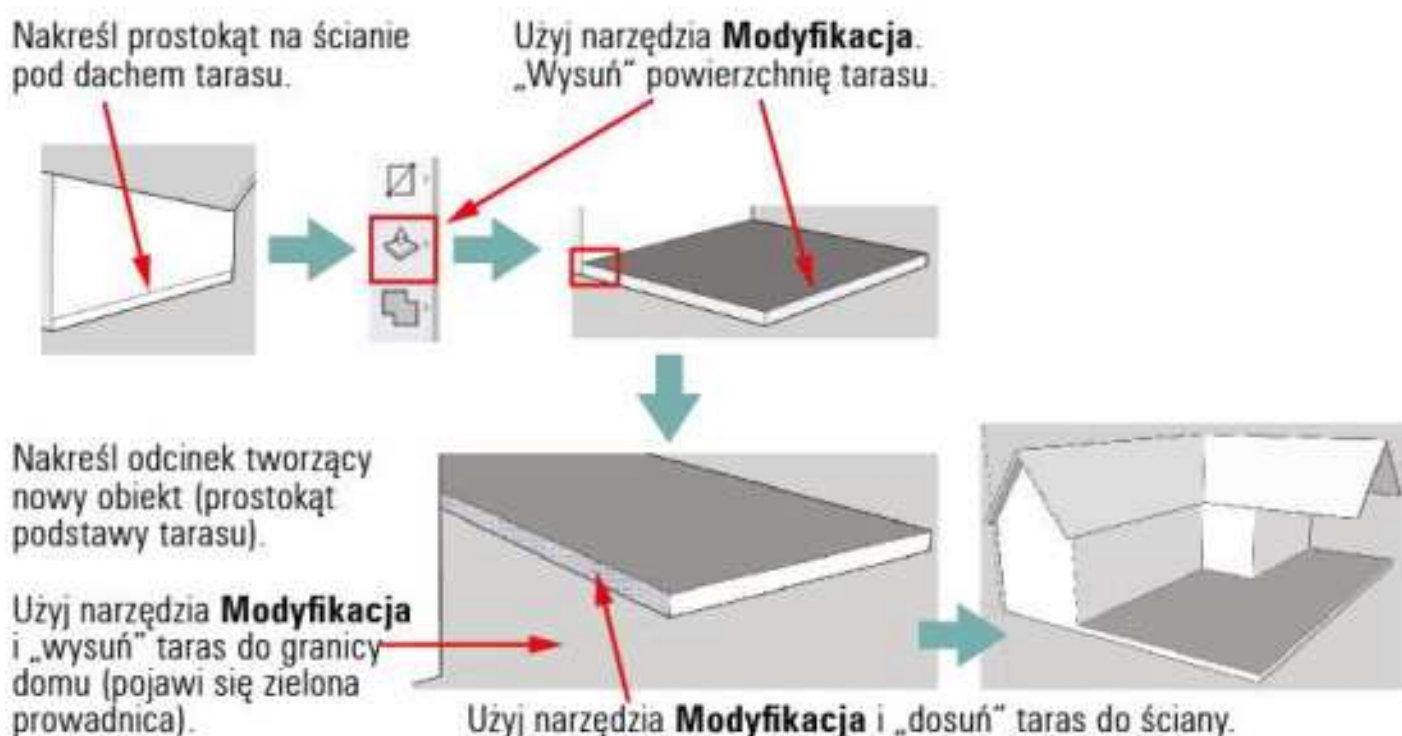
Twój dom nabiera kształtów. Bryłę zmodyfikuj tak, by była zbliżona do rzeczywistych domów. Wydłuż dach, dodaj okna i taras. Użyj do tego znanych już narzędzi – **Kreślenie linii**, **prostokąta** i **Modyfikacja**.

Pamiętaj, że w kreśleniu bardzo pomocne są punkty kontrolne i sygnalizacja kolorami położenia względem osi.



Rys. 19.6. Modyfikacja bryły projektu domu

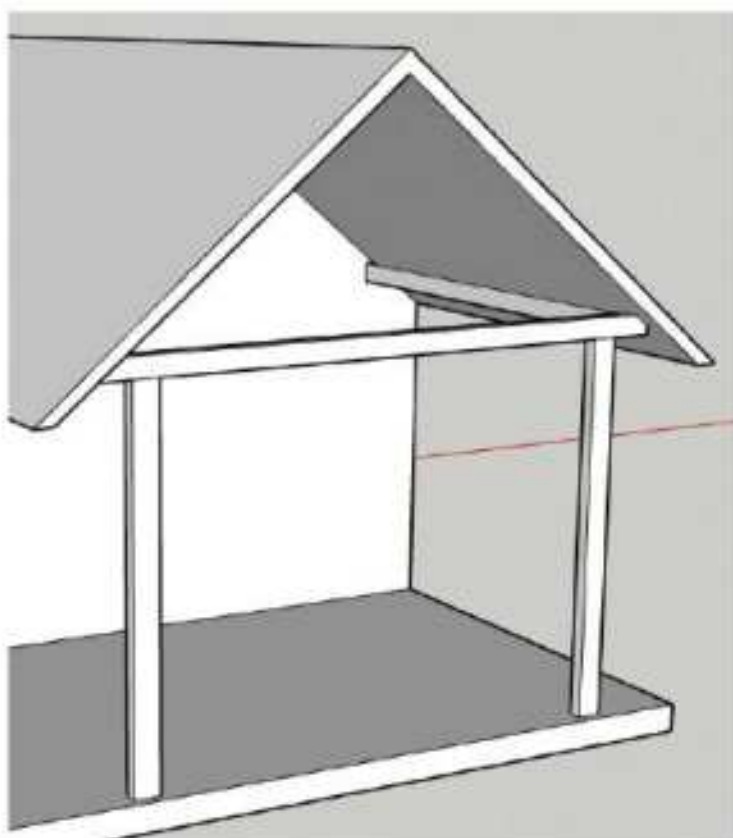
Domyślasz się, co będzie następnym krokiem w tworzeniu projektu domu? Wykonaj taras, podpory dachu tarasu i okna. Potrzebne do tego będą narzędzia używane do tworzenia i modyfikacji bryły. Pamiętaj o korzystaniu z mechanizmu wyrównywania (rys. 19.7.).



Rys. 19.7. Tworzenie tarasu za pomocą narzędzia Modyfikacja



Jeśli taras jest już gotowy, czas podeprzeć dach. Belki możesz kreślić z odcinków lub prostokątów z użyciem **Modyfikacji** (rys. 19.8.).



Rysunek 19.8. Belki podtrzymujące dach tarasu

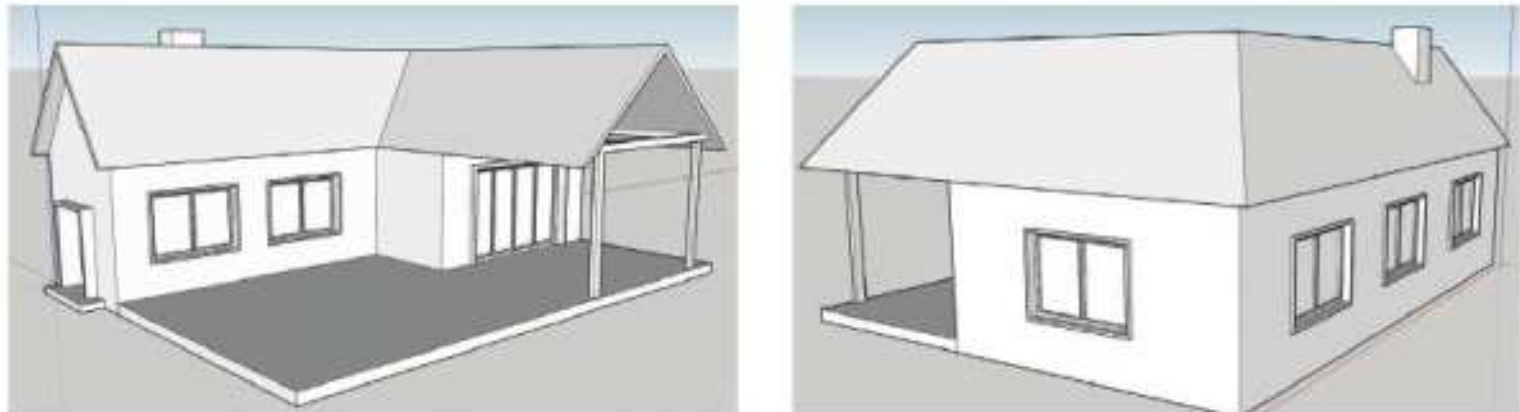
W twoim domu brakuje okien. Postępuj według rysunku 19.9.



Rys. 19.9. Proces tworzenia okien w projekcie domu

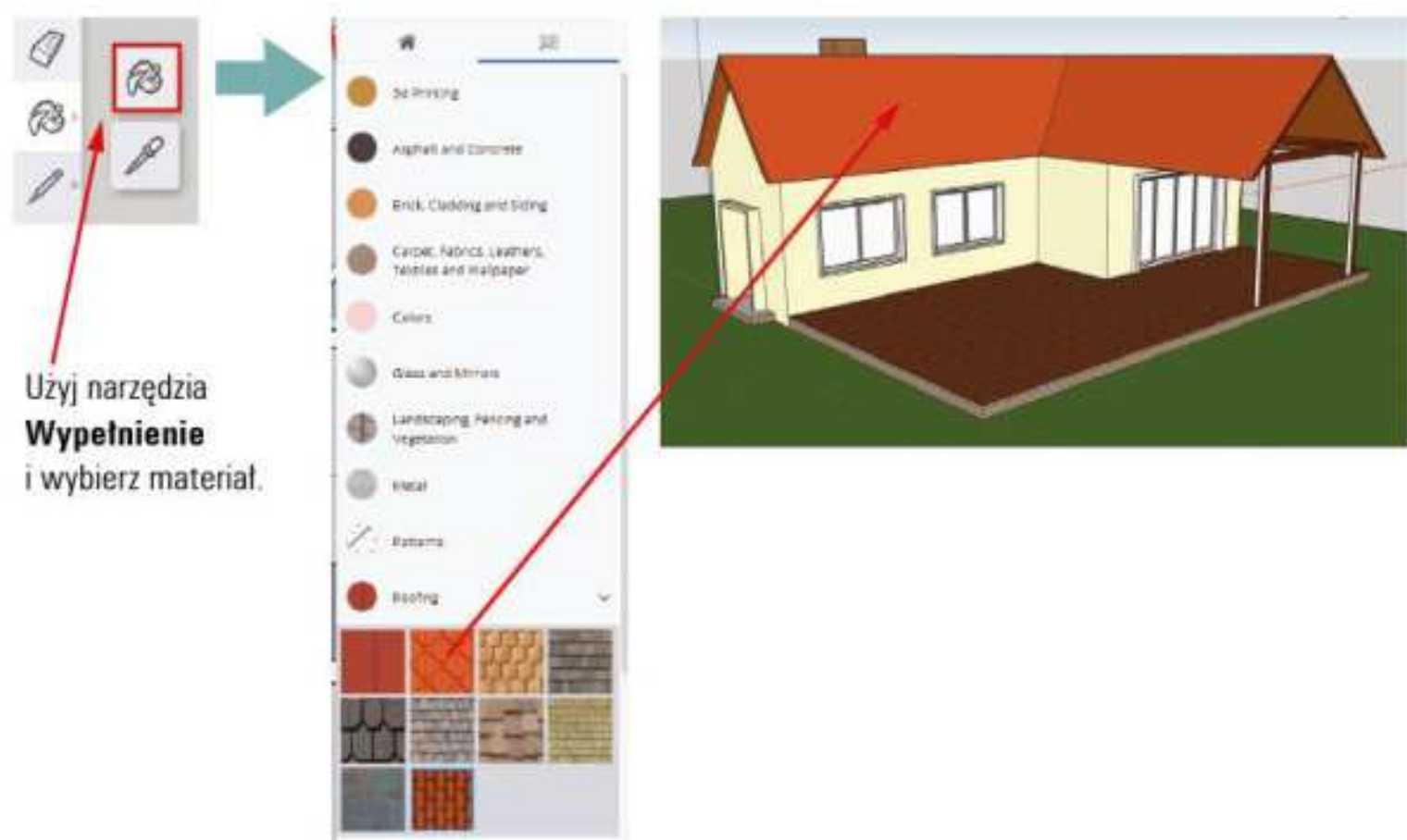


Jeśli wstawiłeś/wstawiłaś już wszystkie okna i drzwi, to projekt domu można uznać za gotowy (rys. 19.10.). Znasz już działanie narzędzi, więc możesz dorysować jeszcze rynny i dopracować szczegóły.



Rys. 19.10. Gotowy projekt bryły domu

Ostatnim aktem tworzenia modelu jest jego kolorowanie i pokrywanie teksturą (rys. 19.11.).



Rys. 19.11. Pokrywanie elementów projektu kolorem i teksturami



ZADANIA DO ROZWIĄZANIA

1. Na stronie *3dwarehouse.sketchup.com* znajdź modele wykonane w SketchUp, które przedstawiają budowle z twojej okolicy. Porównaj je z oryginałami. Wskaż, co można poprawić, aby były bardziej realistyczne lub przedstawiały więcej szczegółów. Sprawdź, jak tę budowlę i twoją okolicę przedstawia się w usługach Google. Wykorzystaj do tego opcję Geolocated.
2. Do modelu stworzonego w trakcie ćwiczenia dodaj detale, np. rynny, teksturę drewna na belkach podtrzymujących daszek, teksturę na drzwiach wejściowych. Oceń, na ile to zmieniło wygląd pracy i czy warto było podjąć ten trud.
3. Do modelu z ćwiczenia dodaj okna dachowe. Przemyśl ich kształt – zdecyduj, czy mają być wpuszczone w dach, zamocowane na nim czy może w postaci lukarny (znajdź znaczenie tego słowa).
4. Zaplanuj wycieczkę klasową. Wykonaj prezentację zawierającą modele budowli wykonane w SketchUp wraz z mapami i widokiem z Google Street View miejsc, które chcesz odwiedzić.
- 5*. Za pomocą SketchUp wykonaj model 3D budynku twojej szkoły. Postaraj się uwzględnić jak najwięcej szczegółów. Zorganizuj wystawę prac w postaci wydruków.

PODSUMOWANIE LEKCJI

- s. 138 • Obok zaawansowanych edytorów grafiki 3D, służących do tworzenia wysokiej jakości i rozdzielczości obrazów, np. do gier, istnieją także narzędzia do tworzenia projektów przestrzennych. Jednym z nich jest **SketchUp**. Oprócz wersji darmowej istnieją także bardziej zaawansowane wersje komercyjne.
- s. 138 • W SketchUp on-line możesz **założyć własne konto** z miejscem na swoje projekty na dysku serwera lub logować się za pomocą Google.
- s. 139 • Pasjonaci architektury używają SketchUp do tworzenia **makiety rzeczywistych budynków**, np. zabytków. Można je znaleźć na stronie *3dwarehouse.sketchup.com* i przeglądać lub edytować. Dzięki powiązaniu z mapami i usługami Google można dowiedzieć się, w jakim miejscu zostały zbudowane i obejrzeć okolicę.
- s. 141 • **Edytory 3D** w tym SketchUp są wyposażone w mechanizmy ułatwiające kreślenie i dopasowujące położenie w przestrzeni obiektów do innych lub do osi, np. podczas przesuwania lub kreślenia. Na **płaszczyzny budynków** można nakładać tekstury i kolory. W edytorze 3D działają wszystkie podstawowe mechanizmy znane z innych programów, np. **zaznaczanie, kopiowanie i wklejanie**.



20. Wypukłości nie tylko 3D, czyli tworzymy elementy graficzne publikacji

NA TEJ LEKCJI:

- nauczysz się tworzyć wypukłe napisy w edytorze GIMP;
- dowiesz się, jak przygotować ciekawe nagłówki stron internetowych itp.

Edytory grafiki rastrowej służą nie tylko do edycji fotografii, lecz także do tworzenia ciekawych rysunków. Część ozdobnych napisów spotykanych na stronach internetowych i w innych publikacjach wygląda atrakcyjniej po dodaniu wypukłości. Oczywiście można skorzystać z gotowych rozwiązań, ale więcej satysfakcji da przygotowanie ich samodzielnie. Ponadto umiejętność tworzenia takich napisów przyda się także do przekształcania innych obiektów, na przykład przycisków, strzałek i innych elementów publikacji, takich jak strony internetowe i prezentacje.

20.1. 3D w GIMP, czyli tworzymy ozdobny napis

Do utworzenia wypukłego napisu użyjemy darmowego edytora grafiki rastrowej GIMP. To narzędzie możesz zainstalować w Windows, MAC OS i Linux. W niektórych dystrybucjach Linux jest ono dołączone i nie wymaga instalacji, na przykład w Ubuntu. W przykładzie zaprezentowano napis 1 LO (rys. 20.1.), ale w swoich ćwiczeniach możesz utworzyć dowolny obiekt, przykładowo logo swojej szkoły. Efekt końcowy powinien przypominać litery znane z wyszukiwarki Google. Ćwiczenie z rysunku 20.1. można zastosować do wszystkich obiektów i wykorzystać na przykład do kolorowania fotografii.

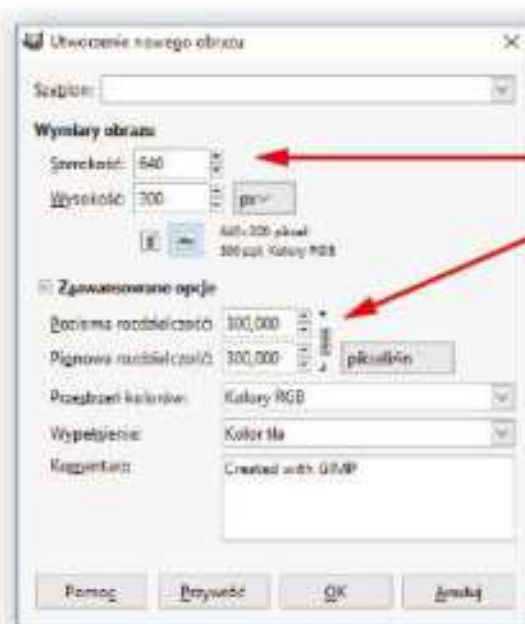
Zanim przystąpisz do realizacji projektu, przypomnij sobie niektóre zasady posługiwania się edytorami grafiki wyposażonymi w system warstw (rys. 20.1.). Jeśli pamiętasz je i biegle posługujesz się GIMP-em, możesz od razu przystąpić do tworzenia napisu. Użyj po prostu opcji zmiany koloru liter.

Warto zapamiętać sposób opisany na rysunku 20.1. Dzięki niemu zmienisz barwę dowolnego obiektu, na przykład ustawiając pewien poziom przezroczystości, pokolorujesz czarno-białe zdjęcia. Użyto przy tym warstw, bez których w każdym z edytorów graficznych niewiele można zrobić. Warstwy przypominają przezroczyste folie, na których rysuje się elementy całej kompozycji. Dzięki temu można je względem siebie przesuwąć, skalować, kolorować i nakładać w różnej kolejności. Poznane w trakcie ćwiczenia zasady operowania warstwami można zastosować także w innych edytorach, na przykład w profesjonalnym PhotoShop. Teraz czas przystąpić do pracy i z płaskich liter utworzyć wypukłe.

warstwy
w edytorze
graficznym



ustawienia
rozdzielczości
i wielkości grafiki



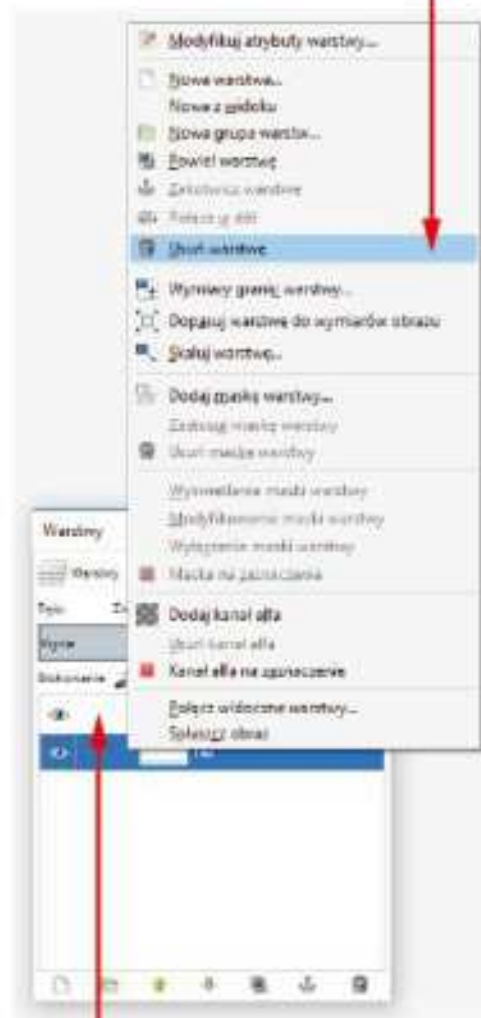
W procesie tworzenia nowego obrazu ustaw rozdzielczość na poziomie 300 pikseli. Wymiary obrazu ustal wg uznania. 640x300 sprawdzają się z napisem 1 LO.



Z **Przybornika** wybierz narzędzie **Tekst**, a następnie dobierz krój liter i wielkość. Dla TimesNewRoman w tym projekcie sprawdza się 300.

W polu wpisz tekst.

Kliknij prawym przyciskiem myszy na nazwie warstwy **Tło** i ją usuń.



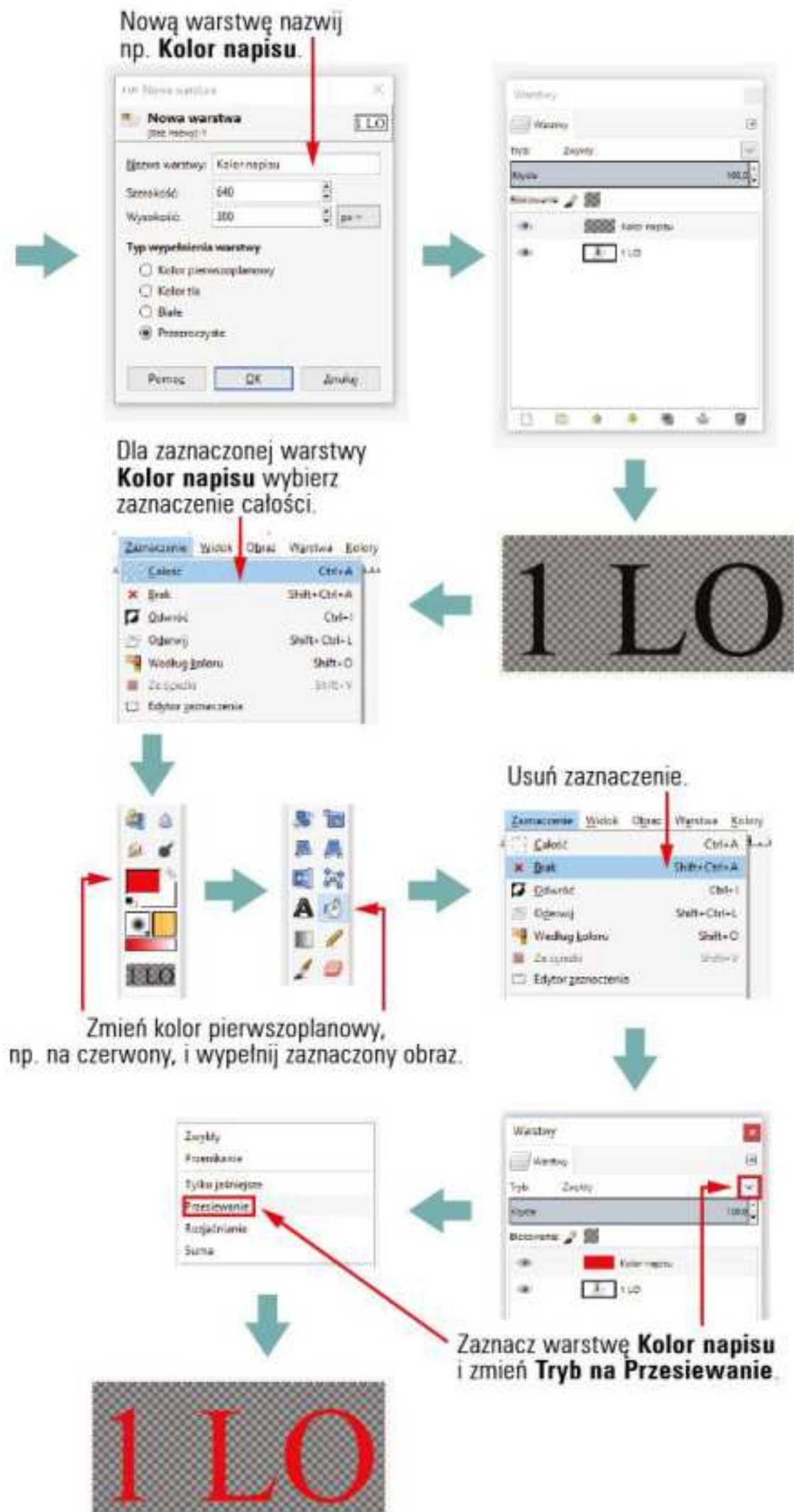
Widok projektu.
Napis na przezroczystym tle.



Utwórz nową warstwę.

Zaznacz typ wypełnienia jako przezroczyste.

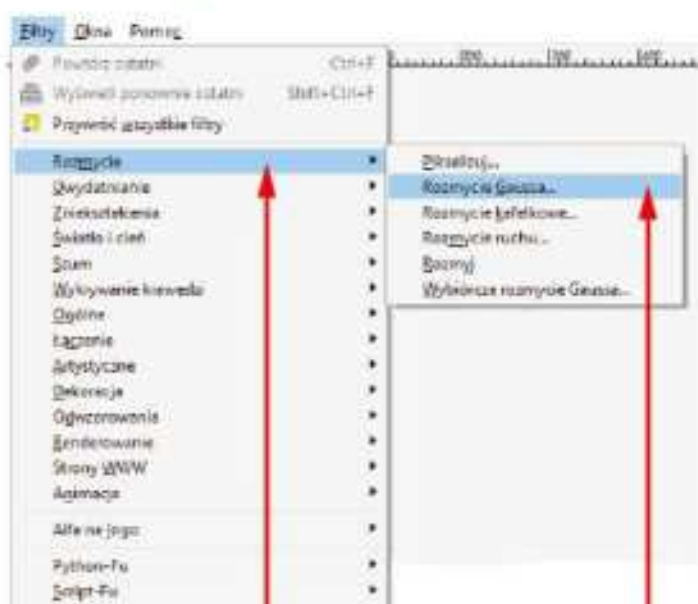
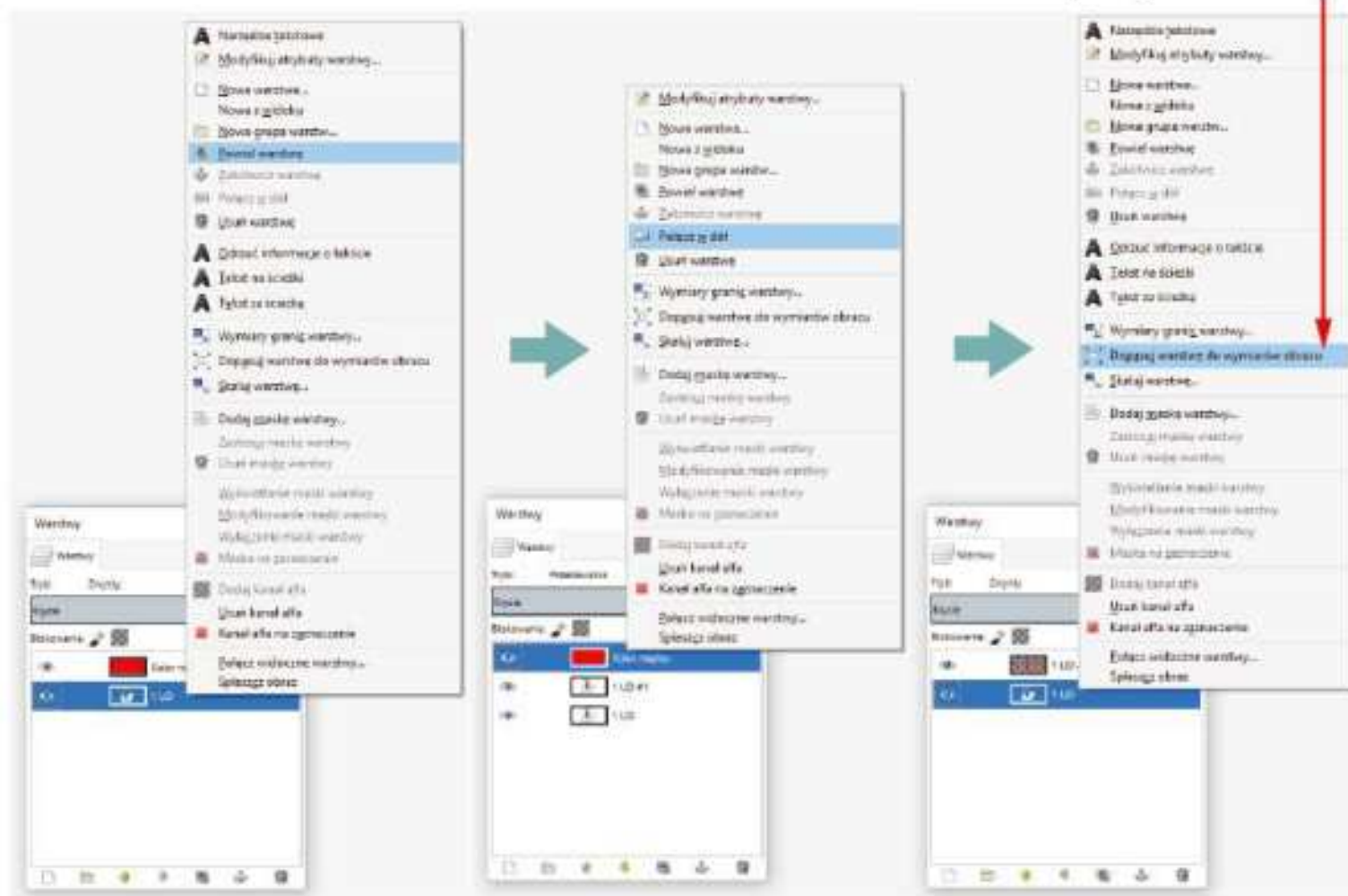
Rys. 20.1A. Proces zmiany koloru obiektu graficznego w GIMP



Rys. 20.1B. Proces zmiany koloru obiektu graficznego w GIMP

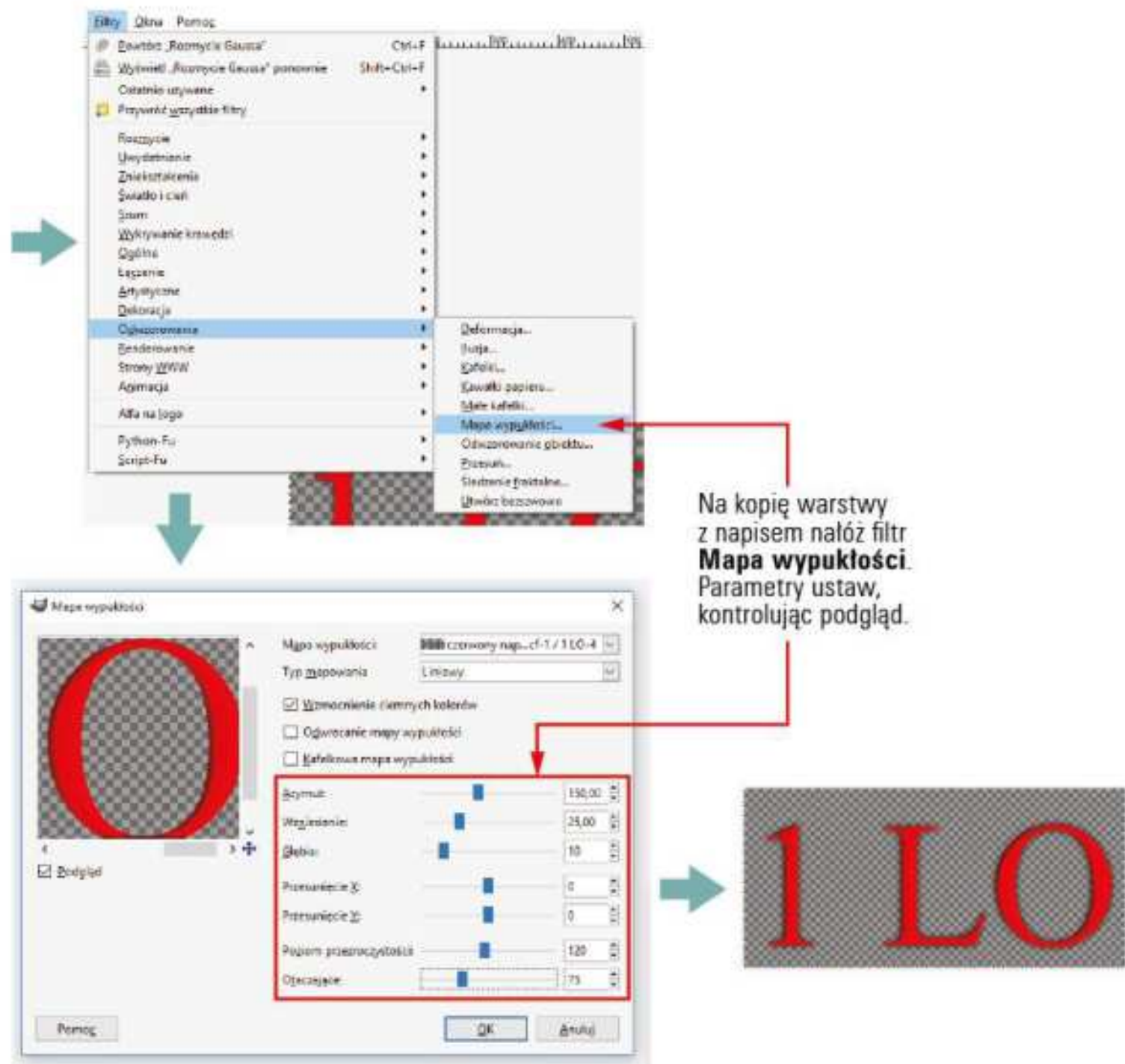


Wykonaj dla obu warstw.



Na warstwę z napisem (1LO) nałóż filtr
Rozmycie Gausa.

Rys. 20.2A. Proces tworzenia wypukłego napisu programem GIMP



Rys. 20.2B. Proces tworzenia wypukłego napisu programem GIMP

Ponieważ nasz napis w wyniku włączenia kanału alfa znajduje się na przezroczystym tle, należy go wyeksportować do pliku PNG lub innego typu zachowującego przezroczystość. Dlaczego? Zapisanie go w postaci JPG spowoduje, że napis znajdzie się na białym tle, co nie jest wskazane, jeśli docelowo ma on być umieszczony na danym tle (rys. 20.3.).



Rys. 20.3. Efekt wklejenia na tło plików JPG i PNG

20.2. Różne warstwy, czyli tworzymy nagłówek

tworzenie innych elementów

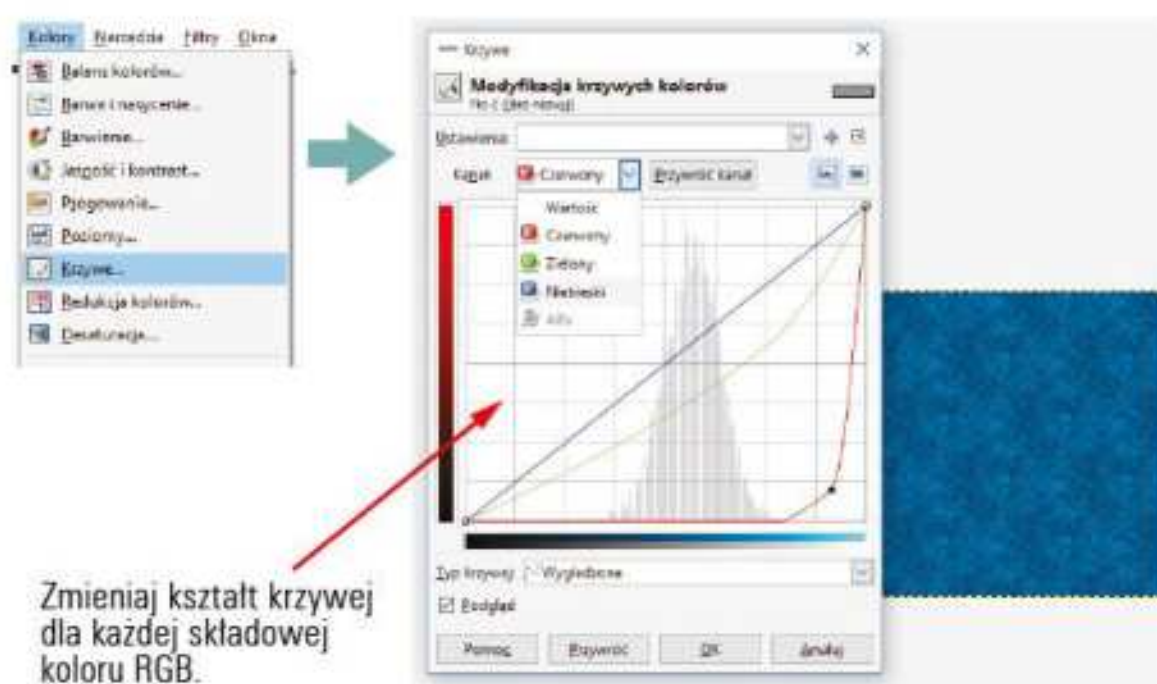
Użycie podobnych narzędzi i warstw możesz wykorzystać do utworzenia ciekawego nagłówka strony internetowej, czasopisma itp.

W opisie (rys. 20.4.) użyto jednej z metod zmiany kolorów rysunku polegającej na użyciu krzywych kolorów. W trakcie wykonywania ćwiczenia możesz sprawdzić, jak ich ułożenie i kształt wpływa na kolor. Nagłówek z ćwiczenia jest niebieski. Możesz uzyskać ten kolor, ustawiając poziomo na poziomie zerowym krzywe kolorów czerwonego i zielonego. Znajomość tej metody przyda ci się do korekcji barw innych rysunków i fotografii. Użyjesz także filtru Rzucania cienia. Jego parametry można regulować i obserwować, jaki mają wpływ na wygląd obrazu.

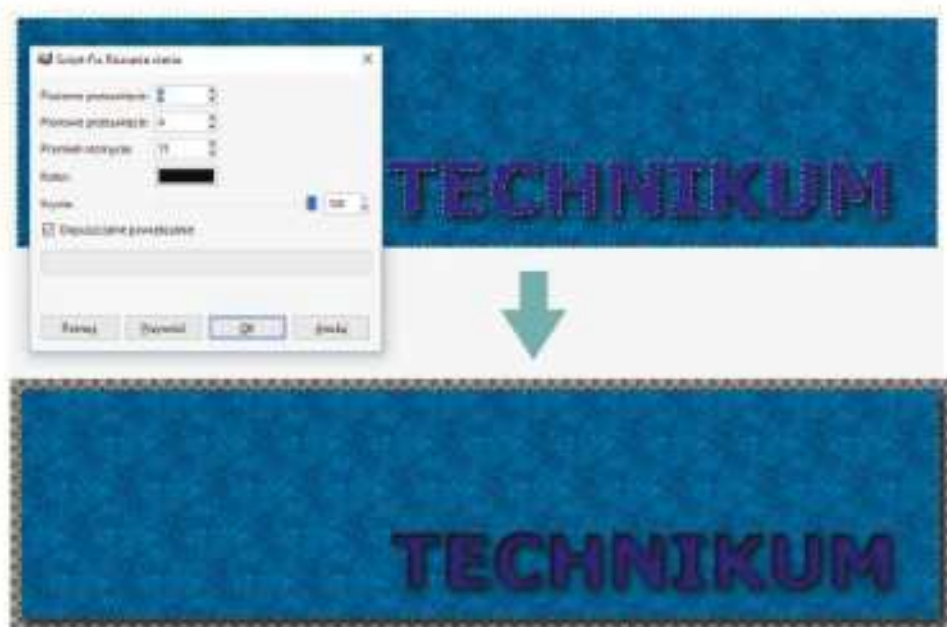
użycie filtrów



Rys. 20.4. Algorytm tworzenia nagłówka z napisem



Rys. 20.5. Zmiana barw za pomocą modyfikacji krzywych kolorów



Rys. 20.6. Nakładanie filtra „Rzucanie cienia” na zaznaczone litery



Rys. 20.7. Efekt włączania i wyłączania warstw w jednym projekcie

Pierwszy z przykładów z rysunku 20.7. może zostać użyty jako znak wodny (rys. 20.8.). Należy go wyeksportować do pliku z zachowaniem kanału alfa.



Rys. 20.8. Znak wodny wykonany w programie GIMP

ZADANIA DO ROZWIĄZANIA

1. Przeanalizuj jeszcze raz ćwiczenie z rozdziału i wytłumacz, dlaczego wyłączanie odpowiednich warstw powoduje dane zmiany.
2. a) Wzorując się na ćwiczeniu z rozdziału, wykonaj napis „Tęcza”. Zastosuj w warstwie Kolor napisu wszystkie barwy tęczy ułożone według kolejności. W ten sposób twój wypukły napis będzie wielokolorowy.
b) Wykonaj ćwiczenie 2.a, stosując płynne przejścia pomiędzy kolorami tęczy. Możesz zastosować zaznaczanie ze zmiękczeniem i zaokrągłaniem krawędzi.
Sprawdź, jak będzie wyglądał twój napis, gdy ćwiczenie wykonasz z pominięciem warstwy Kolor napisu, a barwę liter uzyskasz bezpośrednio poprzez wybór koloru napisu. Opisz różnice i przyczyny ich powstania.
3. Do napisu z ćwiczenia opisanego w rozdziale dodaj cień. Użyj odpowiedniego filtra. Gotowy napis wraz z cieniem wyeksportuj do odpowiedniego formatu zachowującego przezroczystość.



- 4*. Do napisu z ćwiczenia opisanego w rozdziale dodaj cień. Użyj odpowiedniego filtra. Przekształć warstwę z cieniem tak, by całość sprawiała wrażenie stojącego, oświetlonego od przodu napisu. Gotowy rysunek wraz z cieniem wyeksportuj do odpowiedniego formatu zachowującego przezroczystość.
5. Wykorzystaj doświadczenia z ćwiczenia z wklęsłym tekstem i wykonaj znak wodny do trwałego oznaczania twoich zdjęć. Może on zawierać twoje nazwisko lub inicjały.

PODSUMOWANIE LEKCJI

- W trakcie wykonywania ćwiczeń przypomniałeś/przypomniałaś sobie, jak wykorzystać **warstwy w edytorze graficznym** do tworzenia ciekawych efektów graficznych. Umiejętność operowania na warstwach pozwoli ci sprawnie używać innych edytorów grafiki i fotografii. s. 147
- Dzięki możliwości samodzielnego ustawienia **rozdzielczości i wielkości grafiki** możesz także tworzyć projekty drukowane na dużych formatach, np. ściankach reklamowych, rollupach. s. 149
- Zdobyte i utrwalone umiejętności możesz wykorzystać **do tworzenia elementów prezentacji, stron internetowych i dokumentów tekstowych**. s. 152
- **Użycie filtrów** wymagało zmiany ich parametrów w oknach dialogowych. Mogłeś/mogłaś zbadać, jak wpływają one na efekt graficzny. Warto eksperymentować z filrami i warstwami programów graficznych. Można uzyskać zaskakujące efekty. Ten sam projekt bez niektórych warstw może wyglądać zupełnie inaczej. s. 152

21. Z wydruku do komputera, czyli skanujemy i odczytujemy dokumenty

NA TEJ LEKCJI:

- dowiesz się, do czego służy technologia OCR;
- poznasz działanie skanera.

Komputerowa analiza obrazu jest wykorzystywana w wielu urządzeniach. Można nakładać w czasie rzeczywistym na obraz z kamery elementy graficzne, wskazania GPS itp. Ponadto dzięki analizie obrazu tekst widniejący na fotografii da się zmienić na plik edytowalny w edytorze. W jakim celu? Czy to łatwe do wykonania? Sprawdź.

21.1. Skąd wziąć kopie, czyli skanujemy dokumenty

Analiza pisma opiera się na przetwarzaniu obrazu (rys. 21.1.). Może on pochodzić z pliku lub kamery (także z aparatu fotograficznego). Analizowanie pisma odręcznego jest skomplikowane i wymaga zazwyczaj próbek oryginalnego charakteru pisma danej osoby.

OCR

Szybciej i łatwiej analizować zdjęcie druku. W tym celu opracowano technologię OCR (ang. Optical Character Recognition, czytaj: optikal karakter rekognyszyn). Dla uproszczenia programy, które się nią posługują, będziemy nazywać tym skrótem. Do czego może się przydać program OCR?



Rys. 21.1. Kolejność czynności podczas pozyskiwania wersji elektronicznej dokumentu

Przykład 1.

Student pisze pracę, na przykład opis projektu na zaliczenie przedmiotu. Na dzień przed terminem oddania uszkodził dysk komputera. Nie robił kopii zapasowych



ani nie pracował w chmurze. Miał jednak wydrukowaną jedną z wersji pracy. Użycie OCR i skanera pozwoliło mu na odzyskanie jej w postaci elektronicznej.

Przykład 2.

Osoba przygotowująca kalkulację kosztów projektu wydrukowała tabele z programu MS Excel i dołączyła do dokumentacji. Zgodnie z procedurą musi przesłać kopie w postaci elektronicznej, jednak nie ma dostępu do serwera. Skanowanie tabel i zapisanie plików wynikowych działania OCR pozwoli zyskać wersję elektroniczną kalkulacji (oczywiście formuły tabel nie da się w ten sposób skopiować).

Przykład 3.

Dziennikarz szkolnej gazetki internetowej pisze artykuł na pewien temat. Musi posłużyć się cytataми z książek i czasopism. Przepisywanie zabiera czas i może być źródłem błędów. Użycie OCR wyeliminuje te niedogodności.

Przykład 4.

Uczeń ma dokument w formacie PDF. Jednak nie może otworzyć go w licencjonowanym programie do jego edycji. OCR zamieni potrzebne fragmenty dokumentu na pliki tekstowe.

Podobnych przykładów można podać więcej i nigdy nie wiadomo, do czego jeszcze technologia OCR może się przydać.

Co musi wiedzieć użytkownik programu OCR? W opisie można znaleźć listę formatów plików, które dany program przetwarza, na przykład JPG, BMP, TIFF, GIF. Pliki graficzne zawierające tekst trzeba pozyskać poprzez skanowanie, robienie zrzutów ekranu, zdjęć itp.



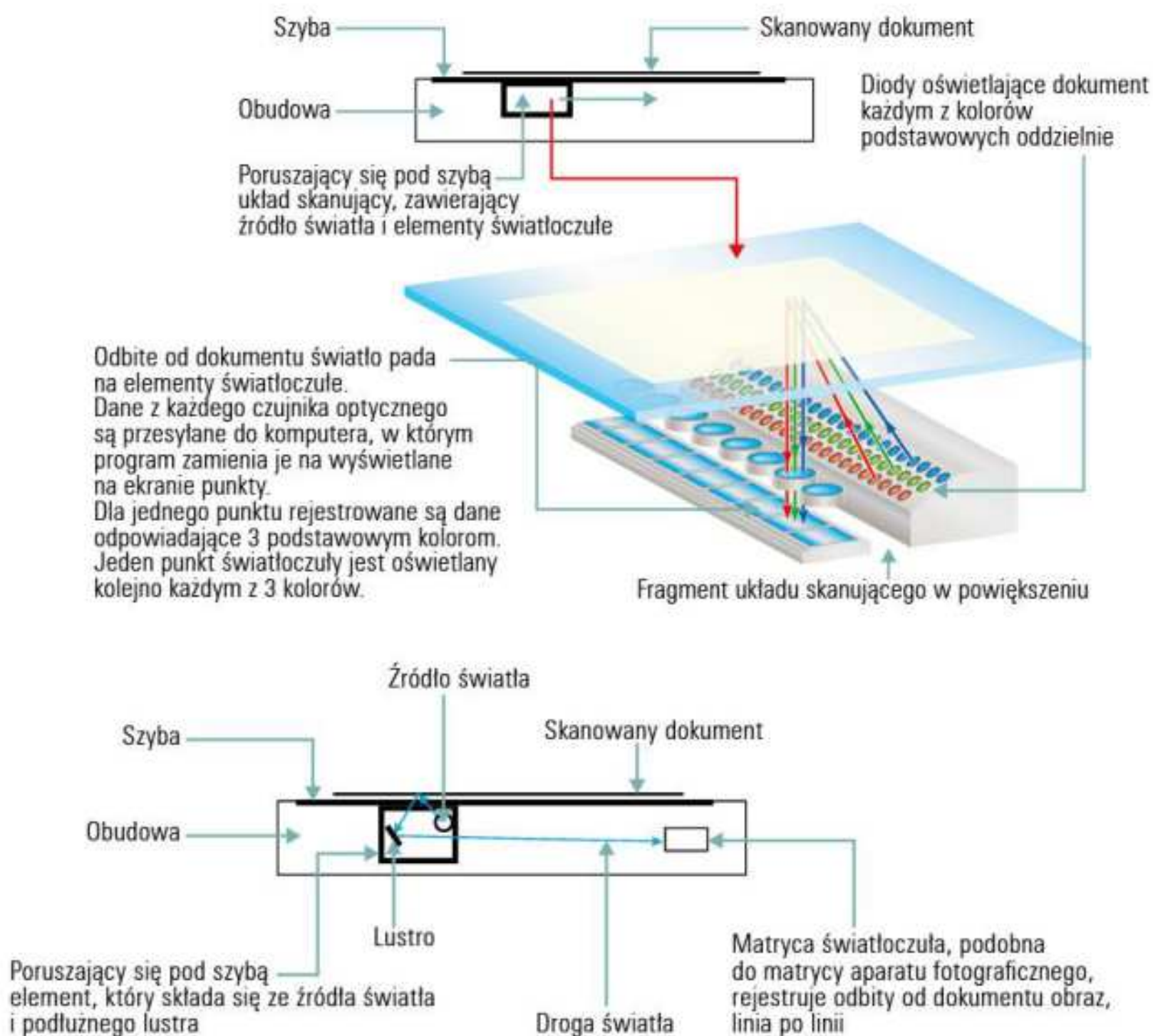
Rys. 21.2. Źródła plików dla programów OCR

działanie skanera

CCD i CIS

Czy wiesz, jak działa skaner? Powszechnie są stosowane dwie konstrukcje.

CIS, w której elementy światłoczułe i źródła światła (RGB) przesuwają się tuż przy skanowanej powierzchni, oraz CCD, w której obraz skanowanego dokumentu (oświetlany białym światłem) dociera poprzez układ luster i soczewek do matrycy podobnej w działaniu do matrycy aparatu fotograficznego (rys. 21.3.).



Rys. 21.3. Matryca CIS (u góry) i CCD (na dole) skanerów stołowych

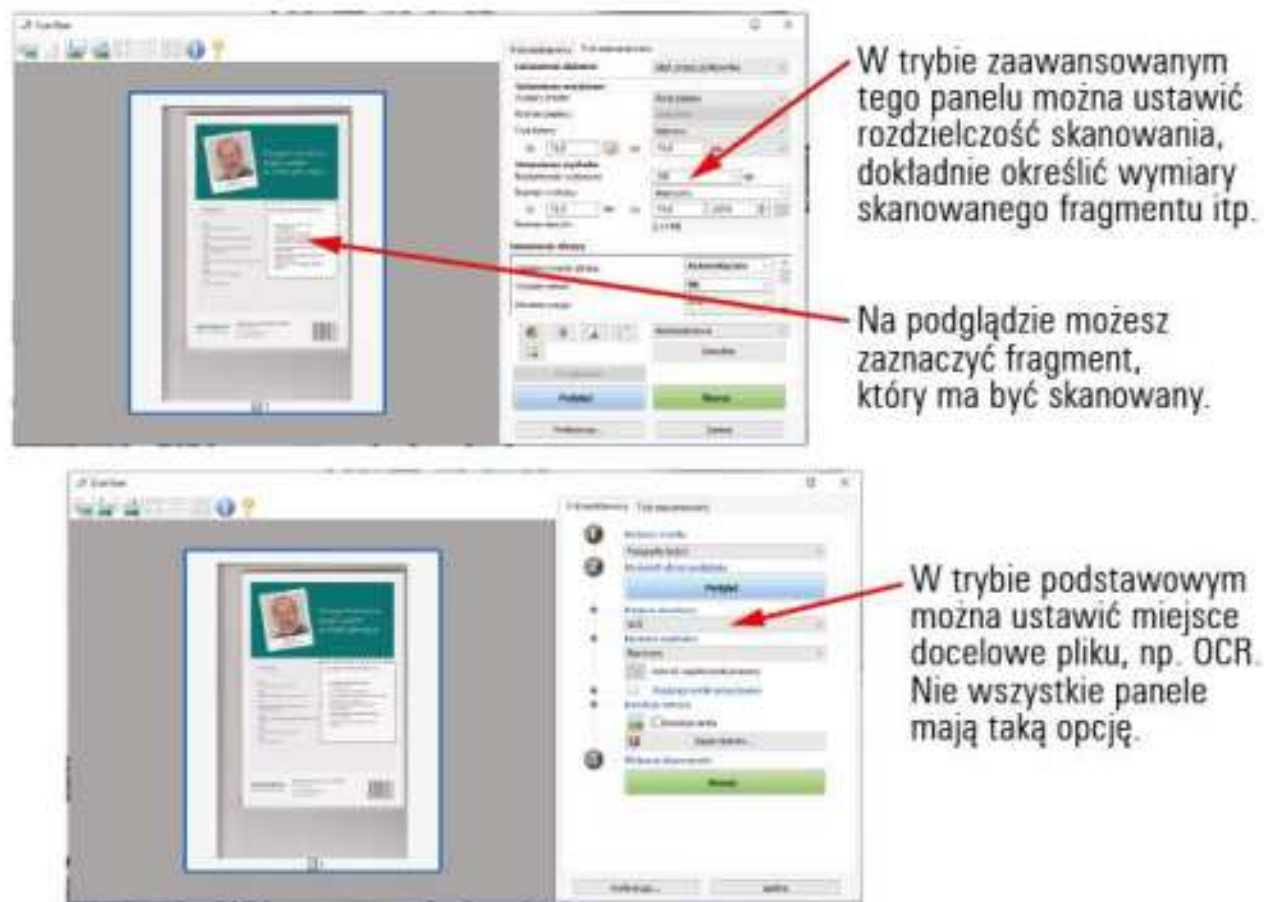
Programy OCR działają skutecznie dla dokumentów zeskanowanych z określoną rozdzielczością. Przyjęło się, że jeśli litery dobrze kontrastują z tłem, to rozdzielczością zapewniającą skuteczność OCR jest minimum 200 dpi (punktów na cal).

Przebieg procesu skanowania zależy od użytego programu. Mimo iż sterowniki skanerów dla systemów operacyjnych używają tego samego standardu komunikacji (protokołu transmisji) TWAIN, panele użytkownika mogą znacznie się różnić. Panel uruchomi się wraz z zainicjowaniem skanowania (rys. 21.4.).

TWAIN

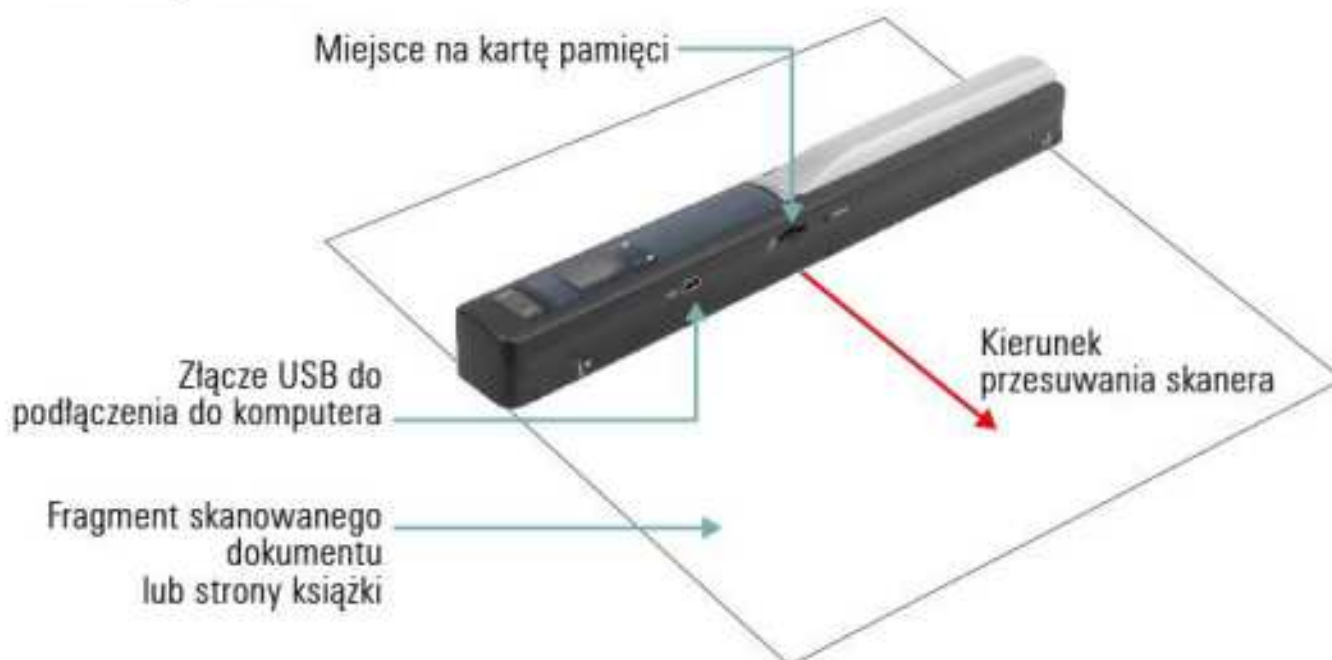


Instaluje się wraz ze sterownikami i stanowi nieocenione narzędzie pozwalające na wykorzystanie wszystkich zalet danego modelu skanera. W zastępstwie skanera można użyć aparatu fotograficznego. Doświadczalnie należy dobrać ustawienia, by tekst był czytelny i kontrastował z tłem.



Rys. 21.4. Przykładowy panel sterujący z ustawieniami parametrów skanowania

Użytkownicy OCR odwiedzający biblioteki czy archiwa zazwyczaj nie mogą korzystać ze skanerów stołowych. Z pomocą przychodzą urządzenia mobilne. Ręczne skanery podobne do długopisów pozwalają na skanowanie wybranych obszarów dokumentów, przechowywanie skanów w pamięci lub przesyłanie do komputera. Analiza i konwersja najczęściej odbywają się w komputerze PC (rys. 21.5.). Zauważ, że w zastosowaniach OCR nie ma znaczenia rodzaj technologii użytej do budowy skanera.



Rys. 21.5. Przykłady skanerów ręcznych do zastosowań OCR

21.2. Ostatnia deska ratunku, czyli OCR odzyskuje dokumenty

Przebieg procesu OCR przeanalizujemy na przykładzie programu działającego on-line. Spośród kilku dostępnych w sieci możesz wybrać *onlineocr.net*. Pod tym samym adresem znajduje się także jego komercyjna wersja. Bez założenia konta możesz poddawać procesowi do 15 stron na godzinę. Program nie ma wielu opcji, a proces przebiega automatycznie (rys. 21.6.).

konwersja
w programie OCR

W tej wersji do konwersji możesz przestać pliki w formacie **PDF, JPG, BMP, TIFF, GIF**, a formatami wyjściowymi mogą być **Word, Excel** lub plik tekstowy **TXT**.

Konwersja nastąpi po wciśnięciu przycisku **KONWERTUJ**.



Rys. 21.6. Konwersja OCR w programie *onlineocr.net* on-line

Pobieranie pliku wynikowego zainicjuj, klikając jego nazwę (poniżej przycisku **Wybierz plik...**).

Uwaga!

Po konwersji programem OCR zawsze sprawdzaj poprawność procesu. Może się zdarzyć, że litery zostaną źle rozpoznane, a wyraz zupełnie zmieni znaczenie.

Dokumenty często zawierają ilustracje. Czy OCR potrafi je rozpoznać? Wypróbuj to na przykładzie dowolnej strony internetowej. Użyj narzędzia **Wycinanie** (wbudowane w Windows) i zaznacz dany fragment okna przeglądarki. Zapisz obraz w jednym z obsługiwanych formatów. W przykładzie wykorzystano fragment, w którym mamy do czynienia z dwoma elementami graficznymi, w tym przyciskiem z napisem. W automatycznych OCR program może rozpoznać taki

przycisk jako tekst z tłem. Rozbudowane programy OCR dają możliwość zaznaczania fragmentów, które mają być rozpoznane jako ilustracja lub tabela.



Rys. 21.7. Przykład konwersji fragmentu WWW z ilustracją

Kolejnym zadaniem stojącym przed OCR jest rozpoznawanie tabel (rys. 21.8.). Często korzystamy z danych z różnych źródeł, na przykład opracowanych przez Główny Urząd Statystyczny i dostępnych na stronie *stat.gov.pl*. Pobrane dane najczęściej mają format PDF. Jeśli narzędziem **Wycinanie** zapiszemy w pliku graficznym fragment takiego dokumentu, będzie możliwa jego konwersja. Wykonanie tego zadania dostarczyło kilku niespodzianek (rys. 21.8.).

Jak widać, darmowe narzędzia OCR nie zawsze są dokładne w odwzorowywaniu dokumentu. Mimo tych wad są bardzo pożyteczne i zdają egzamin w praktyce szkolnej i nieskomplikowanych przypadkach dokumentów.

Tabela 1. LUDNOŚĆ WEDŁUG BIOLOGICZNYCH GRUP WIEKU (w tys. osób)
Stan w dniu 31 XII
POPULATION BY BIOLOGICAL AGE GROUPS (in thousand persons)
As of 31 XII

WYSCZEGÓLNIENIE	2012	2013	2014	2015	2016	SPECIFICATION
0-14 lat	5 797	5 771	5 764	5 755	5 773	0-14 years
15-64 lat	27 245	27 052	26 940	26 835	26 308	15-64 years
65 lat i więcej	5 488	5 673	5 674	6 076	6 303	65 years and more

Źródło: Baza Demografii GUS, www.demografia.stat.gov.pl/BazaDemografii/
Source: CSO Demography Database, www.demografia.stat.gov.pl/BazaDemografii/

Biorąc pod uwagę liczbę ludności według edukacyjnych grup wieku (wykres 3), można zauważyć, że liczba dzieci w wieku 3-6 lat zwiększała się w latach 2012-2013, potem nastąpił spadek (w 2016 r. o 114,1 tys. w porównaniu z 2013 r.). Jedyną grupą wykazującą roczny wzrost liczby była grupa dzieci 7-12 lat. W pozostałych przedziałach wieku liczba osób nadal się zmniejszała.

Rozpoznane adresy WWW zostały zamienione na aktywne odnośniki.

Tabela została rozpoznana prawidłowo. Tekst w tabeli wymaga formatowania i uzupełnienia o nierozpoznane znaki.

Tabela 1. LUDNOŚĆ WEDŁUG BIOLOGICZNYCH GRUP WIEKU (w tys. osób) Stan w dniu 31 XII
POPULATION BY BIOLOGICAL AGE GROUPS (in thousand persons) As of 31 XII

WYSCZEGÓLNIENIE	2012	2013	2014	2015	2016	SPECIFICATION
0-14 lat	5 797	5 771	5 764	5 755	5 773	0-14 years
15-64 lat	27 245	27 052	26 940	26 835	26 308	15-64 years
65 lat i więcej	5 488	5 673	5 674	6 076	6 303	65 years and more

Źródło: Baza Demografii GUS, www.demografia.stat.gov.pl/BazaDemografii/
Source: CSO Demography Database, www.demografia.stat.gov.pl/BazaDemografii/

Biorąc pod uwagę liczbę ludności według edukacyjnych grup wieku (wykres 3), można zauważyć, że liczba dzieci w wieku 3-6 lat zwiększała się w latach 2012-2013, potem nastąpił spadek (w 2016 r. o 114,1 tys. w porównaniu z 2013 r.). Jedyną grupą wykazującą roczny wzrost liczby była grupa dzieci 7-12 lat. W pozostałych przedziałach wieku liczba osób nadal się zmniejszała.

Rys. 21.8. Konwersja OCR dokumentu z tabelą



Jeśli zdecydujesz się na założenie konta na *onlineocr.net*, otrzymasz szereg dodatkowych możliwości, a liczba stron możliwych do konwertowania zostanie ograniczona do 50. Warto jednak zapoznać się z możliwościami tej wersji. Przesłane pliki mogą mieć do 200 MB objętości, liczba formatów wejściowych i wyjściowych znacznie się powiększa (rys. 21.9.). Możliwe jest także wybieranie stron z pliku PDF.

DESKA ROZDZIELCZA

The screenshot displays the 'DESKA ROZDZIELCZA' (Control Panel) of the onlineocr.net service. It is divided into four main sections: 1. 'Język rozpoznawania' (Recognition language) with a dropdown menu showing various languages including English (selected). 2. 'Formaty wyjściowe' (Output formats) with checkboxes for Adobe PDF, Microsoft Excel 97-2003 (.xls), Microsoft Excel (.xlsx), Microsoft Word 97-2003 (.doc), Microsoft Word (.docx), RTF document (.rtf), and Text Plain (.txt). 3. 'Dokument wielostronowy' (Multipage document) with radio buttons for 'Wszystkie strony' (All pages), 'Wybrane strony' (Selected pages) with a text input '1,2', 'Konwertuj na czarno-biały (zabijaj)' (Convert to black and white), and 'Złóż w plik wielostronowy (ZIP)' (Combine into a multipage file). 4. 'Typ dokumentu PDF' (PDF document type) with radio buttons for 'Wykryj automatycznie' (Detect automatically), 'Zoskanowany plik' (Scanned file) with a sub-option 'PDF', and 'Inny plik' (Other file) with a sub-option 'PDF'.

Rys. 21.9. Opcje wersji komercyjnej *onlineocr.net*

Uwaga!

Komercyjne programy znanych producentów oparte na OCR zapewniają większą skuteczność konwersji dzięki możliwości zaznaczania obszarów z grafiką, tabelami i tekstem.

ZADANIA DO ROZWIĄZANIA

1. Zrób zdjęcie jednej strony dowolnego czasopisma z ilustracjami. Wybierz taką, na której ilustracje nie zawierają liter. Zdjęcie skopiuj na dysk komputera. Dokonaj konwersji do pliku w formacie MS Word. Porównaj otrzymany dokument z rzeczywistą stroną czasopisma. Opisz krótko różnice. Zastanów się, skąd się wzięły. Jeśli dojdziesz do wniosku, że niektóre zostały spowodowane niedoskonałością zdjęcia, ćwiczenie wykonaj ponownie, starając się zmienić sposób wykonania zdjęcia.
2. Wczytaj do przeglądarki stronę z artykułem z dowolnego portalu informacyjnego. Wykonaj zrzut fragmentu ekranu z tekstem i ilustracją. Zapisz na dysku. Dokonaj konwersji do pliku MS Word dwoma różnymi programami on-line. Porównaj wyniki. Oceń oba dokumenty. Wskaż różnice.
3. Po wykonaniu zadania 2. znajdź w sieci opinie o obu programach. Porównaj je z własnymi wnioskami. Czy się pokrywają? Jeśli nie, znajdź przyczynę.
4. Załóż, że jesteś pracownikiem wydawnictwa i musisz wybrać program OCR dla dziennikarzy i pracowników redakcji. Zbierz informacje na temat najpopularniejszych programów tego typu i wybierz dwa z najlepszymi recenzjami. Wykonaj tabelę (np. w edytorze tekstu), w której porównasz oba, wpisując ich zalety i wady. Sprawdź, na jakich zasadach można z nich korzystać (zakup licencji stałej, dzierżawa, on-line itp.) i ile to kosztuje.
- 5*. Wczytaj wersje próbne programów wybranych w zadaniu 2. Przetestuj je. Sprawdź, czy potwierdzają się opinie znalezione w internecie. Wybierz jeden i uzasadnij swoją decyzję.



PODSUMOWANIE LEKCJI

- Rodzaj zastosowanego skanera do skanowania dokumentów na potrzeby **OCR** nie ma znaczenia. Liczy się rozdzielczość i dokładność odwzorowania oryginału. s. 156
- **CCD i CIS** to nazwy potocznie określające rodzaj skanera płaskiego. W rzeczywistości określają one rozwiązania techniczne, na podstawie których zbudowano skaner. s. 158
- **TWAIN** to nazwa standardu komunikacji (protokołu transmisji) pomiędzy skanerem a systemem operacyjnym. Jest on stosowany powszechnie w sterownikach do skanerów. s. 158
- Wybranie opcji skanowania w programie graficznym najczęściej uruchamia panel sterujący, w którym można ustawić parametry i kierować całym procesem. s. 158
- Automatycznie działające, najprostsze programy do konwersji na podstawie **OCR** nie zawsze dają pożądaný efekt. Trudność sprawiają im ilustracje z tekstem czy tabele. Programy OCR przetwarzają pliki graficzne zapisane w różnych formatach, np. JPG, PNG, TIFF, GIFF oraz tekstowe dokumenty PDF i eksportują do różnych formatów, np. Word, Excel, TXT, PDF. Poprawność konwersji należy dokładnie sprawdzić. Istnieją programy OCR działające on-line. s. 160



22. Dokumentujemy wydarzenia, czyli aktywna praca z aparatem fotograficznym

NA TEJ LEKCJI:

- poznasz najważniejsze funkcje aparatu fotograficznego;
- nauczysz się, jak poprawnie zrobić ciekawe zdjęcie;
- dowiesz się, w jaki sposób przygotować się do sesji zdjęciowej.

Odkąd cyfrowe aparaty fotograficzne stały się standardem, fotografia to jedno z ważniejszych zastosowań informatyki. Dzięki rozpowszechnieniu smartfonów z aparatami i łatwych w obsłudze edytorów każdy może spróbować swoich sił jako reporter lub fotograf. Czy wszyscy osiągają sukces? Dlaczego nie jest to takie proste, jakby się wydawało? Warto się nad tym zastanowić.

22.1. Ekspozycja, czyli zupełnie co innego niż myślisz

Aparaty fotograficzne montowane w smartfonach są w pełni zautomatyzowane. Ich oprogramowanie pozwala ponadto na korzystanie z ustawień tematycznych, jednak część użytkowników nie posługuje się tymi opcjami. Dlatego jakość zdjęcia zrobionego automatycznie należy poprawić w edytorze. Niestety nie zawsze jest to możliwe. Co zrobić, by zdjęcia były wyraźne i dobrze naświetlone? Oczywiście wszystko zależy od aparatu – ale są pewnie uniwersalne zasady, których przestrzeganie pozwoli ci uzyskać zadowalające efekty. Aby zrozumieć te zasady, musisz poznać podstawowe pojęcia związane z fotografią.

Ekspozycja – to kombinacja czasu naświetlania i przysłony (rys. 22.4.) dająca pożądaną przez fotografa naświetlenie matrycy przy zachowaniu stałej jej czułości. Ogólnie ujmując: krótki czas i otwarta przysłona mogą dać takie samo naświetlenie, co długi czas przy przymkniętej przysłonie. Oczywiście ich poziom zależy od warunków na planie zdjęciowym. W trybie automatycznym ustawienia ekspozycji są dopasowywane w zależności od wybranego programu tematycznego. Na przykład zdjęcia w trybie „Sport” będą wykonywane z krótkim czasem naświetlania, by obiekty nie były poruszone (rys. 22.1.). W trybie „Pejzaż” przysłona zostaje przymknięta, by zwiększyć głębię ostrości (rys. 22.2.).

**Uwaga!**

Przykłady nazw programów tematycznych w różnych aparatach mogą być inne niż w treści podręcznika.



Czas otwarcia migawki
1/15 sekundy,
otwór względny f/16,
krople wody poruszone



Czas otwarcia migawki
1/1000 sekundy,
otwór względny f/5,
krople wody „zamrożone”

Rys. 22.1. Wpływ czasu naświetlania na efekt końcowy fotografii



Czas otwarcia migawki
1/640 sekundy,
otwór względny f/11,
ostrość od planu
najbliższego po dalek



Czas otwarcia migawki
1/1000 sekundy,
otwór względny f/5,
ostrość na obiekcie,
pozostałe plany nieos

Rys. 22.2. Fotografie z dużą i małą głębią ostrości

możliwości
techniczne aparatu

W aparatach montowanych w telefonach większość efektów powstaje elektronicznie po wybraniu odpowiedniej opcji. Na przykład użycie opcji rozmycie tła da podobny efekt do zdjęcia motyla z rysunku 22.2. Użyte wraz z programem portretowym da bardzo ciekawe efekty, a rozmyte tło nie będzie rozpraszać odbiorcy, który skupi się na twarzy portretowanej osoby (rys. 22.3.).



Rozmyte tło portretu

Rys. 22.3. Efekt użycia programu rozmywającego tło w aparacie smartfona

czas naświetlenia
i przysłona

Czas naświetlenia	Otwór względny (przysłona)	Czułość matrycy
– czas, przez który migawka jest otwarta, czyli naświetla się matryca; – w zależności od konstrukcji aparatu można ustawić w zakresie od kilku sekund do tysięcznych części sekundy; – im krótszy, tym mniej poruszonych obiektów na zdjęciu, ale i „ciemniejsze” może być zdjęcie.	– decyduje o ilości światła, które po otwarciu migawki dociera do matrycy; – im mniejszy otwór, tym mniej światła, ale większa głębia ostrości, czyli bliskie i odległe plany mogą być ostre.	– czułość na światło; – im wyższa, tym większy szum na zdjęciu, ale można fotografować przy niewielkiej ilości światła; – im wyższej klasy aparat, tym mniejszy szum i lepsza ostrość na zdjęciach w wysokich czułościach.

Rys. 22.4. Podstawowe pojęcia dla aparatów fotograficznych

1. Ustawienia automatyczne dobrze sprawdzają się w naturalnych warunkach, gdy słońce nie świeci zbyt ostro i brak dużego zachmurzenia. Dlaczego?



Tab. 22.1. Niektóre cechy ustawień automatyki aparatu fotograficznego

Warunki	Efekt działania automatyki aparatu	Przyczyna
Duże nasłonecznienie	Jasne miejsca są dobrze odwzorowane, a cienie ciemne, np. twarz pod daszkiem czapki. Zdarza się, że najjaśniejsze punkty zostają prześwietlone i tracą detale – stają się jasną plamą.	Automatyka ustawia ekspozycję i ostrość na podstawie pomiaru światła. Mocno oświetlone obszary mają na to duży wpływ.
Zmrok, miejsca zacienione	Na zdjęciach występują większe szumy (pojawia się efekt „kaszy”). Głębia ostrości jest mniejsza. Na ciemnych miejscach nie widać szczegółów. Ruchome obiekty mogą być poruszone.	Aparat ustawia większą czułość matrycy. Im większa, tym większe szumy. Wydłuża się także czas naświetlania.
Obiekty w ruchu, np. rowerzysta, biegacz, dzieci	Obiekty w ruchu na zdjęciu są poruszone.	Od otwarcia do zamknięcia migawki obiekt przebędzie już pewną drogę. Efekt zamrożenia obiektu w ruchu uzyskuje się przez skrócenie czasu naświetlania matrycy. Automatyka ustawia czas zgodnie z pomiarem światła.
Zdjęcia z niewielkiej odległości	Mogą być nieostre, gdy tło będzie jaśniejsze. Na ekranie smartfona można dotykem ustawić miejsca ostrości, co znacznie ułatwia fotografowanie.	Automatyka samoczynnie ustawia ostrość na jasnych obiektach. W zależności od sposobu ostrzenia (w punkt lub na wielu punktach) można uzyskać różne efekty.

2. Warto stosować programy tematyczne, na przykład sportowe, tylne światło, rozmycie tła, by uzyskać ciekawe efekty i dobrej jakości zdjęcia.
3. Aby zdjęcia miały naturalne kolory, należy stosować automatyczną korektę, na przykład światło żarowe (żarówki, halogeny itp.) – ze zdjęcia zostanie usunięty nadmiar składowej czerwonej. Wykonane z tą nastawą zdjęcia w świetle dziennym będą miały niebieskie zabarwienie.



22.2. Reporter i jego aparat, czyli dokumentujemy wydarzenia

Dzięki łatwemu dostępowi do aparatów fotograficznych w komórkach każdy jest w stanie sfotografować ciekawe wydarzenie i opublikować relację. Ty także możesz zostać szkolnym dziennikarzem.

Uwaga!

Upubliczniając swoje zdjęcia lub relacje, musisz przestrzegać podstawowych zasad.

1. Nie publikuj wizerunków osób bez ich zgody. Wyjątek stanowią zdjęcia tłumu, np. młodzież na holu szkoły. Tu jednak także musisz uważać na to, czy ktoś nie został w jakiś szczególny sposób wyróżniony, na przykład ostrością, kolorem, wielkością na zdjęciu.
2. Nigdy nie publikuj zdjęć w złych intencjach, na przykład ośmieszenia kogoś, naruszenia prywatności.
3. Przekazuj rzeczywisty obraz wydarzeń. Zachowaj obiektywizm.
4. Podpisuj swoje relacje.
5. Działaj zgodnie z prawem. Jego nieznajomość nie zwalnia z odpowiedzialności.

przygotowanie do sesji zdjęciowej

Jeśli używasz swojego aparatu do zadań reporterskich, odpowiednio go przygotuj.

1. Ustaw automatykę w taki sposób, by jak największa liczba zdjęć nadawała się do publikacji, czyli była odpowiednio naświetlona i ostra.
2. Dbaj o odpowiednią ilość miejsca na karcie telefonu. Musisz liczyć około 4 do 10 MB na jedno zdjęcie. Warto sprawdzić, ile miejsca zajmują w twoim aparacie.
3. Noś przy sobie odpowiedni kabel do połączenia z komputerem lub czytnik kart. Jeśli swoje relacje upubliczniasz bezpośrednio z telefonu, miej zawsze aktywny pakiet internetowy. Zainstaluj odpowiednie oprogramowanie, na przykład aplikację portalu społecznościowego.
4. Naładuj baterię aparatu przed pracą, a jeśli to możliwe, zaopatrz się w zapasową lub powerbank.
5. Dobrze poznaj możliwości i opcje swojego aparatu.

Zdjęcia z twojego aparatu możesz wczytać do komputera, a następnie je edytować. W tym celu wykorzystasz bluetooth, WiFi lub kabel USB. Dobrym rozwiązaniem jest też odczytanie karty SD bezpośrednio w czytniku komputera. Sprawdź, jakie możliwości ma twój aparat.

przesyłanie zdjęć do komputera



Przykład

Założmy, że relacjonujesz mecz szkolnej drużyny. Fotoreportaż i artykuł musisz napisać jak najszybciej. Jak poradzić sobie z takim zadaniem?

Znasz składy drużyn, termin meczu, miejsce itp., więc część tekstu możesz ułożyć przed meczem. Jeśli przygotujesz relacje na dwa warianty wyniku, to uzupełnienie jej faktami nie zabierze ci dużo czasu.

Zdjęć jednak nie przygotujesz przed wydarzeniem. Zaplanuj swoją pracę, by ułatwić sobie zadanie. W czasie wydarzeń sportowych nie da się niczego powtórzyć czy ustawić ponownie. Zapoznaj się więc z miejscem akcji. Zaplanuj, skąd będziesz fotografować wydarzenia na boisku, a skąd publiczność, ławkę trenera, występy w czasie przerw. Ustaw odpowiednio swój aparat i przeprowadź próbę na miejscu. Jeśli mecz jest wyjazdowy i nie wiesz, jak wygląda hala rywalów, znajdź jej zdjęcia w internecie. Narysuj plan swojej pracy, zaznacz miejsca, w których spodziewasz się dobrego widoku. Zaplanuj, w jakiej fazie spotkania staniesz w tych miejscach. W razie potrzeby, na przykład gdy wydarzy się coś nieprzewidywalnego, reaguj spontanicznie i improwizuj. Jednak twój aparat musi być w ciągłej gotowości.

Nie publikuj wszystkich zdjęć. Wybierz takie, które możesz ułożyć w logicznej kolejności. Twój fotoreportaż powinien opowiadać o tym wydarzeniu, by odbiorca poczuł atmosferę i dramaturgię wydarzenia.

Jest więcej zasad, które mogą sprawić, że twoja praca będzie ciekawa. Poczytaj o nich na stronach internetowych i blogach fotografów (rys. 22.5.).

Utrzymaj rytm

Zdjęcia pasujące do siebie warto rozłożyć „rytmicznie”. Na poniższym przykładzie połączono zdjęcia o podobnych cechach (miejsce i kolorystyka) na grupy:

- z wałów przeciw powodziowych,
- ze szkoły dla powodzian.



Rytm 111222. To mogłyby być 2 różne reportaże.



Rytm 121212. Dwa miejsca akcji, ale jeden reportaż. W tym przypadku ten rytm lepiej łączy zdjęcia.

Rys. 22.5. Fragment strony poświęconej komponowaniu fotoreportaży dg.art.pl/zasady-komponowania-reportazu-fotoreportazu/

22.3. Przypomnij sobie, czyli krótko o kompozycji obrazu

kadrowanie

Zasady komponowania obrazu nie dotyczą tylko fotografii. To ogólnie stosowane sposoby dobrego przekazu za pomocą obrazu. Mają zastosowanie w sztuce, grafice komputerowej, stronach internetowych, grach itp. Spotkałeś/spotkałaś się z nimi nie tylko na lekcjach informatyki, lecz także innych przedmiotów.

1. Przed zrobieniem zdjęcia zawsze pomyśl o tym, co chcesz na nim przedstawić. Zwracaj uwagę na detale, które mogą skupiać na sobie uwagę odbiorcy, ale nie są tematem zdjęcia (rys. 22.6.).
2. Jeśli nie masz doświadczenia lub pomysłu na kompozycję, stosuj znane reguły, na przykład trójpodziału. Sprawdza się ona w większości przypadków. Podczas kadrowania najważniejszy obiekt umieść w silnym punkcie, czyli w jednym ze skrzyżowań linii trójpodziału (rys. 22.7.). Jeśli fotografujesz pejzaż, możesz wykorzystać pionowe lub poziome linie i zgrać z nimi linie obrazu, na przykład podłużny obiekt, brzeg morza, drzewa itp. Wiele aparatów fotograficznych i w telefonach ma opcję nałożenia na podgląd linii trójpodziału – wykorzystaj ją.

reguła trójpodziału



Rys. 22.6. Przykład zdjęcia zakłóconego przez zbędne elementy

Elementy odwracające uwagę od głównego tematu zdjęcia.



Rys. 22.7. Przykład zastosowania reguły trójpodziału

Czytanie o fotografowaniu i komponowaniu reportaży nie nauczy cię praktycznych umiejętności. Musisz wziąć aparat do ręki, wykonać zdjęcia, a potem samodzielnie je posegregować i ułożyć. Im więcej ćwiczysz i sprawdzania się w akcji, tym lepsze będą wyniki. Próbuje więc i baw się dobrze, pamiętając, że opisane w rozdziale przykłady są niewielkim fragmentem wiedzy na temat fotografowania.

ZADANIA DO ROZWIĄZANIA

1. a) Znajdź w internecie instrukcję obsługi swojego aparatu fotograficznego lub telefonu z aparatem. Wskaż programy tematyczne lub nastawienia automatyki i omów ich przeznaczenie. Sprawdź ich działanie, fotografując ten sam obiekt w podobnych warunkach.



- b) Przygotuj plik, w którym umieścisz obok siebie zdjęcia wykonane w punkcie a. Dla każdego z nich określ, jaki program lub ustawienie automatyki zostały użyte do jego wykonania. Uzasadnij odpowiedź.
2. Wyszukaj w internecie strony lub blogi redagowane przez zawodowców na temat fotografowania wydarzeń sportowych itp. Zbierz najważniejsze rady profesjonalistów i wykonaj prezentację z przykładami. Możesz wykorzystać fragmenty znalezionych stron w charakterze cytatów.
 3. Omów w sieci przykłady zastosowania reguły trójpodziału w sztuce na przestrzeni wieków. Opisz kilka ciekawych przykładów, np. na obrazach Jana Matejki. Sporządź prezentację, w której przedstawisz najciekawsze przypadki.
 4. Wykonaj zdjęcia wody w ruchu, np. ciekącej z kranu, z różnymi ustawieniami migawki lub programów tematycznych. Uzyskaj dwa efekty – woda jest w ruchu (czyli krople są poruszone) i ruch wody jest „zamrożony”. Wykonaj prezentację na temat zjawisk, które pozwoliły na uzyskanie takich efektów.
 5. Wybierz ze swoich zbiorów kilka zdjęć z uroczystości, np. zakończenia roku szkolnego. Skomponuj z nich fotoreportaż. Skorzystaj z rad podanych na blogu dg.art.pl/zasady-komponowania-reportazu-fotoreportazu/ lub innej, wiarygodnej strony na ten temat. Wydrukuj zdjęcia lub utwórz elektroniczny album i zaprezentuj je znajomym nieuczestniczącym w wydarzeniu. W rozmowie zapytaj o ich odczucia i sprawdź, czy odczytali przekaz zgodnie z twoją intencją.

PODSUMOWANIE LEKCJI

- Każdy aparat fotograficzny nadaje się do dokumentowania wydarzeń. Należy optymalnie wykorzystać jego **możliwości techniczne**. s. 166
- Optymalne naświetlenie matrycy może być wynikiem różnych kombinacji **czasu naświetlania i przysłony**. Dają one jednak różniące się efekty, np. „zamrożenie” lub poruszenie obiektu, płytką lub głęboką głębią ostrości. Świadome korzystanie z ręcznych ustawień ekspozycji lub programów tematycznych może znacznie poprawić **jakość zdjęcia**. Ma to także wpływ na treść fotografii, np. poprzez wyostrenie niektórych obiektów, rozmycie tła. s. 166
- Przed sesją zdjęciową należy zadbać o naładowanie akumulatorów, przygotowanie dodatkowego źródła zasilania, zabezpieczenie odpowiedniej ilości pamięci na karcie. Jeśli będziesz fotografować wydarzenie sportowe, uroczystość szkolną itp., postaraj się zapoznać z miejscem akcji, przygotować plan miejsc, w których staniesz itd. s. 168
- Zdjęcia z aparatu możesz przysyłać do komputera za pośrednictwem **łączy bezprzewodowych bluetooth lub WiFi, kablem USB** lub poprzez odczyt karty wyjętej z urządzenia. s. 168
- W czasie fotografowania zwróć uwagę, czy w **kadrze** nie znajdują się niepotrzebne, przykuwające wzrok obiekty. Nie wszystko da się usunąć w programie komputerowym. s. 170
- Jeśli nie masz doświadczenia lub pomysłu na skadrowanie zdjęcia, skorzystaj z **reguły trójpodziału**. Możesz wykorzystać opcję nałożenia jej na podgląd w swoim aparacie. s. 170



23. Szturmowiec w chmurze, czyli poprawiamy zdjęcia w edytorze grafiki rastrowej

NA TEJ LEKCJI:

- poznasz, jakie są niekomercyjne edytory grafiki rastrowej;
- użyjesz narzędzi niekomercyjnych programów do obróbki zdjęć.

Na pewno masz już pewne doświadczenia związane z przetwarzaniem zdjęć. Wiesz, że można je kadrować, zmieniać jasność, barwy, a także że do tych czynności są potrzebne odpowiednie narzędzia. Do prostych prac nad fotografią nie trzeba wykorzystywać komercyjnych programów graficznych. Istnieje wiele darmowych programów komputerowych, które z powodzeniem zastąpią niektóre narzędzia profesjonalne. Są wśród nich również programy w chmurze. Gdy dojdiesz do wprawy w posługiwaniu się nimi, znacznie szybciej opanujesz narzędzia profesjonalne.

edytory grafiki
rastrowej
w chmurach
informatycznych

23.1. Warstwy w chmurach, czyli edytujemy zdjęcie

Wśród edytorów chmurowych możesz znaleźć także takie, których interfejs i podstawowe czynności edycyjne są bardzo zbliżone do profesjonalnych programów, takich jak PhotoShop. Jednym z nich jest **Pixlr** (www.pixlr.com). Dzięki niemu wykonasz wszystkie podstawowe czynności edycyjne. Oczywiście wiele z nich pozwala również na zaawansowaną edycję. Warto się z nimi zapoznać. Opisane ćwiczenia możesz także wykonać z pomocą innego edytora grafiki.

Nowy edytor najlepiej poznać poprzez ćwiczenia.

Przykład

Pewien reporter wybrał się na zlot miłośników filmów *science fiction*. Jedno ze zdjęć wykonał z dużej odległości spośród tłumu uczestników. Trudno mu było odpowiednio ustawić kadr. Na fotografii znalazło się też kilka niepotrzebnych elementów. Należy więc poddać ją obróbce elektronicznej. Spośród swoich zdjęć wybierz takie, które wymaga retuszu i poddaj je podobnym zabiegom, jak w poniżej opisanym ćwiczeniu.

Użyj programu z chmury informatycznej *pixlr.com* (rys. 23.1.).

Przeanalizuj przykładowe zdjęcie i porównaj ze swoim. Być może będzie miało te same wady.

Analiza zdjęcia

1. Wymaga kadrowania. Prawie połowa powierzchni jest czarna.
2. Na dole kadru głowa jednego z uczestników zbyt przykuwa uwagę. Niczego nie wnosi do zdjęcia.
3. Po prawej stronie głównej postaci widać fragment hełmu innego uczestnika.

Jak pozbyć się tych niedoskonałości? Użyjmy narzędzi programu Pixlr. Pracę nad korektą rozpocznij od kadrowania. Dla pełnej swobody zaznaczania obszaru dla narzędzia **Kadrowanie** wybierz parametr **Brak ograniczenia**.



Rys. 23.1. Zdjęcie wczytane z dysku do edytora pixlr.com w chmurze



Rys. 23.2. Kadrowanie zdjęcia w pixlr.com

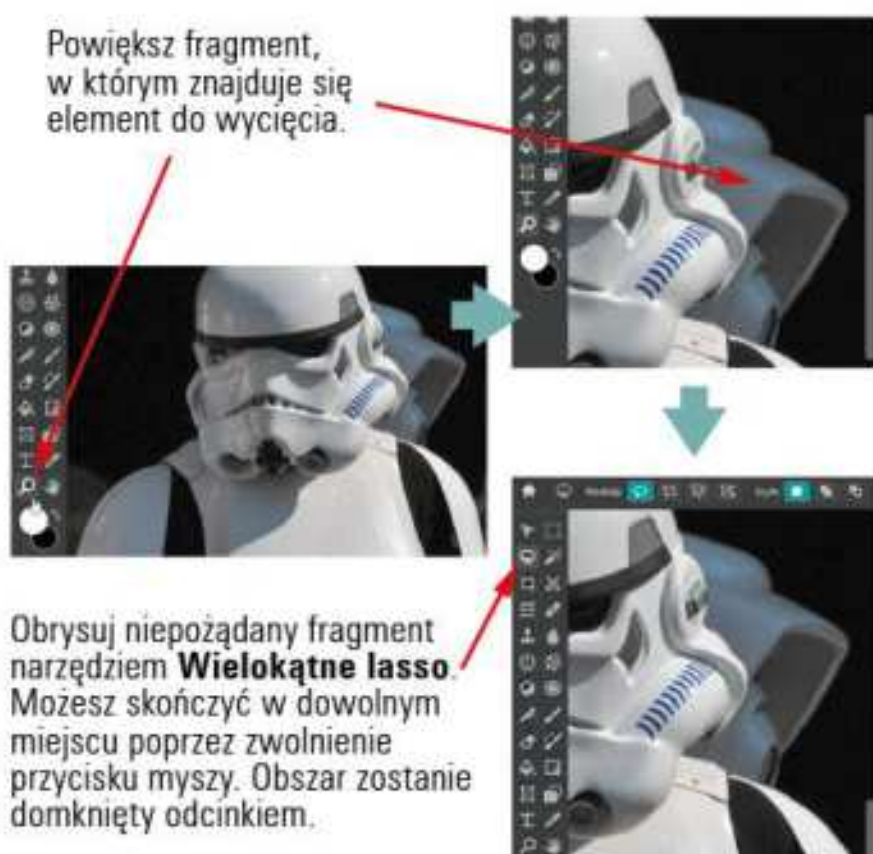
zapisywanie
wyniku pracy

Wynik swojej pracy co pewien czas zachowuj. Rób to, pobierając kopię na dysk swojego komputera. Oczywiście te same czynności wykonaj po zakończeniu pracy.



Rys. 23.3. Zachowanie zdjęcia edytowanego w chmurze pixlr.com

Czas na usunięcie niechcianego elementu. Posłuż się narzędziem **Lasso**. Wybierz **Wielokątne lasso** i zaznacz pole **Wygładzanie**. Lasso tego typu wyeliminuje problem dokładności zaznaczania, ponieważ będzie możliwe zaznaczanie obszaru za pomocą odcinków (dowolnej długości), co w połączeniu z wygładzaniem krawędzi da oczekiwany efekt. W przypadku korzystania z **Odręcznego lassa** dokładność zaznaczania zależy od precyzji prowadzenia kursora i rozdzielczości myszki. Sprawdza się przy zaznaczaniu skomplikowanych kształtów. Opcja **Pióro** ustala stopień wygładzania. Im większa liczba, tym większy łuk wygładzania. Ponieważ granica zaznaczanego obszaru może być stosunkowo długa, dobrym sposobem jest zaznaczenie fragmentu (rys. 23.4.), wykonanie na tym obszarze odpowiednich działań i powtórzenie ich dla następnych zaznaczeń. Jeśli precyzyjnie operujesz myszką, możesz zaznaczyć cały obszar w jednej operacji.



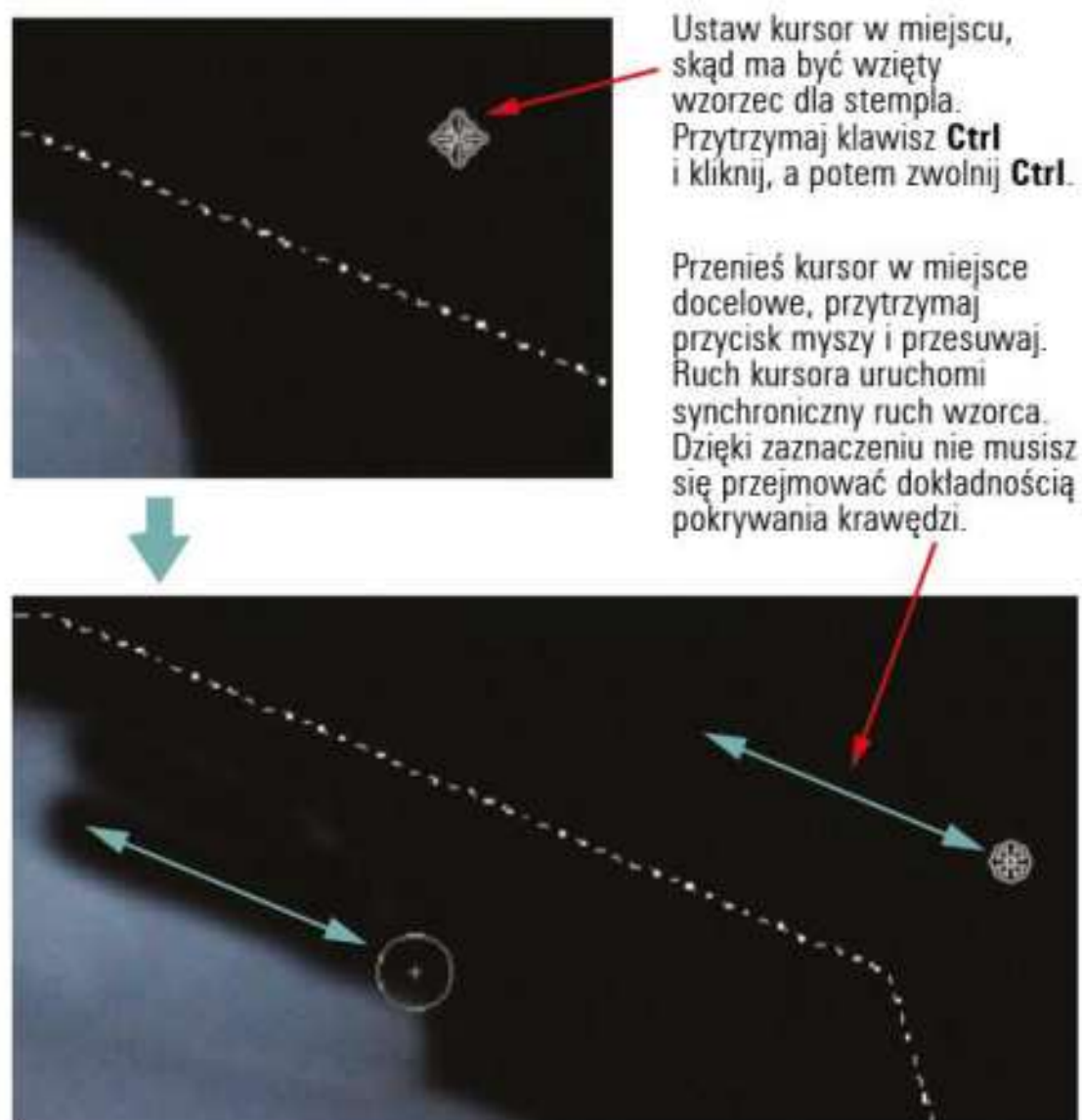
Rys. 23.4. Zaznaczanie narzędziem Wielokątne lasso



Zaznaczony obszar należy wypełnić. W tym przypadku użycie konkretnego koloru nie zda egzaminu. Mimo że tło w przykładzie wydaje się jednolite, w rzeczywistości takie nie jest – ma sporo odcieni. Użyjmy więc narzędzia **Stempel** (rys. 23.5., 23.6.). Jest ono bardzo przydatne przy retuszu zdjęć.



Rys. 23.5. Proces użycia narzędzia Stempel



Rys. 23.6. Efekt użycia narzędzia Stempel



Rys. 23.7. Opcja cofnięcia czynności w edytorze

Efekt, jaki uzyskasz, zależy od precyzji i cierpliwości. W razie nieudanych posunięć możesz wykorzystać z menu **Redaguj** opcję **Cofnij** lub **Ctrl+Z** (rys. 23.7.).

Podobne czynności możesz wykonać na przykład podczas usuwania fragmentu broni szturmowca.

Zauważ, że jego lewe ramię sprawia wrażenie, jakby kończyło się nienaturalnie tuż za zbroją. W rzeczywistości tak nie jest. Poprawmy ten mankament (rys. 23.8.).



Rys. 23.8. Rozjaśnianie fragmentu zdjęcia



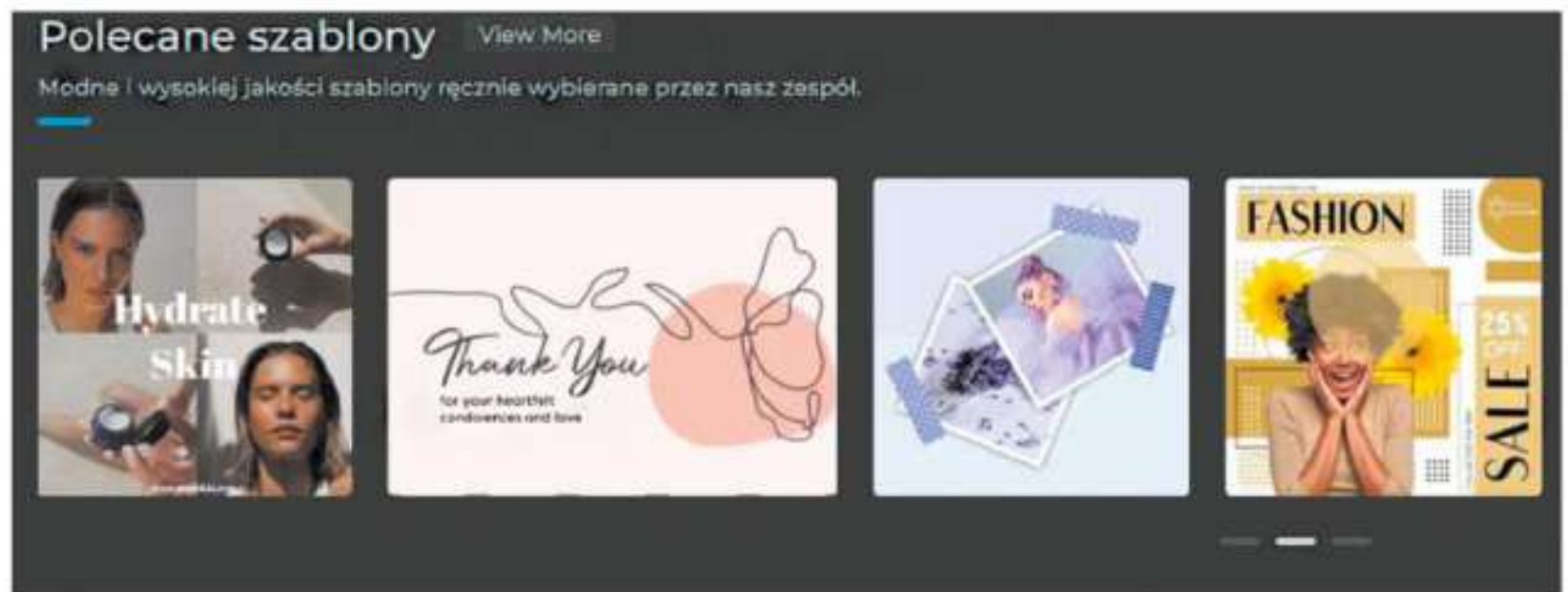
Rys. 23.9. Zdjęcie po korekcie

Jeśli dokładnie wykonałeś/wykonałaś swoje ćwiczenia, to zapewne powstały efekt jest dobry. Teraz możesz użyć go w swoich publikacjach. Użyte narzędzia działają podobnie w innych programach, w tym także komercyjnych. Mogą różnić się liczbą parametrów, które można ustawiać, jednak ogólne zasady ich stosowania są bardzo podobne. Ćwiczenia z darmowymi edytorami dają dobrą podstawę do poznania i używania zaawansowanych edytorów.

Pixlr możesz także wykorzystać do tworzenia grafiki, podobnie jak GIMP w jednym z poprzednich ćwiczeń. Jeśli jesteś dociekliwy/dociekliwa, znajdziesz na stronie głównej jeszcze inne



niespodzianki. Przykładem może być zbiór plików graficznych, które można wykorzystać w swoich projektach (rys. 23.10.).



Rys. 23.10. Darmowe grafiki ze strony pixlr.com

Jeśli nie skończysz edycji w Pixlr, a używane były warstwy, zachowaj swoje zdjęcie lub grafikę w formacie PXD. Dzięki temu będzie możliwa dalsza edycja w późniejszym terminie. Podobnie postępuj z innymi edytorami. Część programów do edycji zdjęć ma także swoje odpowiedniki dla smartfonów. Są one zazwyczaj pozbawione wielu narzędzi z powodu obsługi za pomocą ekranu dotykowego, co znacznie ogranicza precyzję operowania (rys. 23.11.).



Rys. 23.11. Pixlr w wersji mobilnej

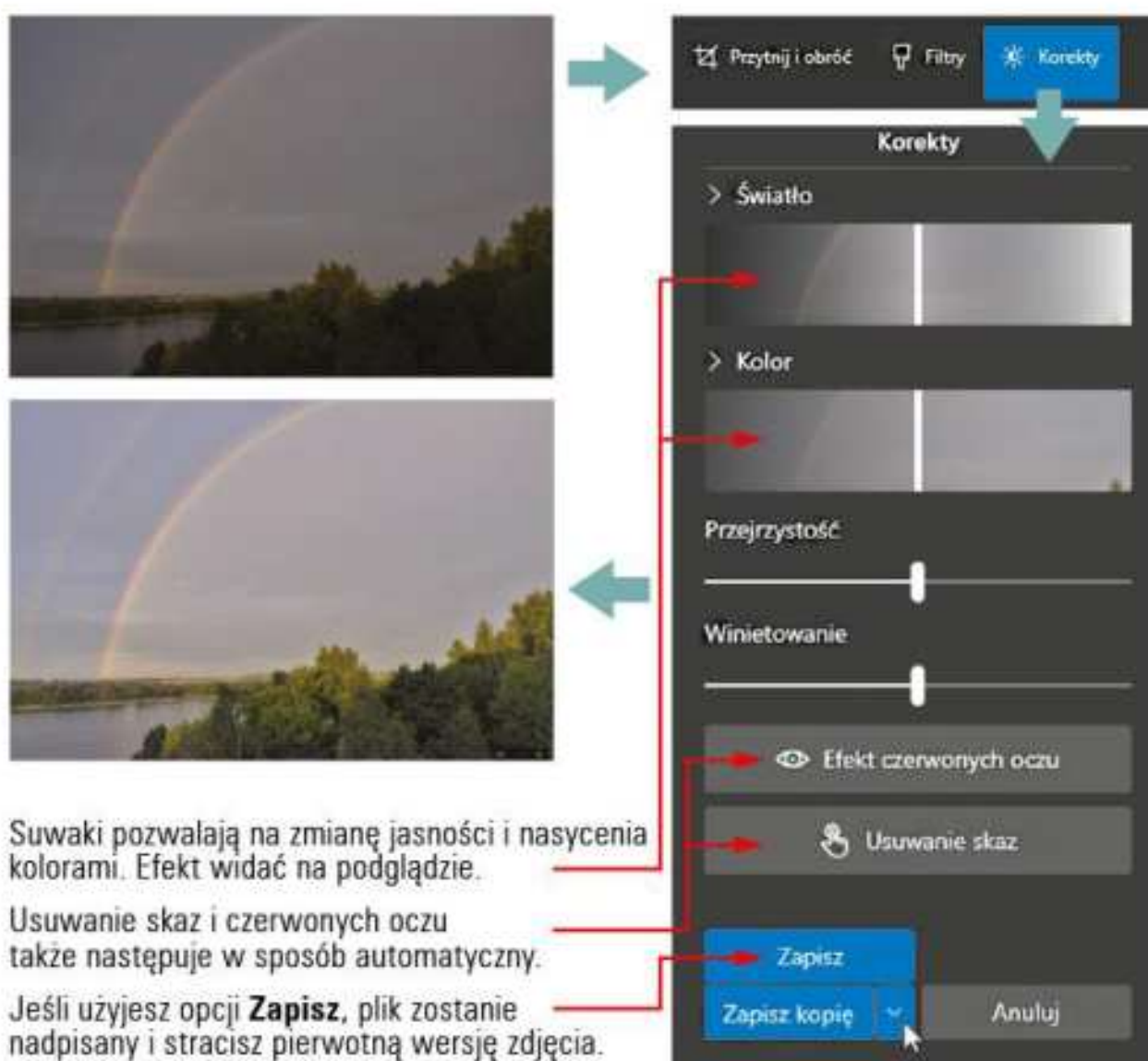
23.2. Nie wyważaj otwartych drzwi, czyli narzędzia do korekty grafiki

Do tej pory zajmowaliśmy się edycją szczegółów fotografii lub grafiki. Obok niedokładnego kadrowania najczęściej występującą wadą zdjęć są źle naświetlone kadry. Przyczyną mogą być złe warunki oświetleniowe. Zdjęcie może więc być niedoświetlone, prześwietlone lub zabarwione, na przykład światłem odbitym od kolorowej ściany. Korektę można przeprowadzić zaawansowanym edytorem, jak GIMP czy PhotoShop.

uproszczone
edytory zdjęć

Jednak w wielu sytuacjach, szczególnie w przypadku zdjęć amatorskich, wystarczy użycie prostszych programów, na przykład dostępnej w Windows aplikacji **Zdjęcia**.

Przykładowe zdjęcie jest niedoświetlone. Użyjemy programu **Zdjęcia** i poprawmy tę wadę. Miniaturę pliku zdjęcia kliknij prawym przyciskiem myszy i wybierz **Otwórz za pomocą \ Zdjęcia**.



Rys. 23.12. Edycja podstawowych parametrów programem „Zdjęcia”

Program **Zdjęcia** (dostępny w Windows) jest dowodem na to, że nieskomplikowane programy mogą także być skuteczne w amatorskim przygotowaniu fotografii do publikacji. Podobnych aplikacji można znaleźć więcej. Warto z nich korzystać, by nie instalować zaawansowanych programów do wykonania prostych czynności.



ZADANIA DO ROZWIĄZANIA

1. Użyj wyszukiwarki internetowej i znajdź kilka edytorów grafiki rastrowej. Porównaj ich możliwości i sposób posługiwania się podstawowymi narzędziami edycyjnymi z Pixlr. Opisz różnice i ustaw programy w kolejności od najlepszego do najmniej przydatnego. Ponieważ będzie to twoja subiektywna ocena, uzasadnij swój wybór.
2. Wykorzystaj wybrany w zadaniu 1. sieciowy edytor grafiki rastrowej (nie Pixlr) do zmiany nasycenia barw, jasności i kontrastu zdjęcia z własnej kolekcji. Staraj się zapamiętać działanie poszczególnych opcji i narzędzi.
3. Załóż konto za pośrednictwem programu Pixlr i sprawdź, jakie są zalety posiadania takiego konta. Omów funkcje i przydatność **Biblioteki obrazków**.
4. Uruchom program **Zdjęcia**. Odnajdź opcję **Utwórz film z muzyką**. Wykorzystaj ją do utworzenia filmu ze zdjęć z własnej kolekcji. Postaraj się odpowiednio dobrać kolejność i tematykę zdjęć. Zaprezentuj swoje dzieło i uzasadnij dobór zdjęć i ich kolejności.
5. Sprawdź działanie pozostałych opcji programu **Zdjęcia**. Dodaj do zdjęcia ze swojego zbioru efekty 3D (opcja **Dodaj efekty 3D**). W portretach usuń efekt czerwonych oczu. Podaj kilka przykładów wykorzystania dodanych efektów.

PODSUMOWANIE LEKCJI

- Większość czynności podczas amatorskiej obróbki zdjęć możesz wykonać za pomocą **niekomercyjnych programów**. s. 172
- Edytory grafiki rastrowej znajdziesz także w **chmurach informatycznych**. s. 172
- **Narzędzia i opcje niektórych niekomercyjnych edytorów** zdjęć lub grafiki rastrowej są podobne do narzędzi komercyjnych. Różnią się liczbą parametrów i zakresem działania. Opanowanie ich w programie darmowym znacznie ułatwi późniejszą pracę z profesjonalnymi programami. s. 173
- Jeśli wykonujesz wiele prób **modyfikacji zdjęcia**, zawsze zachowuj je w oddzielnych plikach. Dzięki temu zachowasz oryginał zdjęcia. s. 174
- **Uproszczone edytory zdjęć** przydają się w szybkim kadrowaniu, regulacji jasności i nasycenia kolorów. Ułatwiają też likwidację skaz i tzw. czerwonych oczu. s. 178

24. Tego tu nie było, czyli poprawiamy rzeczywistość na fotografiach

NA TEJ LEKCJI:

- przypomnisz sobie zasady pracy z warstwami;
- poznasz narzędzia do nakładania warstw na zdjęciu.

Umiesz już wykorzystać edytor grafiki rastrowej do kadrowania i poprawiania jakości zdjęć. Posługujesz się edytorami grafiki z chmur informatycznych i aplikacjami lokalnymi. Twoje zdjęcia są coraz lepsze i możesz ich używać do tworzenia różnych elementów graficznych, na przykład tła, nagłówków stron, banerów reklamowych. W tego rodzaju projektach bardzo przydatne są warstwy. Używałeś/używałaś ich już podczas tworzenia trójwymiarowego napisu. Czas na zastosowanie w edycji zdjęcia.



Rys. 24.1. Kadr z filmu wykonany z wykorzystaniem warstw

24.1. Warstwy jak folie, czyli odbudowujemy fasadę

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, otrzymamy początkowy kadr filmu animowanego (rys. 24.1.). Teraz wystarczy powtórzyć to dla następnych etapów poruszania się bohatera. Dysponując takimi zdjęciami, możemy utworzyć film animowany. Dokładnie takiej metody używali twórcy filmów rysunkowych, zanim wprowadzono animacje komputerowe.

warstwy w edycji zdjęcia

Wykorzystajmy warstwy do operacji na fotografii. Użyj programu GIMP lub innego edytora grafiki rastrowej obsługującego warstwy, na przykład pixlr.com.

Użyj fotografii ze swoich zbiorów, która wymaga korekty ze względu na znajdujące się na niej zbędne elementy. Trzeba je usunąć poprzez przykrycie innym fragmentem, na przykład rośliną, latarnią. W przykładzie użyto zdjęcia wejścia do grobowca ozdobionego czterema rzeźbami, a właściwie trzema, ponieważ jedna z nich została zniszczona. Czy można to naprawić i przywrócić prawdopodobny wygląd tego miejsca sprzed zniszczenia? W rzeczywistości będzie to trudne, ale za pomocą edytora grafiki możesz „wyrzeźbić” brakujący fragment posągu.

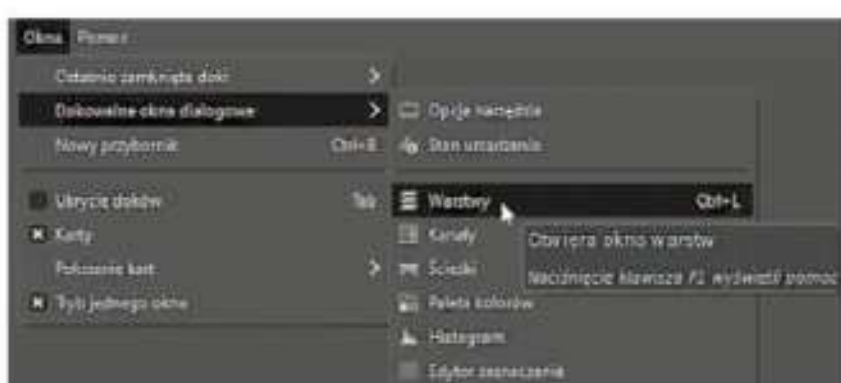


Pracę rozpocznij od wczytania zdjęcia do edytora (rys. 24.2). W razie potrzeby popraw kadrowanie, jasność, kontrast i nasycenie kolorami. Opis dotyczy edytora GIMP. Użycie innego nie powinno sprawić ci problemów. Narzędzia i sposoby ich użycia czy obsługa warstw są bardzo podobne w różnych programach. Musisz jednak odnaleźć odpowiadające sobie narzędzia i opcje. Zanim rozpoczniesz edycję sprawdź, czy okno z warstwami jest widoczne (rys. 24.3.).



Rys. 24.2. Oryginał zdjęcia starożytnych posągów

Obejrzyj dokładnie swoje zdjęcie i zaplanuj działania. Zlokalizuj fragment do poprawy i określ zakres prac. Gdy już ustalisz wielkość i kształt obszaru, dopasuj do niego fragment, którym go zakryjesz. Może on pochodzić z innego zdjęcia. Kolejność następnych działań może być taka, jak na rysunku 24.4.



Rys. 24.3. Otwieranie okna Warstwy w GIMP

Przed edycją upewnij się, czy okno warstw jest aktywne. Możesz je otworzyć z menu głównego.

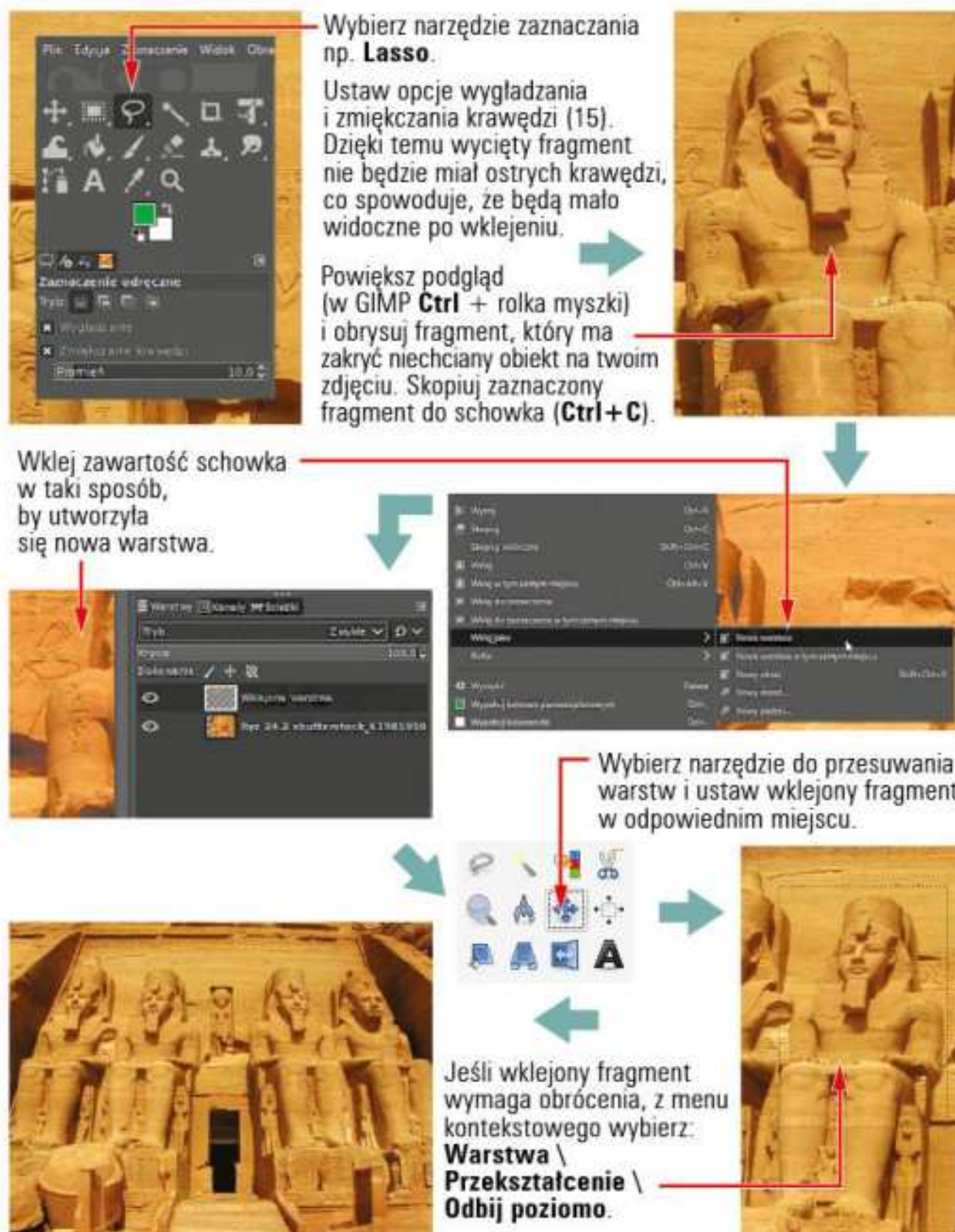


Rys. 24.4. Proces wycinania i wklejania fragmentów zdjęcia za pośrednictwem schowka i warstw

Efekt końcowy zależy już tylko od twojej inwencji, cierpliwości i sprawności posługiwania się myszką. Pomocny będzie rysunek 24.5.

narzędzie
zaznaczania

wygladzanie
i zmiękczenie
krawędzi



Rys. 24.5. Proces rekonstrukcji z wykorzystaniem innego fragmentu zdjęcia

Uwaga!

Fragment rysunku obrysowuj odcinkami o różnej długości, klikając **Lassem** w odpowiednie miejsca (rys. 24.6.). Większość kątów między poszczególnymi fragmentami zostanie zaokrąglona przez program (proces wygladzania) po zakończeniu zaznaczania. Każdy z punktów końcowych odcinków możesz przemieszczać, dzięki temu łatwo poprawisz niedokładności.



Rys. 24.6. Użycie narzędzia „Lasso” do zaznaczania odcinkami

Twoje zdjęcie zostało poprawione. Do czego można wykorzystać tę technikę? Spore możliwości daje podczas odrestaurowywania starych fotografii. Jeśli masz takie w rodzinnych zbiorach, możesz przeprowadzić renowację i wydrukować. Często to jedyna zachowana odbitka i kopia może być atrakcyjnym upominkiem dla członków rodziny (rys. 24.7.).

renowacja
starych zdjęć



Rys. 24.7. Archiwalne zdjęcie po renowacji

Innym pomysłem na wykorzystanie warstw i kopiowanie fragmentów zdjęć może być przygotowanie zdjęć znanych miejsc na konkurs „Znam naszą okolicę”. Fałszywe i autentyczne zdjęcia mogą ilustrować ciekawe pytania (rys. 24.8.).



Rys. 24.8. Przykład montażu zdjęcia na potrzeby konkursu



ZADANIA DO ROZWIĄZANIA

1. Obejrzyj dokładnie zdjęcie wykorzystane w ćwiczeniu. Jeśli znajdziesz inne niedoskonałości, napraw je znaną ci już metodą. O końcowym efekcie w dużej mierze zadecyduje wybór obszaru użytego do ich zakrycia.
2. Wróć do edycji zdjęcia. Zbadaj różnice pomiędzy zaznaczaniem obszarów z różnymi nastawami parametru **Zmiękczenie krawędzi**. Sprawdź także wpływ opcji **Wygladzanie** na efekt zaznaczenia. Oceń ich wpływ na skopiowany i wklejony fragment. Określ granice (wartość nastawy), przy których wklejony fragment odróżnia się od tła i gdy wtapia się niezauważalnie.
3. Znajdź w swoich zbiorach zdjęcie ulicy lub placu, na którym jest mało zieleni. Wklej do niego odpowiednio skalowane drzewa i inne rośliny w taki sposób, by pokazać, jak mogłaby wyglądać okolica, gdyby zadbano o zieleni. Wykorzystaj fragmenty innych zdjęć, np. z alei parkowej, lasu.
4. Wykonaj lub znajdź w swoich zbiorach zdjęcie dużego drzewa. Po uzyskaniu zgody kolegów i koleżanek wykonaj ich portrety. Umieść ich miniatury na drzewie. Dołóż tekst zawierający nazwę szkoły, rok i klasę (użyj oddzielnej warstwy). W ten sposób powstanie ciekawa pamiątka szkolna. Takie zbiorowe zdjęcie grupy, np. klasy, nazywane jest tablo.
5. Zeskanuj stare rodzinne zdjęcia. Wybierz te najbardziej zniszczone i przeprowadź ich renowację. Przed uzupełnianiem braków i zagnieceń popraw kontrast i jasność zdjęcia. Jeśli zajdzie potrzeba zmiany tych parametrów jedynie dla fragmentu zdjęcia (np. twarzy lub detalu), wykorzystaj warstwę. Przed skanowaniem ustaw odpowiednią rozdzielczość skanera, np. 300 dpi.

PODSUMOWANIE LEKCJI

- s. 180 • **Warstwy** są jednym z najważniejszych mechanizmów tworzenia grafiki i montażu oraz retuszu zdjęć.
- s. 181 • Każdą warstwę możesz edytować osobno. Możesz włączać i wyłączać **widzialność każdej warstwy**. Warstwę można także **skalować i obracać**, co ułatwia dopasowanie wklejanego elementu.
- s. 182 • Do zaznaczania obszarów możesz użyć dowolnego **narzędzia zaznaczania**. Wybieraj najodpowiedniejsze do danego kształtu.
- s. 182 • Wklejony element będzie wyglądał naturalnie, gdy podczas zaznaczania użyjesz opcji **zmiękczenia krawędzi i wygladzania**. Wartości tych parametrów można regulować. Warto wypróbować ich działanie.
- s. 183 • Warstw można także używać do **renowacji starych zdjęć**.



25. Własny film, czyli jak twórczo wykorzystać kamerę

NA TEJ LEKCJI:

- poznasz podstawy filmowania;
- dowiesz się, jak układać scenariusze do filmów.

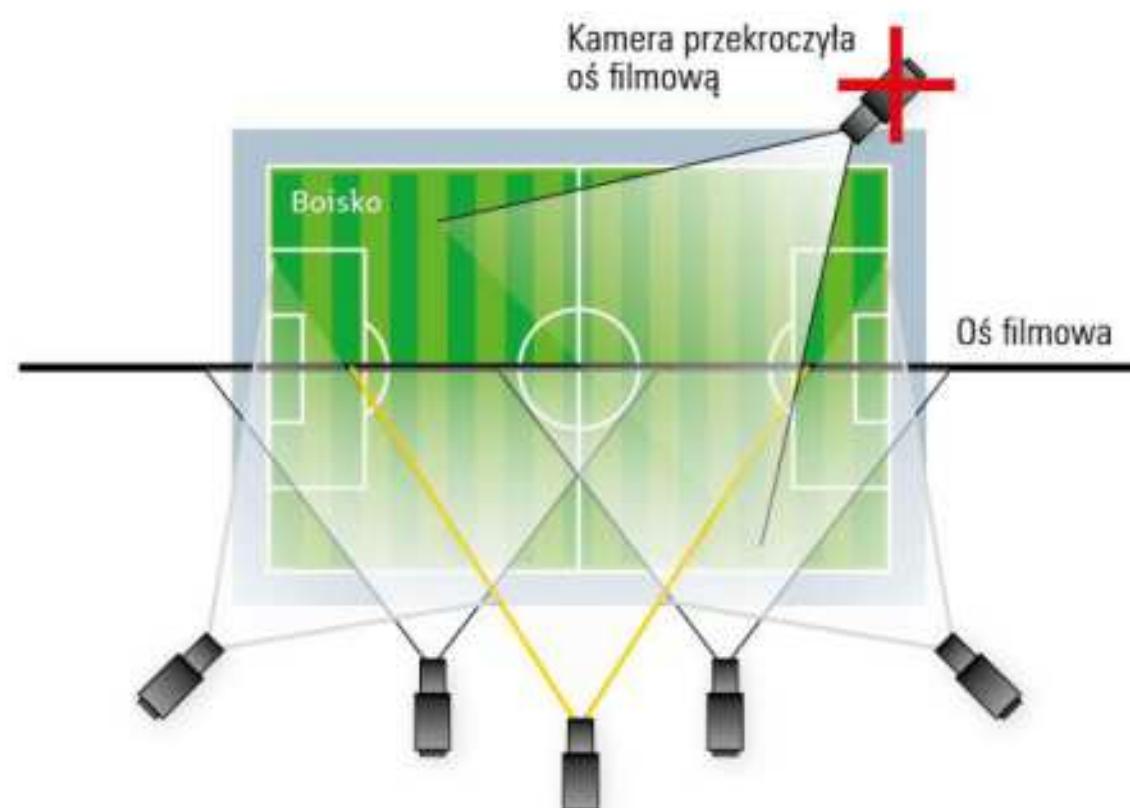
Fotografia przeszła wielką ewolucję od czasów szklanych negatywów do aparatów cyfrowych. Podobną ewolucję przeszły kamery filmowe i technika tworzenia ruchomych obrazów. Każdy smartfon może stać się narzędziem dla reportera lub filmowca amatora. Filmowałeś/filmowałaś swoim aparatem już nie raz, ale czy wiesz, dlaczego niektóre twoje filmy nie udały się tak, jak zaplanowałeś/zaplanowałaś? Podobnie jak w fotografii, także w filmie obowiązują pewne zasady. Warto je poznać, by później móc twórczo ich używać.

25.1. Ustawienia kamer

Rejestracja i późniejszy montaż ruchomych obrazów oprócz zachowania zasad kadrowania wymaga także dbałości o inne szczegóły techniczne. Jednym z nich jest zachowanie osi filmowej. Jeśli ją zignorujesz, widzowie mogą stracić orientację w przestrzeni twojego filmu. Przykładowo mogą nie zorientować się, kto kogo goni, kto na kogo patrzy lub z kim rozmawia. Wszystko wyjaśni przykład.

Wielokrotnie oglądałeś/oglądałaś relację z meczów piłkarskich. Ustawienie kamer nie jest przypadkowe.

zasada
nieprzekraczania
osi filmowej



Rys. 25.1. Przykład rozstawienia kamer na meczu piłkarskim

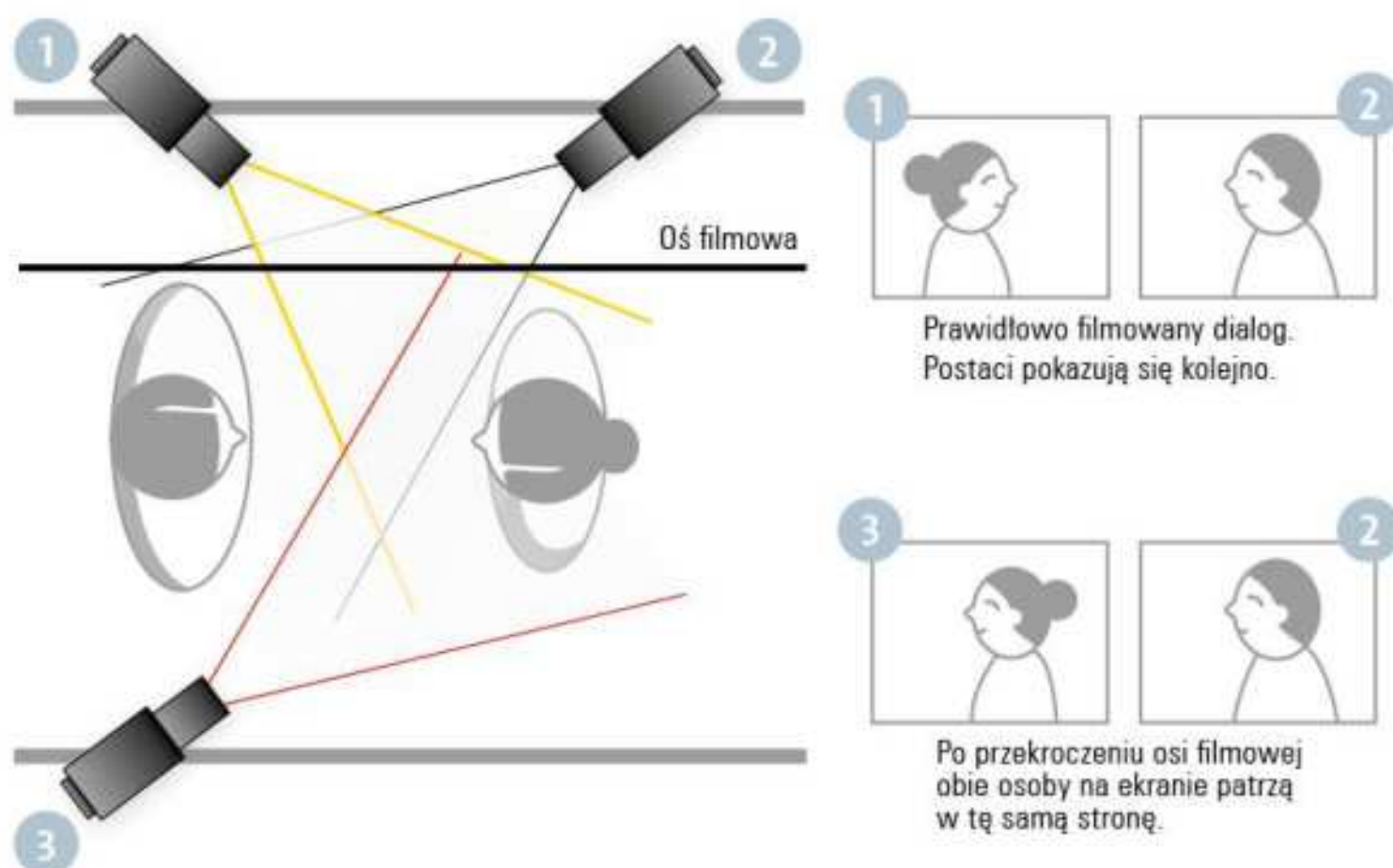


Widz wie, w którą stronę atakuje dana drużyna. Jeśli zostanie zachowana zasada nieprzecinania osi filmowej, z punktu widzenia widza kierunek ataku dla danej drużyny jest stały bez względu na to, z której kamery ogląda obraz. Co stałoby się, gdyby oś została przekroczona (rys. 25.1.)? Obserwator odniósłby wrażenie, że obie drużyny strzelają do tej samej bramki.

Uwaga!

Obecnie prowadzi się transmisję za pomocą bardzo wielu kamer i niektóre są ustawione po przeciwnej stronie osi. Pokazują one tylko detale, np. nogi zawodnika kopiącego piłkę czy wyrzucającego ją z autu. Widz nie traci przez to orientacji.

Tę samo zasadę stosuje się przy filmowaniu dialogu dwóch osób (rys. 25.2.). Jeśli rozmowę pokazują na zmianę kamery 1 i 2, widz wie, z której strony stoi dana osoba, i prawidłowo odczytuje scenę – rozmówcy stoją naprzeciwko siebie i rozmawiają. Jeśli tę samą rozmowę pokazywałaby kamera 2 i 3, widz miałby wrażenie, że obie osoby rozmawiają z kimś trzecim, patrząc w tę samą stronę, w tym przypadku w lewo.



Rys. 25.2. Oś filmowa podczas rejestracji sceny rozmowy dwóch osób

Można podać jeszcze inne takie przykłady – chociażby sceny pościgów, w których wyraźnie musi być widać, kto ściga, a kto ucieka. Przy złamaniu osi widz odniesie wrażenie, że biegną naprzeciwko siebie.



Podczas prowadzenia kamery i kadrowania oczywiście obowiązują także zasady dobrej fotografii. Są one często naginane przez wybitnych twórców dla uzyskania zaplanowanego efektu artystycznego lub narracyjnego. Jednak początkujący filmowcy powinni przestrzegać zasad, co pozwoli im uniknąć błędów.

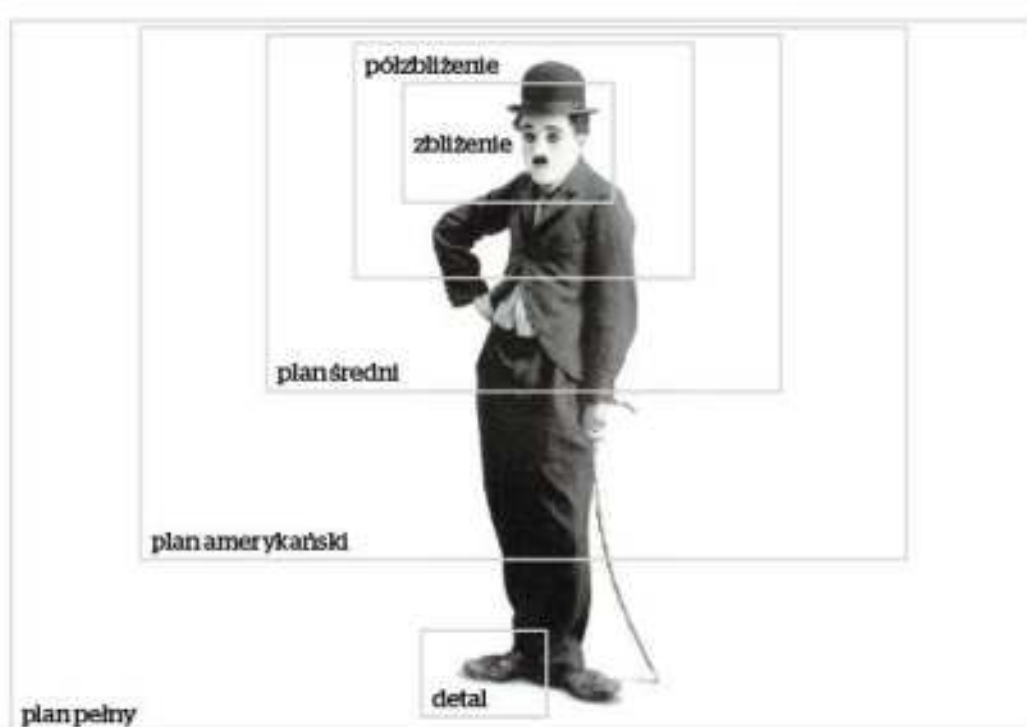
25.2. Własny tutorial, czyli filmujemy nie tylko kamerą

Zapewne często korzystasz z filmowych tutoriali zamieszczanych w sieci przez zawodowców i amatorów. Jak one powstają? Co jest potrzebne do ich zrealizowania? Okazuje się, że technicznie niewiele i najważniejszy okazuje się jak zwykle scenariusz. Nawet największe środki przeznaczone na film nie pomogą stworzyć dzieła na podstawie nieciekawego scenariusza.

scenariusz

Prace nad własnym tutorialiem rozpocznij więc od wymyślenia scenariusza. W przykładzie zajmijmy się prezentacją planów filmowych. Ich nazwy wchodziły w skład podstawowych pojęć, jakimi posługują się filmowcy (rys. 25.3.).

Oprócz planów z rysunku 25.3. istnieją jeszcze dwa szersze. W planie ogólnym widać okolicę, w której dzieje się akcja, a bohater może być rozpoznawalny, jednak nie dominuje. Plan totalny jest bardzo szeroki, a bohater praktycznie niewidoczny. Oba są używane do pokazania widzowi miejsca akcji.



Rys. 25.3. Podstawowe plany filmowe związane z bohaterem



Rys. 25.4. Plan ogólny



Rys. 25.5. Plan totalny



Klasyczny scenariusz składa się z trzech części.

Początek, inaczej zwany w filmie **ekspozycją**, prezentuje tło i czas akcji oraz postaci.

Rozwinięcie akcji – tu rozgrywa się fabuła, na przykład perypetie bohaterów.

Zakończenie jest rozwiązaniem akcji. Może być jednoznaczne lub pozostawiać widzowi swobodną interpretację. Jednak powinno doprowadzić do końca najważniejsze wątki.

nagrywanie scen

Czy wszystkie części mogą wystąpić w scenariuszu samouczka? Wróćmy do przykładu. Jeśli jeszcze nie znasz nazw planów filmowych, będzie on dla ciebie ważny w kontekście tworzenia własnego filmu.

AUTOR – NARRATOR PRZY BIURKU W TYPOWYM, UCZNIOWSKIM POKOJU.

AUTOR

Cześć, dzisiejszy odcinek poradnika filmowca amatora poświęcimy planom filmowym. I nie chodzi tu o plany związane z produkcją, a o sposób pokazywania obiektów na ekranie. Czasami zdarza się wam mówić do operatora „Bliżej. Jeszcze. Za blisko.”? Koniec z tym. Dziś poznacie pojęcia, które jednoznacznie określą plan ujęcia.

Na ekranie szybko zmieniające się po kolei zdjęcia różnych planów filmowych tej samej sceny. Od najszerszych do najbliższych.

AUTOR

Zdjęcia przedstawiają różne plany filmowe. Najszerszy pokazuje miejsce akcji, a najbliższy detal postaci. Zaczniemy od najszerszego.

Na ekranie zdjęcie planu ogólnego.

AUTOR

Taki plan nazywamy ogólnym. Widać na nim miejsce akcji, otoczenie bohatera. Bohater nie jest głównym obiektem na ekranie. Można jednak określić relacje pomiędzy nim a otoczeniem.

Na ekranie zdjęcie planu pełnego.

AUTOR

A to plan pełny. Widać całą postać lub obiekt.

Na ekranie zdjęcie planu amerykańskiego.

AUTOR

W planie amerykańskim postać jest pokazywana od głowy do kolan.

Na ekranie zdjęcie planu średniego.

AUTOR

Reporterów, dziennikarzy wiadomości, bohaterów w scenie, w której ważny jest ruch ciała, np. ręk, pokazuje się w planie średnim, czyli od pasa w górę.



Na ekranie zdjęcie planu półzblizenie.

AUTOR

To jest półzblizenie. W tym planie widać już emocje bohatera. Zaczynają być ważne umiejętności aktora.

Na ekranie zdjęcie planu zblizenie.

AUTOR

Zblizenie pokazuje obiekt lub aktora z niewielkiej odległości. Stawia wysokie wymagania przed aktorami.

Na ekranie zdjęcie planu detal.

AUTOR

Detal jest najbliższym planem. Za jego pomocą zwracamy uwagę na drobne elementy.

Zdjęcia zebrane na jednym ekranie.

AUTOR

Totalny widok na wszystkie plany. Dlaczego totalny? To nazwa najszerzego planu, pokazującego miejsce akcji, np. miasto w ujęciu z drona.

Na ekranie narrator przy biurku.

AUTOR

I to już wszystko. Od dziś, jeśli powiecie kamerzyście: teraz plan amerykański, będzie wiedział, o co chodzi. Oczywiście pod warunkiem, że też obejrzał mój film. Do zobaczenia w następnym odcinku samouczków dla amatorów filmowania.



Rys. 25.6. Przykładowe zdjęcia do filmu o planach filmowych

Jako ilustracji możesz użyć kolejno kadrowanego zdjęcia (rys. 25.6.).

Mamy więc **wstęp**, w którym autor wyjaśnia, o czym będzie film, **rozwińnięcie**, czyli omówienie planów, i **zakończenie**.

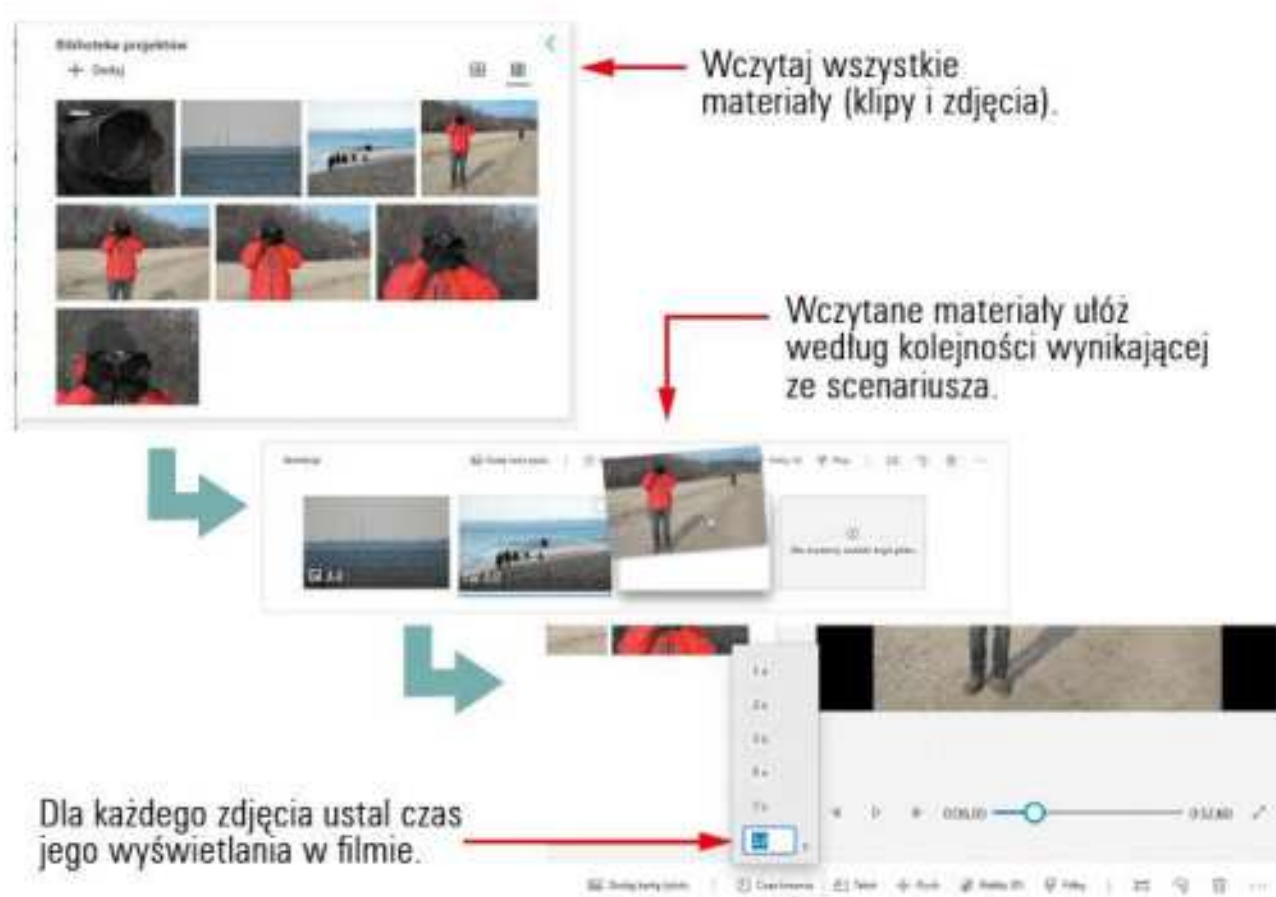


Do zrealizowania tego filmiku potrzeba jeszcze oprogramowania do montażu, kamery i mikrofonu. Ponieważ film jest przeznaczony do internetu, a oglądany będzie na telefonach lub ekranach komputerów, wystarczą najprostsze narzędzia.

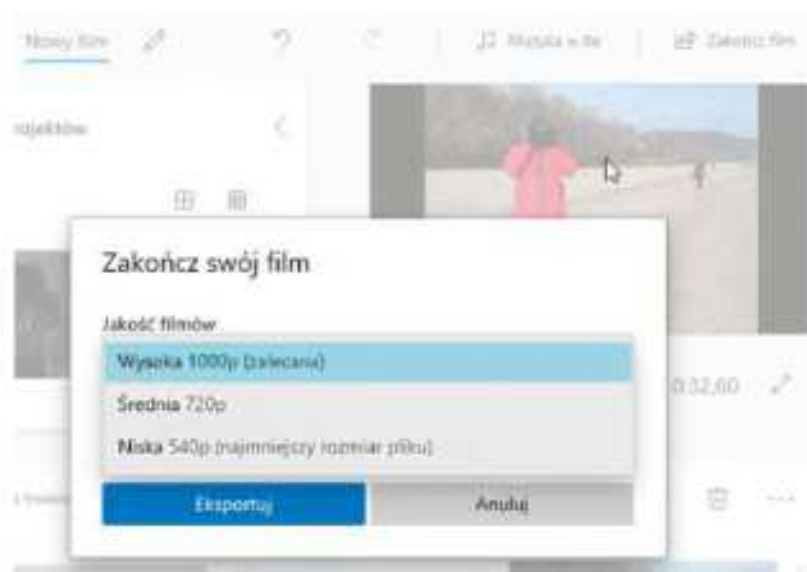
Dla systemu Windows 7 i 8 ogólnodostępny program do montażu to Microsoft MovieMaker. Obecnie nie jest on już dostępny w sklepie Microsoft, a jego pobranie z innych źródeł grozi zainfekowaniem komputera. W systemie Windows 10 użyj aplikacji **Zdjęcia**. Czas więc wziąć się do pracy.

Ustaw kamerę w taki sposób, by kadr odpowiadał założeniom scenariusza. Zarejestruj wszystkie sceny, w których występuje autor, a następnie dźwięk do podłożenia pod zdjęcia.

dźwięk filmu



Rys. 25.7. Proces szybkiego tworzenia filmu w MovieMaker



Rys. 25.8. Opcje MovieMakera do zapisu filmu w pliku

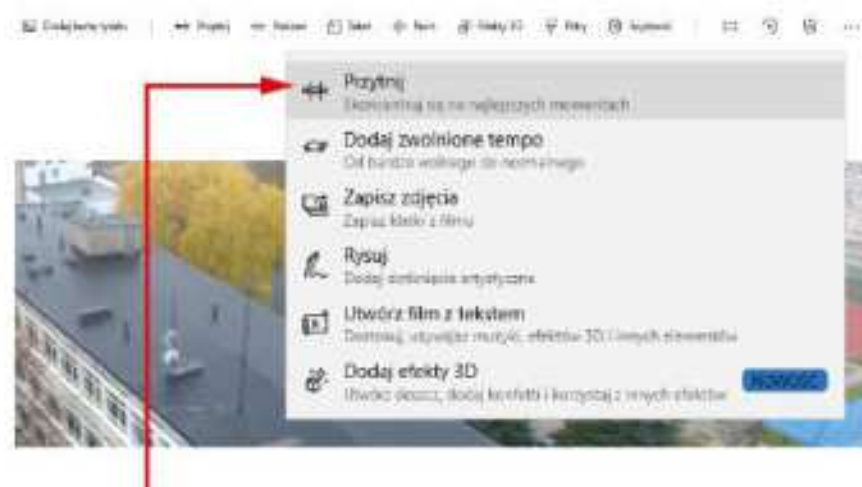
Gdy film będzie gotowy, dodaj do niego napisy. Nie zapomnij wymienić autorów zdjęć i nazwisk uczestników produkcji.

Pozostaje już tylko zapisać film w postaci pliku (wyrederować; rys. 25.8.) lub od razu upublicznić w portalach społecznościowych (rys. 25.9).



Rys. 25.9. Dodawanie efektów specjalnych do filmu w programie Zdjęcia

Aplikacja Zdjęcia (dostępna w Windows 10) jest również bardzo łatwa w obsłudze i intuicyjna (rys. 25.10.).



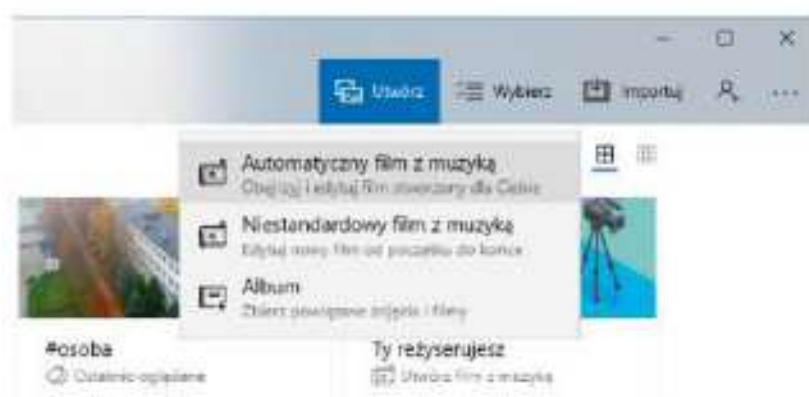
Po wczytaniu klipu możesz wybrać jedną z opcji. Jeśli chcesz przyciąć film, czyli wyciąć pewien fragment, użyj opcji **Przytnij**.



Rys. 25.10. Edycja filmu w aplikacji Zdjęcia (prycinanie klipu)

Uwaga!

Opcja **Zapisz projekt** nie eksportuje filmu w postaci gotowej do obejrzenia w odtwarzaczu. To plik, w którym program zapisuje dane potrzebne do odtworzenia projektu, czyli nazwy i miejsce pobrania klipów lub zdjęć, czas ich trwania, miejsca cięć itp. Film zostanie zapisany w pliku dopiero po użyciu opcji **Zapisz film** lub **Publikuj film**.



Rys. 25.11. Opcje szybkiego tworzenia filmów w aplikacji Zdjęcia

Aplikacja ta umożliwia dodawanie efektów 3D, np. fajerwerków, baloników i innych animacji nakładanych na film. Jeśli chcesz szybko zrobić film ze zdjęć i klipów (tak jak w ćwiczeniu z MovieMaker), użyj jednej z opcji menu **Utwórz** (rys. 25.11.).

Zaawansowani amatorzy używają rozbudowanych edytorów. Są wśród nich także niekomercyjne, darmowe aplikacje,

jak **Nuke** – edytor w wersji komercyjnej używany przez znanych producentów filmowych oraz **DaVinci Resolve** – program znanego producenta sprzętu studyjnego.

YouTube także udostępnia ciekawy i intuicyjny w obsłudze program do edycji filmu. Możesz w nim przycinać sceny, dodawać dźwięki, darmową muzykę, napisy itp. Program działa po przesłaniu klipów do chmury YouTube.

Jeśli udało ci się stworzyć ciekawy film, upublicznij go. Może stanie się przebojem internetu?

Istnieją także programy do rejestracji ekranu lub poszczególnych okien. Dzięki nim można tworzyć samouczki, na przykład o obsłudze któregoś z programów. Przykładem może być **Debut Video Capture Software** (rys. 25.12.). Może on także pobierać i rejestrować obraz ze źródeł zewnętrznych – kamery internetowej, sieciowej, kamery analogowej lub kamery USB.



Rys. 25.12. Okno programu Debut Video Capture Software



ZADANIA DO ROZWIĄZANIA

1. W czasie oglądania filmu lub programu TV zwróć uwagę na pracę kamer. Analizuj kadrowanie, czas trwania ujęć, miejsce ustawienia kamery itp. Policz, ile kamer filmowało sceny. Wynotuj wszystkie spostrzeżenia i omów krótko zasadność użycia takich środków. Przedstaw własny pomysł na daną scenę.
2. Wyszukaj w sieci informacje o darmowych programach służących do rejestracji obrazu z ekranu komputera. Sprawdź działanie dwóch wybranych na podstawie opinii użytkowników lub testów w portalach branżowych. Opisz ich najciekawsze funkcje, np. efekt kliknięcia myszką, powiększanie.
- 3*. Ułóż scenariusz tutoriala (samouczka) wideo opisującego zakładanie konta pocztowego w dowolnej, darmowej usłudze. Zrealizuj swój pomysł za pomocą programu do rejestracji ekranu komputerowego lub dowolnej kamery. Zmontuj i upublicznij swój film.
4. Odnajdź we własnych zasobach zdjęcia ze szkolnej wycieczki lub uroczystości. Ułóż scenariusz filmu, na który będą się one składały. Zanotuj, jak długo wybrane zdjęcia mają być widoczne na ekranie. Ułóż komentarze lektora. Za pomocą programu MovieMaker lub dowolnego innego edytora wideo zrealizuj ten film-prezentację. Postaraj się ułożyć zdjęcia w ciekawej kolejności. Zorganizuj pokaz. Pamiętaj, że upublicznienie filmu może odbyć się jedynie z poszanowaniem praw do wizerunku osób ze zdjęć.
5. Sprawdź działanie twojej kamery w różnych warunkach oświetleniowych – od zaciemnionego pomieszczenia do pełnego światła w słoneczny dzień. Oceń subiektywnie zarejestrowane obrazy. Zwróć uwagę na szумы, wyrazistość, kontrast i barwy. Oceń, przy jakim najmniejszym oświetleniu twoja kamera rejestruje akceptowalny co do jakości obraz. Użyj programów tematycznych. Zanotuj swoje spostrzeżenia. Będą przydatne podczas rejestracji ujęć filmu.

PODSUMOWANIE LEKCJI

- Zachowuj zasadę **nieprzekraczania osi filmowej**. s. 185
- Nie zaczynaj pracy nad filmem, dopóki nie skończysz układać **scenariusza**. s. 187
- **Sceny** z aktorami rejestruj kilka razy. Za każdym razem staraj się uniknąć zrobionych wcześniej błędów. s. 188
- Mów naturalnie, ale głośno i wyraźnie, by mikrofon mógł zarejestrować dźwięk na odpowiednim poziomie. Wykonaj **próby dźwięku** dla wszystkich kamer i aktorów w celu wyrównania ich poziomu głośności. Będzie mniej pracy przy montażu. s. 190
- **Do rejestracji i montażu** możesz użyć dowolnego programu zgodnie z licencją. Na początku korzystaj z mało rozbudowanych aplikacji. s. 190

26. Klatki do komputera, czyli jak transmitować strumieniowo

NA TEJ LEKCJI:

- dowiesz się, do czego służy strumieniowanie i czym jest transmisja strumieniowa;
- poznasz narzędzia służące do uruchomienia i odbierania transmisji strumieniowej i ich użyjesz.

Podłączenie aparatu do komputera lub odczytanie zdjęć lub filmu z karty pamięci nie jest trudne. Umiesz już to zrobić. Jednak kamery mogą przysyłać obraz także na żywo do komputera, sieci lokalnej lub internetu. Można je wtedy podłączyć na kilka sposobów, wykorzystując łącza kablowe lub bezprzewodowe. Tak podłączona kamera może spełniać wiele funkcji, z których większość już znasz. Współczesna telewizja odbierana za pośrednictwem anteny działa na podobnej zasadzie. Strumień danych płynie z nadajnika i jest przetwarzany na obraz i dźwięk w odbiorniku.

26.1. Strumienie, czyli jak kamera przesyła obraz

strumieniowanie

Czy wiesz, jak to się dzieje, że obraz z kamery można obserwować na ekranie komputera oddalonego o wiele kilometrów? Projektanci takich urządzeń użyli znanych rozwiązań – postanowili przysyłać dane o obrazie i dźwięku strumieniowo.

Strumień danych przesyłany do komputera, sieci lub zapisywany na nośniku.



Obiektyw skupia światło na matrycy światłoczułej.

Dane o poziomie naświetlenia każdego punktu odczytuje układ elektroniczny.

Kamerę tworzą trzy zasadnicze części: obiektyw, matryca światłoczuła i układ elektroniczny. Wpadające przez obiektyw światło dociera do matrycy, gdzie punkt po punkcie jest odczytywany stopień naświetlenia. Te dane pobiera sterujący matrycą układ elektroniczny. Po zebraniu danych z całej matrycy powstaje mapa bitowa klatki, czyli zdjęcie albo pojedyncza klatka filmu. Aby przestać

Rys. 26.1. Uproszczony schemat kamery cyfrowej



je do komputera, monitora lub sieci, należy postąpić podobnie, czyli dane każdego punktu klatki z pamięci kamery są przesyłane po kolei – tworzą strumień. W urządzeniu wyświetlającym punkty zostają ułożone z zachowaniem miejsca z oryginału. Jeśli obraz ma być jedynie przeglądany, na przykład gdy oglądasz film z portalu kinowego, na tym proces się kończy. Jeśli jednak zdjęcie lub film ma być zachowany, komputer musi zapisać dane ze strumienia w pliku (rys. 26.1.).

strumień danych

Przesłanie strumienia do komputera może nastąpić przewodowo lub bezprzewodowo. Najczęściej używanym złączem kablowym jest magistrala USB, w którą wyposażono każdy współczesny komputer. Połączenie bezprzewodowe może odbywać się za pośrednictwem karty sieciowej WiFi lub interfejsu Bluetooth.

Pobieranie plików z kamery przebiega standardowo, ponieważ system jej pamięć traktuje tak samo, jak inne pamięci masowe, na przykład dyski, karty SD, płyty, pendrive'y.

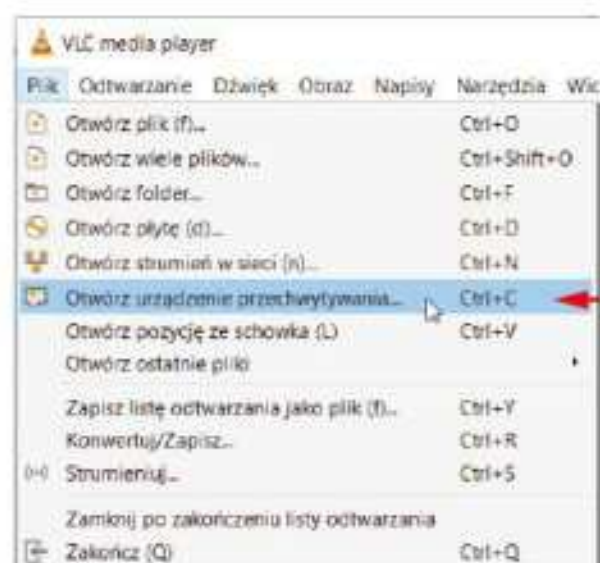
Przesyłanie strumienia z kamery do komputera wymaga użycia odpowiedniego programu lub opcji aplikacji, przykładowo komunikatora. Taki program może rejestrować obraz w pliku lub przesyłać go dalej, chociażby do internetu.

26.2. Na dysk, czyli jak zarejestrować obraz

Aparat fotograficzny lub kamera (np. w smartfonie) może rejestrować zdjęcia i filmy w pamięci i zapisywać je w pliku. Zdarza się jednak, że zachodzi potrzeba rejestracji obrazów z kamer pozbawionych takiej możliwości, na przykład kamer USB (także tych zainstalowanych w komputerach przenośnych). W takiej sytuacji należy użyć odpowiedniego programu – najlepiej skorzystać z aplikacji dołączonej przez producenta kamery lub z darmowych programów, jak VLC. Ciekawe możliwości VLC wykorzystamy w dalszej części rozdziału.

transmisja na żywo

Jeśli do komputera dołączamy kamerę USB lub inne źródło obrazu i dźwięku (rys. 26.2.), musimy zadbać o prawidłowe zainstalowanie sterowników. Umożliwi to programowi odnalezienie źródła i wykorzystanie jego możliwości.



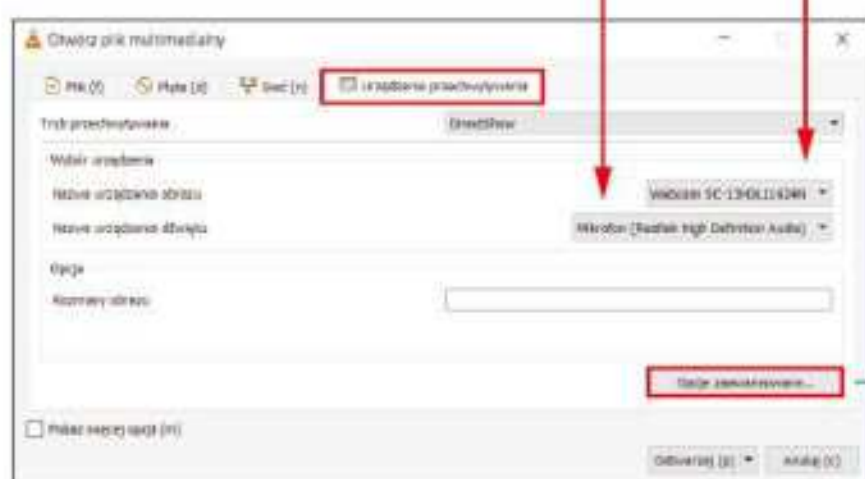
Wybierz tę opcję lub wciśnij **Ctrl + C**, aby wyświetlić i zarejestrować obraz z dźwiękiem z kamery USB

Rys. 26.2. Opcje menu Plik programu VLC

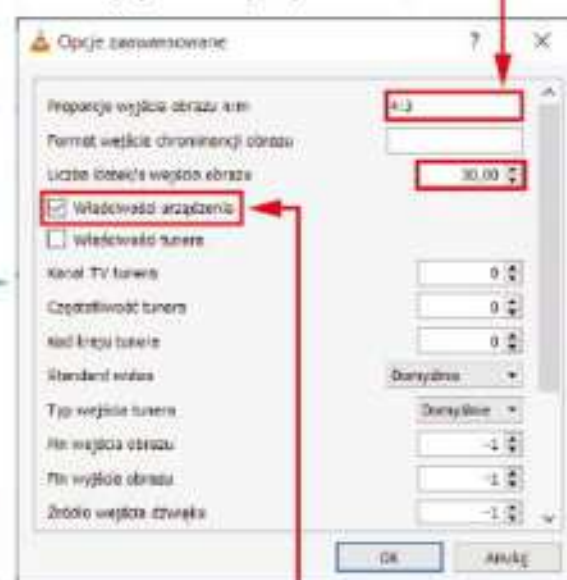
Rejestrację z kamery lub innego urządzenia podłączonego do USB należy zacząć od wybrania odpowiedniego źródła (rys. 26.2.). Zauważ, że VLC (podobnie jak wiele innych odtwarzaczy) może odtwarzać multimedia z wielu źródeł.

Jeśli do USB dołączono więcej źródeł, wybierz jedno z nich. W przeciwnym przypadku zostanie wybrane domyślne.

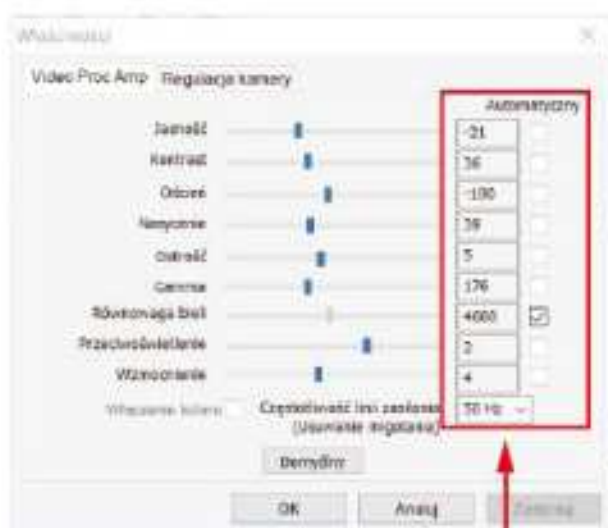
Źródło dźwięku nie musi pochodzić z kamery.



W opcjach zaawansowanych możesz ustawić m.in. proporcje obrazu, np. 4:3 lub 16:9, oraz liczbę klatek na sekundę dla twojej kamery, np. 30 kl./s.

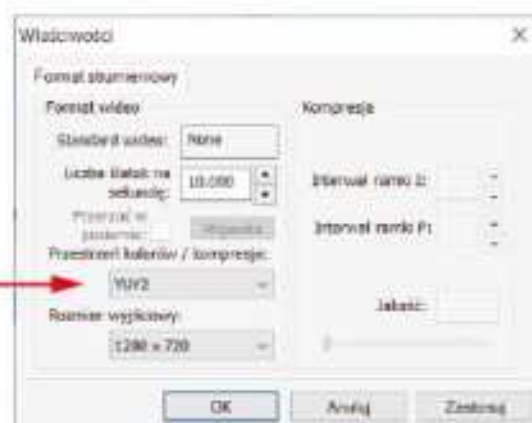


Jeśli nie znasz parametrów urządzenia wejściowego, np. kamery, wybierz opcję **Właściwości urządzenia**. Część ustawień zostanie wpisana zgodnie z jego właściwościami.



Dostosowanie parametrów obrazu. Możesz zaznaczyć opcję automatycznego ustawienia dla każdego z parametrów.

W kolejnym kroku ustawiania opcji zaawansowanych możesz ustawić parametry formatu strumieniowego. Z nimi będzie rejestrowany obraz i dźwięk.



VLC jest gotowy do wyświetlenia i rejestrowania obrazu ze wskazanego źródła.



Rys. 26.3. Proces przygotowania programu VLC do rejestrowania obrazu i dźwięku z zewnętrznego źródła



Takie podłączenie kamery i ustawienie opcji służy do wyświetlania obrazu na dużym ekranie (np. za pomocą projektora) podłączonym do komputera. Można to wykorzystać w pracowni fizyki lub chemii do prezentacji przebiegu doświadczenia. Zdarza się, że z różnych przyczyn takie doświadczenia wykonuje się tylko raz, ponieważ obiekt w jego trakcie ulega zniszczeniu. Warto więc je zarejestrować. Zrobisz to za pomocą innej opcji VLC (rys. 26.4.). Plik z nagraniem trafi do folderu domyślnego dla filmów w Windows.

Ten komputer/Wideo

VLC zapisuje go w formacie AVI. Pliki zajmują więc sporo miejsca na dysku. Oczywiście można je przekonwertować na inny format, na przykład mp4. W tym także pomoże VLC.



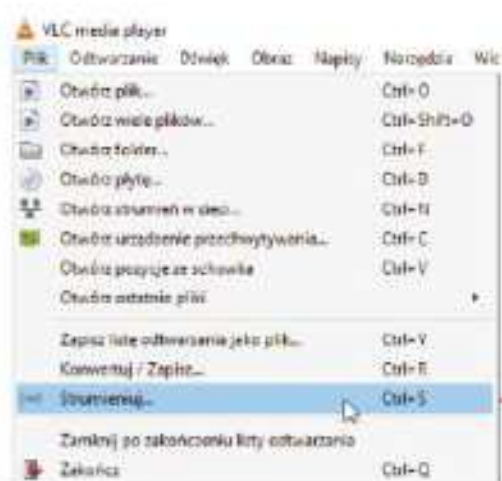
Rys. 26.4. Opcje odtwarzania i rejestrowania obrazu i dźwięku w VLC

26.3. Nadajemy, czyli jak uruchomić transmisję w sieci

Obraz i dźwięk można transmitować i nagrywać wprost z kamery USB za pomocą VLC lub innego programu o podobnych właściwościach (rys. 26.5.). Przykładem zastosowania transmisji w sieci lokalnej może być relacja na żywo z ważnej uroczystości szkolnej lub emisja filmu w salach lekcyjnych odbywająca się w tym samym czasie.

Uwaga!

Ćwiczenie możesz wykonać w pracowni, prowadząc transmisję pomiędzy komputerami, lub wykorzystaj do tego swój komputer z dwiema sesjami programu VLC.



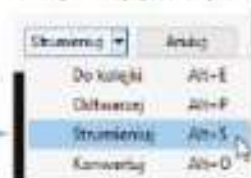
Opcja **Strumienia** daje możliwość wysyłania do sieci obrazu i dźwięku ze wskazanego źródła.



W zakładkach można wybrać źródło strumienia.

Jeśli transmisja ma pochodzić z kamery USB, wybierz **Urządzenie przechwytywania**.

Start wysyłania strumienia do sieci nastąpi po wciśnięciu przycisku. Możesz także wybrać inną akcję dotyczącą danych ze źródła.



Określenie celu to w rzeczywistości wskazanie sposobu (protokołu) transmisji. Jeśli transmisja ma być odbierana za pośrednictwem przeglądarki, wybierz **HTTP**.

Rys. 26.5. Opcja Strumienia w programie VLC

Jeśli chcesz swoją transmisję oznakować logo, jak na materiałach komercyjnych stacji telewizyjnych, wybierz opcję **Dostosowanie i efekty** (rys. 26.6.).

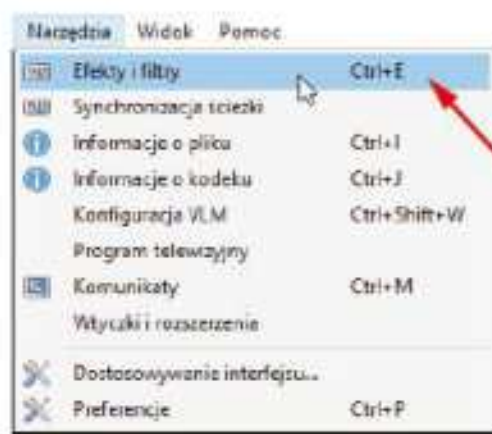


Jeśli chcesz wstawić logo do pliku z filmem, wykonaj czynności z rysunku 26.6. i użyj opcji **Strumieniowania do pliku**, wybierając opcję **Plik** w oknie konfiguracji celu (rys. 26.4.).

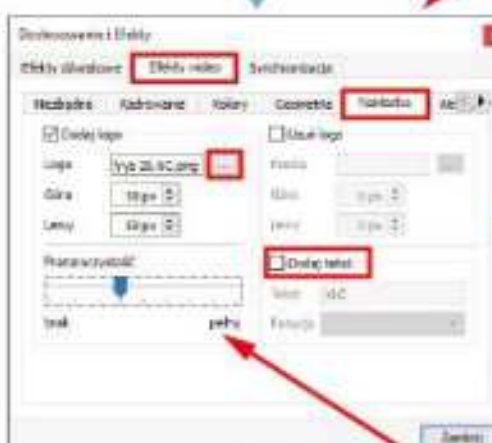
Odbieranie transmisji – w salach lekcyjnych lub w trakcie ćwiczenia na komputerach w pracowni – może odbywać się na kilka sposobów. Jeśli została wybrana transmisja HTTP, wystarczy uruchomić przeglądarkę i wpisać adres IP komputera strumieniującego w sieci oraz numer portu (domyślnie 8080). Przykładowa postać pełnego adresu wygląda następująco:

`http://192.168.1.120:8080`

W przypadku wysyłania strumienia za pośrednictwem innego protokołu po stronie odbiorczej będzie potrzebny program VLC lub inny odtwarzacz wyposażony w odpowiednią opcję.



Z menu **Narzędzia** wybierz **Efekty i filtry**, a następnie zakładkę **Efekty wideo** oraz **Nakładka**.



Zaznacz opcję **Dodaj logo** i ustaw ścieżkę dostępu do pliku z grafiką. Możesz również dodać tekst, który będzie nałożony na obraz. Jeśli twoje logo nie ma kształtu prostokąta, użyj grafiki z przezroczystym tłem (kanał alfa) zapisanej w pliku PNG.



Ustal położenie logo względem narożnika ekranu oraz poziom jego przezroczystości.

Fragment kadru filmu z nałożonym przezroczystym logo szkoły.

Rys. 26.6. Nakładanie logo na strumieniowany obraz

26.4. Do portalu, czyli nadajemy z komórki

Zapewne transmisje strumieniowe znasz z portali społecznościowych i informacyjnych. Każdy, kto ma smartfon podłączony do internetu, może nadawać własne relacje na żywo. Również w szkole przydałaby się taka relacja – transmisja z jakiejś uroczystości czy debaty wyborczej do samorządu uczniowskiego. W salach lekcyjnych lub indywidualnie na ekranach telefonów każdy zainteresowany mógłby obejrzeć transmisję. Można także zorganizować transmisję ważnej lekcji dla chorego ucznia, ale to jednak wymaga zgody nauczyciela. Do uruchomienia transmisji potrzebujesz na przykład aplikacji Facebook i połączenia z internetem. Transmisje strumieniowe są bardzo przydatne w różnych okolicznościach. Pamiętaj jednak, że twoi odbiorcy muszą o nich wiedzieć. Przed relacją powiadom o jej terminie jak największą liczbę znajomych i poproś ich, by rozesłali tę informację wśród innych osób. Gdy rozpoczniesz transmisję,

transmisja
strumieniowa



twoi zalogowani do portalu społecznościowego znajomi zostaną o tym powiadomieni i mogą obserwować twój przekaz.

Uwaga!

Do transmisji strumieniowej za pośrednictwem telefonu komórkowego można wykorzystać różne aplikacje. Podczas ich doboru zwróć uwagę na bezpieczeństwo. Zanim je zainstalujesz, zbierz informacje i opinie na ich temat.

Podczas prowadzenia transmisji na żywo, pamiętaj o poszanowaniu prawa do wizerunku osób pokazywanych w trakcie strumieniowania.

ZADANIA DO ROZWIĄZANIA

1. Odnajdź w sieci darmowe odpowiedniki programu VLC. Na podstawie opinii i opisów ich opcji oceń ich przydatność do transmisji strumieniowych. Porównaj ich możliwości z VLC.
2. Przeprowadź transmisję dowolnego pliku zawierającego film z dźwiękiem w sieci lokalnej pracowni komputerowej. Zwróć uwagę na odpowiednie dobranie portu. Zaproponuj odebranie transmisji koleżankom i kolegom z klasy. Zwróć uwagę na odpowiednie wybranie źródła. Podaj odbiorcom dane do odbioru transmisji i wytłumacz ich znaczenie.
3. Dowolny film, np. z wycieczki klasowej, opatrz swoim zdjęciem. Użyj odpowiednio przygotowanego swojego selfie. Dla lepszego efektu użyj kanału alfa dla przezroczystego tła. Zwróć uwagę na odpowiednie umiejscowienie logo. Zapisz strumień w pliku.
4. Przeprowadź transmisję na żywo, posługując się kamerką USB lub wbudowaną w komputer. Zwróć uwagę na odpowiednie wybranie źródła. Podaj odbiorcom dane do odbioru transmisji i wytłumacz ich znaczenie.
5. Wyszukaj w sieci informacje na temat strumieniowania w sieci dźwięków za pośrednictwem darmowego serwera radia internetowego. Zapoznaj się z możliwościami i zaletami takiego rozwiązania. Zastanów się nad tym, czego potrzebujesz do uruchomienia własnego radia internetowego. Pamiętaj o poszanowaniu praw autorskich.

PODSUMOWANIE LEKCJI

- s. 194 • **Strumieniowanie** pozwala przysyłać dane obrazu lub dźwięku za pośrednictwem sieci w taki sposób, by można je było odtwarzać w tym samym czasie po stronie odbiorcy. Jest to podobne do działania tradycyjnej telewizji cyfrowej odbieranej za pośrednictwem anteny.
- s. 195 • Twój komputer może być zarówno nadajnikiem, jak i odbiornikiem **strumienia danych**. Strumienie mogą być nadawane w sieci lub zapisywane w pliku i odtwarzane po zakończeniu transmisji.
- s. 195 • Większość smartfonów umożliwia **transmitowanie na żywo** za pomocą wbudowanej kamery i mikrofonu.
- s. 199 • Do **transmisji strumieniowych** można użyć wielu różnych programów i aplikacji. Należy je wybierać po sprawdzeniu wiarygodności źródła i opinii o ich działaniu.



PODSUMOWANIE PRZED SPRAWDZIANEM

17. Modele w przestrzeni, czyli podstawy druku 3D

- Drukarka 3D nakłada kolejno na siebie warstwy roztopionego filamentu. s. 124
- Modele (obiekty) do drukowania w trzech wymiarach tworzy się za pomocą edytora trójwymiarowej grafiki wektorowej lub poprzez skanowanie 3D. s. 124
- Opracowano wiele formatów plików dla projektów 3D. Należy zwrócić uwagę na to, z którymi z nich dana drukarka jest kompatybilna. Najpopularniejsze z nich to STL i 3MF. s. 124

18. Modelujemy, czyli jak projektować obiekty 3D

- Są edytory 3D w chmurze. Przykładem może być Tinkercad. s. 129
- Pliki dla drukarki 3D w formacie STL można przeglądać w Paint 3D. s. 134

19. Wizualizacja pomysłów, czyli projektujemy dom w edytorze 3D

- Niektóre edytory, np. SketchUp, są połączone z mapami elektronicznymi. Dzięki temu można tworzyć modele obiektów architektonicznych i zaznaczać ich rzeczywiste lokalizacje. s. 138
- Obiekty 3D to nie tylko elementy do drukowania. Z modeli 3D są zbudowane m.in. gry komputerowe, a także animacje w filmach. s. 139
- Edytory 3D są wyposażone w szereg mechanizmów ułatwiających kreślenie brył i wypełnianie powierzchni. s. 141

20. Wypukłości nie tylko 3D, czyli tworzymy elementy graficzne publikacji

- Umiejętność wykorzystania warstw pozwala na stworzenie zaawansowanych projektów graficznych. Dzięki filtrom można przekształcać elementy graficzne. Wypróbuj ich działanie z różnymi ustawieniami parametrów. s. 147
- Efekt wypukłości elementów graficznych uzyskuje się dzięki odpowiedniemu użyciu filtrów i warstw. s. 152

21. Z wydruku do komputera, czyli skanujemy i odczytujemy dokumenty

- OCR – technologia zamiany skanowanych dokumentów na edytowalny tekst. s. 156
- CCD, CIS, TWAIN – pojęcia związane z działaniem skanera i jego użytkowaniem. s. 158

22. Dokumentujemy wydarzenia, czyli aktywna praca z aparatem fotograficznym

- Optymalne naświetlenie matrycy może być wynikiem różnych kombinacji czasu naświetlania i przysłony. s. 166
- Zdjęcia z aparatu możesz przysyłać do komputera za pośrednictwem łączy bezprzewodowych Bluetooth lub WiFi, kablem USB lub poprzez bezpośredni odczyt z wyjętej z urządzenia karty. s. 168
- Jeśli nie masz doświadczenia lub pomysłu na skadrowanie zdjęcia, skorzystaj z reguły trójpodziału. s. 170

23. Szturmować w chmurze, czyli poprawiamy zdjęcia w edytorze grafiki rastrowej

- Edytory grafiki rastrowej znajdziesz także w chmurach informatycznych. s. 172
- Większość czynności podczas obróbki zdjęć (np. retuszowanie, zmiana barw) możesz wykonać za pomocą niekomercyjnych edytorów grafiki rastrowej. s. 173

24. Tego tu nie było, czyli poprawiamy rzeczywistość na fotografiach

- Retusz i inne operacje na fotografiach najwygodniej wykonywać z wykorzystaniem mechanizmu warstw. Warstwy są jednym z najważniejszych mechanizmów tworzenia grafiki i montażu oraz retuszu zdjęć. s. 180
- Opcje narzędzi, np. wycinanie, pozwolą na uzyskanie odpowiednich efektów, np. dodania obiektu do zdjęcia. s. 181



25. Własny film, czyli jak twórczo wykorzystać kamerę

- s. 185 • Przestrzegaj zasad, np. nieprzekraczania osi filmowej i dobrego kadrowania.
- s. 187 • Zanim zaczniesz filmować, poznaj podstawy filmowania i ułóż scenariusz filmu.

26. Klatki do komputera, czyli jak transmitować strumieniowo

- s. 194 • Strumieniowanie pozwala przesyłać dane obrazu lub dźwięku za pośrednictwem sieci w taki sposób, by można je było odtwarzać w tym samym czasie po stronie odbiorcy.
- s. 195 • Twój telefon lub komputer mogą zarówno strumieniować media, jak i je odbierać.
- s. 199 • Strumieniowane dane można rejestrować w pliku.

V. Komputer w sieci

- ▶ Działanie internetu
- ▶ Łączenie sieci z internetem
- ▶ Topologia sieci komputerowych
- ▶ Odczytywanie danych z karty sieciowej
- ▶ Konfigurowanie połączenia
- ▶ Parametry urządzeń peryferyjnych





27. Nie wszystko jest takie oczywiste, czyli jak działa internet

NA TEJ LEKCJI:

- dowiesz się, jak działa internet oraz czym są protokoły komunikacyjne.

Internetem posługujesz się na co dzień. Wiesz, jak działa? Odpowiedź nie jest taka prosta, jakby mogło się wydawać. Aby zrozumieć procesy składające się na funkcjonowanie internetu, należy odpowiedzieć sobie na kilka pytań: Skąd biorą się dane w sieci i gdzie są składowane? Jak je odnajdujemy i w jaki sposób trafiają do komputera? W jaki sposób miliardy komputerów mogą funkcjonować jednocześnie w jednej sieci? W tym rozdziale się tego dowiesz.

27.1. Szkielet, czyli łączymy sieć z internetem

internet

Ułożenie prostej definicji internetu nie sprawi ci kłopotu. Można powiedzieć, że **internet** jest siecią łączącą mniejsze sieci (rys. 27.1.), by możliwa była wymiana danych pomiędzy tworzącymi je komputerami i innymi urządzeniami komputerowymi (kamery, dyski sieciowe itp.).

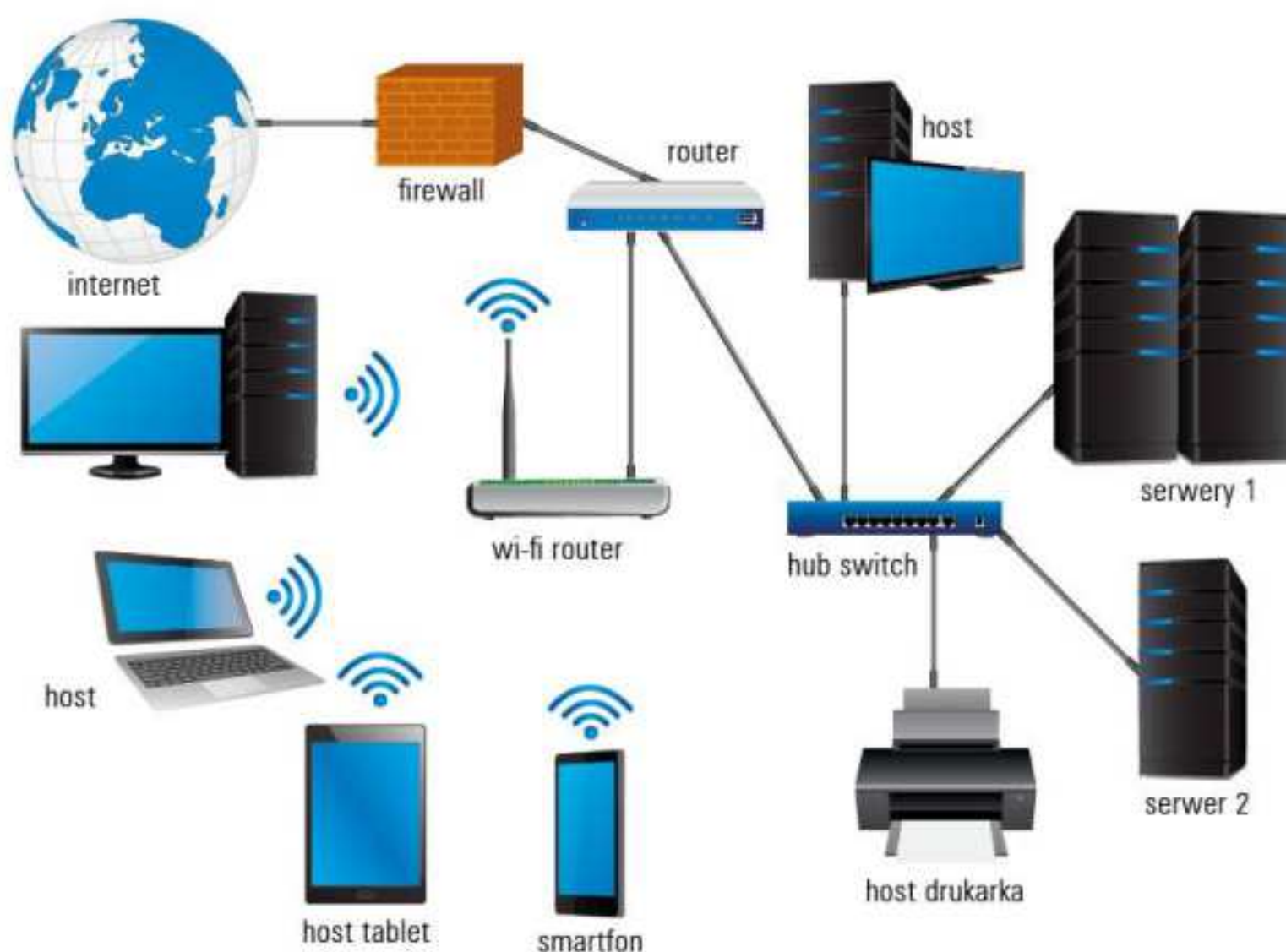


Rys. 27.1. Uproszczony schemat topologii internetu

Czym są sieci, które w uproszczeniu nazwaliśmy mniejszymi? Domyślasz się zapewne, że mogą to być sieci o zasięgu lokalnym, a więc na przykład obejmujące dzielnicę, wieś, budynek (biurowiec, dom mieszkalny itp.), biuro,



mieszkanie lub pokój. Skoro tak, to potrzebna jest infrastruktura i oprogramowanie. Małe sieci łączą swoje komputery (często nazywane klientami) za pośrednictwem **przełącznika (switcha)**. Przełączniki mogą być połączone do osobnego przełącznika, tworząc podsieci. W jakim celu? Po pierwsze – takie rozwiązanie pozwala oddzielić od siebie różne części sieci, którymi łatwiej zarządzać. Po drugie – w ten sposób zostaje uproszczona infrastruktura. Można także utworzyć podsieć, w której pracują serwery, co ułatwi zapewnienie im bezpieczeństwa. Przełącznik łączący podsieci może być połączony z **routerem**, za pośrednictwem którego cała sieć uzyska połączenie z internetem (rys. 27.2.).



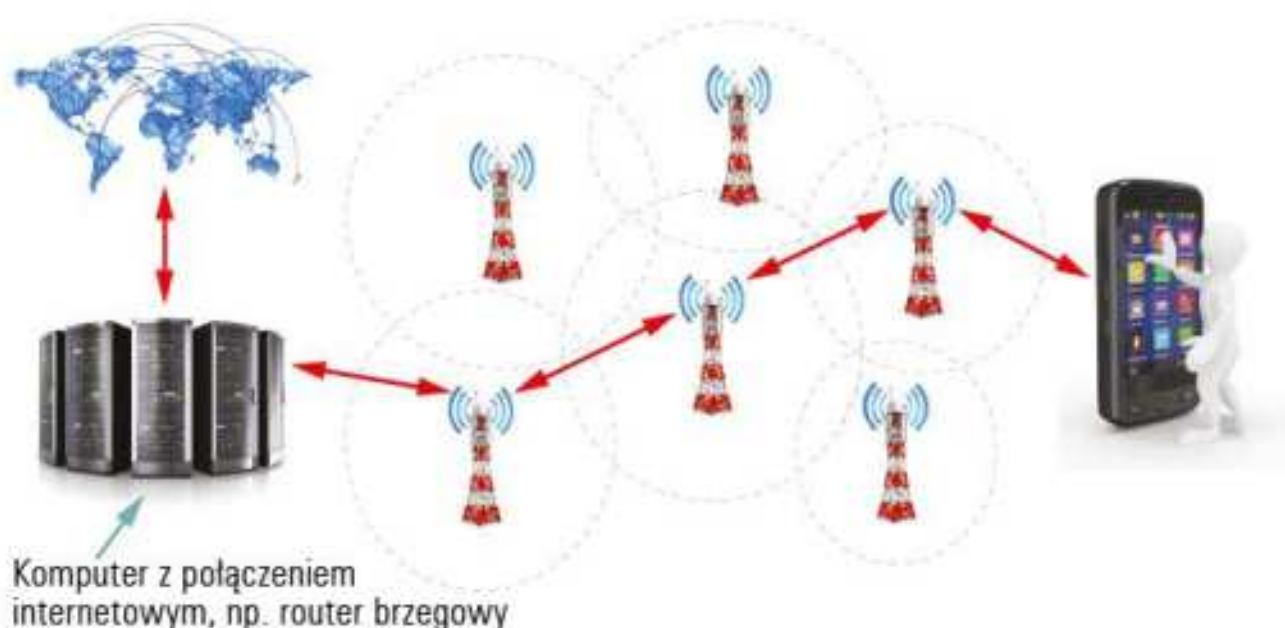
Rys. 27.2. Przykładowy schemat sieci lokalnej LAN

Wiesz już, jak łączy się w sieć komputery i inne urządzenia. Medium transmisyjnym może być kabel miedziany, światłowód lub połączenie bezprzewodowe WiFi. Taki model sieci komputerowej nie obejmuje połączeń za pośrednictwem sieci telefonii komórkowej. Jeśli łączysz się z innym telefonem, twój smartfon nie musi korzystać z internetu. Jednak z racji tego, że transmisja rozmowy to w rzeczywistości przesyłanie danych cyfrowych, wykorzystywane są te same stacje przekaźnikowe (rys. 27.3.).



Rys. 27.3. Połączenie w trakcie rozmowy w telefonii komórkowej

Podczas połączeń smartfona z internetem ta sama sieć radiowa łączy go z centrum komputerowym pełniącym funkcję routera brzegowego. Takie urządzenia pracują na granicy między siecią komórkową danego operatora a internetem (rys. 27.4.). Niektóre współczesne stacje przekaźnikowe także są wykorzystywane jako routery.



Rys. 27.4. Przykładowe połączenie pomiędzy smartfonem a internetem w sieci komórkowej

27.2. Zasady ruchu, czyli protokoły w internecie

Połączenia internetowe pomiędzy komputerami muszą zapewniać nie tylko bezpośrednią łączność, lecz także sprawić, by komputery wzajemnie się rozumiwały. To możliwe dzięki **protokołom komunikacyjnym**, czyli ściśle ustalonym i wspólnym w całej sieci zasadom przesyłania informacji i nawiązywania połączeń. Opisują one zarówno zagadnienia informatyczne, jak i elektryczne, wyznaczają parametry dla konstruktorów urządzeń sieciowych i oprogramowania.



Urządzenia sieciowe realizują je automatycznie. W internecie funkcjonuje wiele protokołów komunikacyjnych, najważniejsze to:

IP (Internet Protocol, czytaj: internet protokol) – odpowiedzialny za wybór trasy i przesłanie informacji w postaci pakietów. Został on utworzony w taki sposób, by w razie awarii automatycznie zestawiał alternatywne połączenie.

TCP (Transmission Control Protocol, czytaj: transmyszyn kontrol protokol) – zapewnia bezpieczne przesyłanie danych, potwierdza odebrane dane i zapewnia kontrolę poprawności ich przesyłania. Dzięki niemu możesz bez błędów przeglądać strony internetowe, przysyłać pocztę, przysyłać pliki itp.

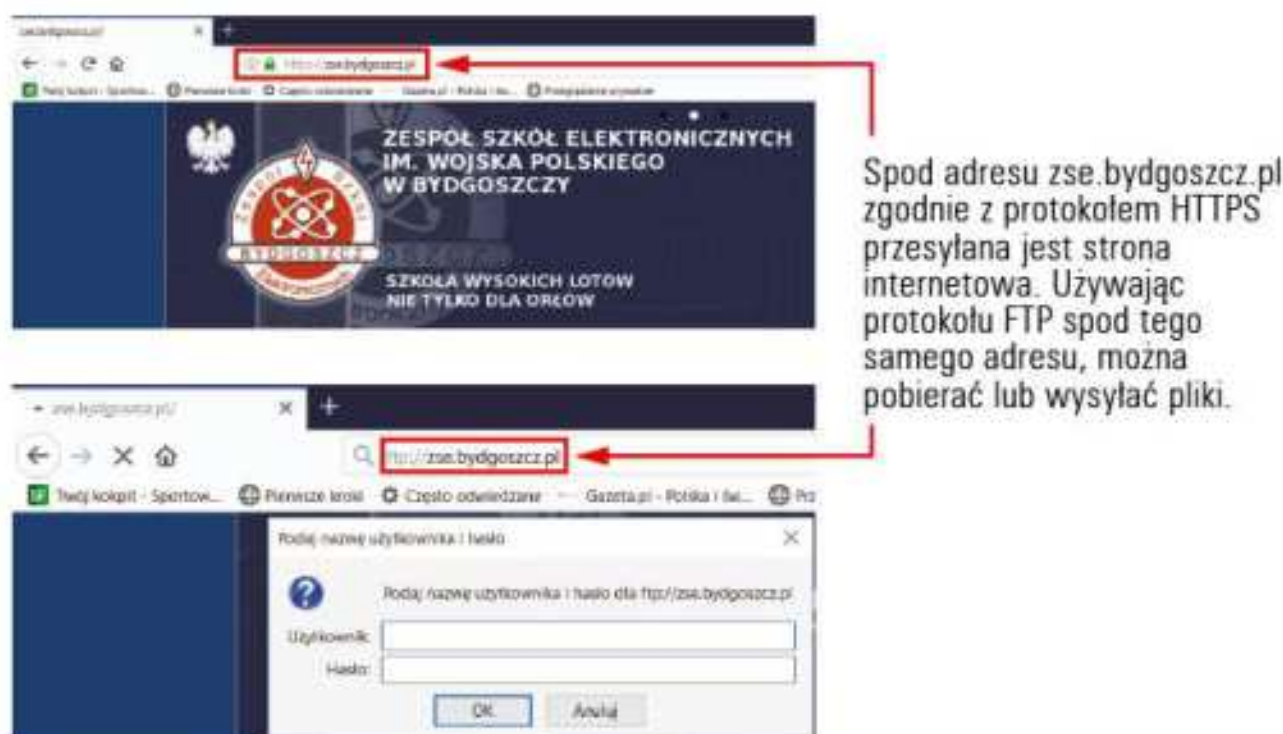
Oba protokoły są wykorzystywane przez inne, bardziej wyspecjalizowane:

HTTP (Hypertext Transfer Protocol, czytaj: hajpertekst transfer protokol) lub **HTTPS** (Hypertext Transfer Protocol Secure, czytaj: sekjur) – umożliwia przesyłanie i wyświetlanie stron internetowych, czego ślad widać w pasku adresów przeglądarki, w których pokazuje się adres poprzedzony nazwą protokołu (rys. 27.5.). Odmiana HTTPS pozwala na szyfrowanie połączenia. Dzięki temu łączenie się ze stronami (zakładanie kont, logowanie się na własne profile, na przykład konta bankowe, aukcje, sklepy, pocztę) staje się bezpieczniejsze.



Rys. 27.5. Nazwa protokołu przed adresem strony

FTP (File Transfer Protocol, czytaj: fajł transfer protokol) – służy do przesyłania plików. Korzystają z niego programy do zarządzania zawartością dysków sieciowych, w tym także przeglądarki internetowe (rys. 27.6.).



Rys. 27.6. Przykłady działania protokołów dla tego samego adresu



SMTP (Simple Mail Transfer Protocol, czytaj: simpyl mejl transfer protokol) – został stworzony do przesyłania poczty elektronicznej.

Warunki i zasady opisane w specyfikacji protokołów obowiązują w całym internecie. Dzięki temu nie ma problemu z kompatybilnością. Przeglądarka i klient usług internetowych akceptują je bez względu na zainstalowany system operacyjny.

27.3. Komputery ponumerowane, czyli adresy w internecie

Wiesz już, że komputery w sieciach mają swoje adresy IP. Adresy IP w sieci mają hosty. Host to komputer lub urządzenie sieciowe. Jednak w sieci pracuje niezliczona ilość urządzeń, czy zatem wystarczy dla wszystkich adresów? Przede wszystkim internetowe adresy IP są przypisane jedynie urządzeniom podłączonym bezpośrednio do internetu. Na przykład szkolna pracownia może być włączona do internetu za pośrednictwem modemu z routerem, a w sieci globalnej cała będzie funkcjonować pod jednym adresem przydzielonym modemowi (adres publiczny). Komputery wewnątrz sieci dostaną adresy przypisane przez router (adresy prywatne). Twój komputer może mieć taki sam adres, jak inny w sąsiedniej pracowni obsługiwanej przez inny router (pracujący w innej podsieci).

protokół DHCP

Jak komputer uzyskuje IP? W twojej sieci lokalnej odpowiada za to protokół **DHCP** (Dynamic Host Configuration Protocol, czytaj: dajnamik host konfigu-rejszyn protokol), czyli usługa działająca w urządzeniu zarządzającym siecią. To dzięki niemu komputer podłączony do sieci pobiera niezbędne dane ustawień parametrów, które odpowiadają za poprawną pracę. Wśród nich jest także adres IP.

adresy dynamiczne i statyczne

Urządzenie zarządzające siecią może przyznawać **adresy dynamicznie** – za każdym razem, gdy host loguje się do sieci, może otrzymać różne IP, najczęściej pierwsze wolne. IP w sieci może być także **statyczne**, czyli przydzielone danemu hostowi na stałe. Za każdym razem, gdy loguje się on do sieci, otrzymuje ten sam adres.

ICANN

A kto przyznaje adresy urządzeniom sieciowym w internecie? Aby zachować zasadę, że każdy z adresów jest niepowtarzalny, zarządza nimi instytucja o światowym zasięgu – ICANN (The Internet Corporation for Assigned Names and Numbers, czytaj: de internet korporejszyn for asajnd nejms end nambers). Robi to za pośrednictwem organizacji regionalnych; na terenie Europy jest to RIPE (Réseaux IP Européens). Te z kolei przydzielają adresy instytucjom zarządzającym w poszczególnych krajach. W Polsce adresy dostawcom internetu nadaje NASK (Naukowa i Akademicka Sieć Komputerowa).



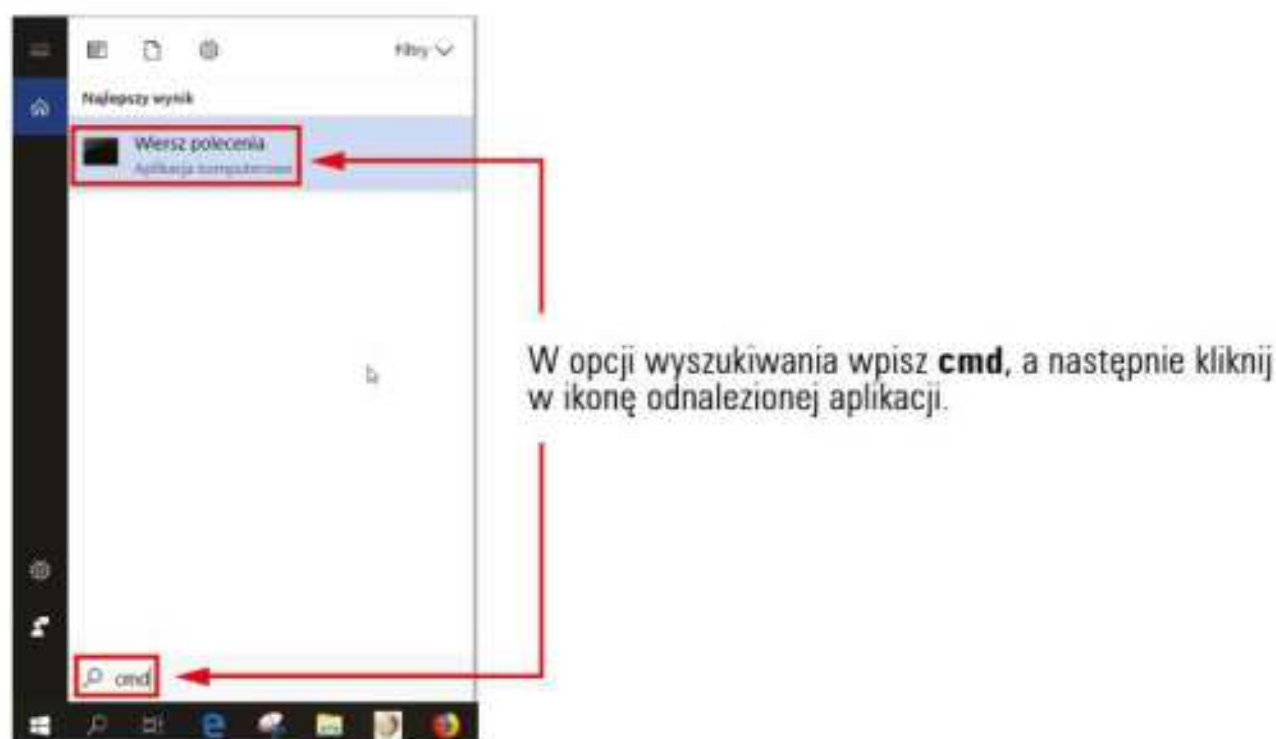
Rys. 27.7. Instytucje czuwające nad przyznawaniem IP w internecie

Strony internetowe i usługi mają więc adres komputera, na którym je zainstalowano. Taki komputer nazywamy **hostem**. Udostępnianie przez dostawcę miejsca na usługi internetowe i strony to **hosting**.

Protokół IP zestawia połączenia między hostami i może je zmieniać (tak samo jak w przypadku wystąpienia zakłóceń lub awarii). Zdarza się, że podczas wczytywania strony internetowej dane biegną do nas inną drogą. Taki proces nazywamy routowaniem dynamicznym wyboru trasy. Czy możliwe jest śledzenie takiego połączenia? Wypróbuj dwa sposoby.

Uruchom **Wiersz poleceń (CMD)** systemu Windows (rys. 27.8.). Następnie wpisz polecenie **tracert** i adres strony, do której drogę chcesz prześledzić. **Tracert** wyświetla informacje o trasie i punktach węzłowych (routerach), przez które muszą przejść pakiety danych, by dotrzeć z serwera do twojego komputera. Polecenie to wyświetla także czas, w którym pakiety przebywają dany odcinek. Jako pierwsze zostaną wyświetlone węzły najbliższe twojemu komputerowi (rys. 27.9.).

polecenie tracert



Rys. 27.8. Uruchamianie Wiersza poleceń w systemie Windows 10



```
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>tracert men.gov.pl

Tracing route to men.gov.pl [194.54.27.118]
over a maximum of 30 hops:

  0  <1 ms    <1 ms    <1 ms    livebox.home [192.168.1.1]
  1  21 ms     20 ms     21 ms     byd-bng2.tpnet.pl [80.50.158.186]
  2  21 ms     20 ms     21 ms     byd-r1.tpnet.pl [80.50.158.185]
  3  25 ms     25 ms     24 ms     war-ar2.tpnet.pl [195.205.0.10]
  4  29 ms     26 ms     26 ms     z-netia-war-ar6.tpnet.pl [80.50.235.90]
  5  *         25 ms     25 ms     warsc001rt12.inetia.pl [83.238.248.175]
  6  25 ms     25 ms     30 ms     81.219.213.85
  7  26 ms     25 ms     25 ms     as24929.r9-xe-10-1-0-902-Mow-Linx-PL.linxtelecos.net [212.27.226.91]
  8  25 ms     25 ms     25 ms     men.gov.pl [194.54.27.118]

Trace complete.

C:\Users\Wojtek>
```

IP przydzielone przez DHCP domowego routera użytkownika wczytującego stronę

IP routerów (węzłów) pośredniczących

IP serwera badanej strony

Rys. 27.9. Efekt działania polecenia tracert na przykładzie strony men.gov.pl

Drugim sposobem jest użycie jednego z programów, jak **VisualRoute** (rys. 27.10.). Dane są przedstawione w formie wykresu. W wersji komercyjnej można również śledzić połączenia na mapie świata. Zauważ, że oba sposoby wskazały inne połączenie. Nie jest to błędne wskazanie, lecz efekt działania protokołu IP, który zestawiał połączenia.



Rys. 27.10. Śledzenie połączenia z serwerem strony internetowej za pomocą programu VisualRoute

W obu przypadkach końcowym IP jest 194.54.27.118, ponieważ pod tym adresem umieszczono daną stronę.

27.4. Zrozumiałe dla użytkownika, czyli adresy symboliczne

W ćwiczeniu obok IP wyświetlano też nazwy, które łatwo zidentyfikowałeś/zidentyfikowałaś jako adresy internetowe. Można więc uznać, że IP w postaci czterech liczb oddzielonych kropkami są z nimi tożsame. Sprawdź to na przykładzie badanej przez ciebie strony.



Otwórz przeglądarkę i w osobnych kartach wpisz adres symboliczny `men.gov.pl` oraz `194.54.27.118` (rys. 27.11.). Wciśnij **Enter** i poczekaj na efekt. Czy wczytała się ta sama strona w obu kartach? Przeglądarka w obu przypadkach połączyła się z tym samym serwerem stron i wczytała tę samą stronę.

Dlaczego tak się dzieje? Odpowiedź jest prosta – chodzi o wygodę korzystania z sieci WWW. Czy w trakcie czytania rozdziału zapamiętałeś/zapamiętałaś IP strony internetowej? Właśnie. A adres symboliczny zapisał się w pamięci bez problemu.

Jak to możliwe, że wpisanie adresu symbolicznego jest także zrozumiałe dla sieci? Zawdzięczamy to komputerom w centrach DNS, w których są przechowywane informacje o adresach IP i odpowiadających im domenach, czyli adresach symbolicznych. Przeglądarka wysyła zapytanie do DNS o adres symboliczny i po jego otrzymaniu łączy się z tym adresem. Skąd dane w DNS? Wykupując domenę, czyli adres symboliczny, właściciel musi ją powiązać z konkretnym serwerem, który świadczy usługę hostingową, co oznacza, że udostępnia stronę w sieci. A ten ma konkretne IP.

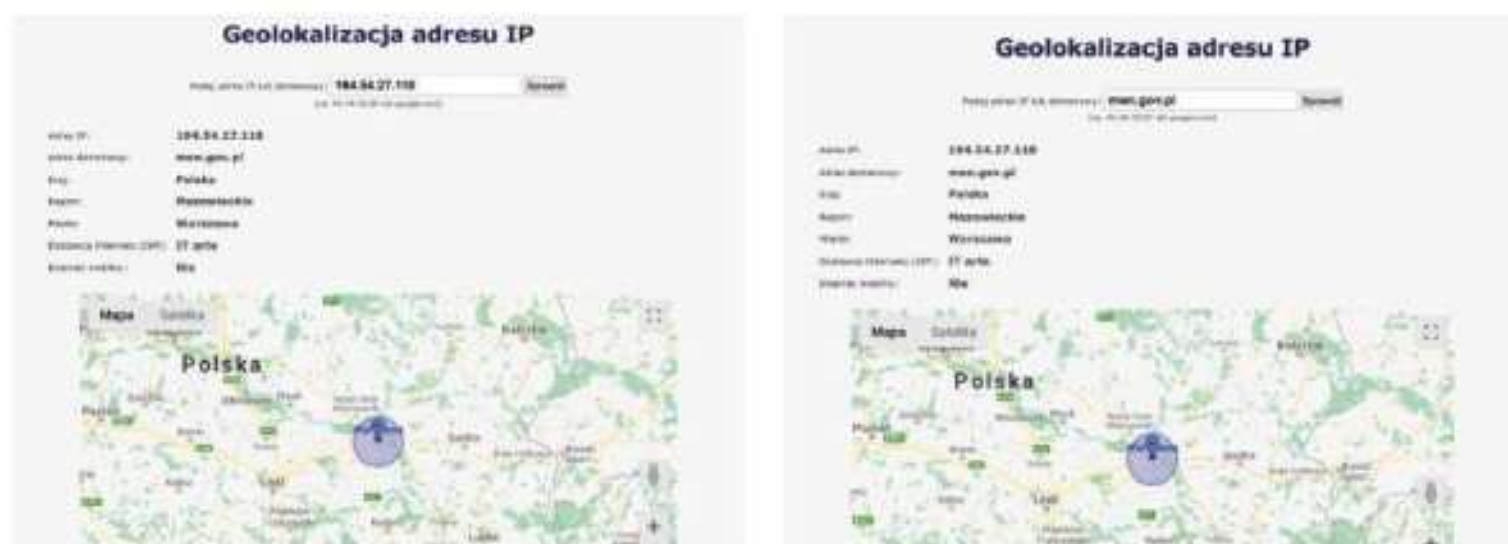
DNS

Za pomocą **tracert** możesz sprawdzić IP znanych ci stron, na przykład strony szkoły.



Rys. 27.11. Adresy tej samej strony w różnej postaci

W sieci funkcjonują także strony, które odnajdują dane właścicieli domen oraz przypisane im IP lub na podstawie IP podają domenę (rys. 27.12.). Zauważ, że te same dane pokażą się po wpisaniu zarówno IP, jak i adresu symbolicznego.



Rys. 27.12. Dane adresowe strony internetowej odczytane ze strony z usługą geolokalizacji



ZADANIA DO ROZWIĄZANIA

1. Wczytaj do przeglądarki stronę men.gov.pl i sprawdź, jakie symbole pojawiają się po prawej stronie adresu wyświetlanego w polu adresowym przeglądarki. Korzystając z dowolnych, wiarygodnych źródeł, wytłumacz ich znaczenie. Odpowiedz, dlaczego pojawiły się w tym miejscu i jaki to ma wpływ na relacje pomiędzy użytkownikiem a właścicielem strony.
2. Skorzystaj z programu VisualRoute w wersji lite (darmowej), by zbadać trasę zestawioną przez protokół IP od twojego komputera do twojej ulubionej strony informacyjnej. Powtórz ćwiczenie i porównaj wyniki działania programu. Wskaż różnice i wyjaśnij, dlaczego się pojawiły.
3. Użyj polecenia `tracert` do odczytania drogi połączenia ze stroną twojej szkoły. Odczytaj nazwy węzłów i sprawdź, jak długo trwało łączenie pomiędzy nimi. Na podstawie informacji pozyskanych z wiarygodnych stron wyjaśnij, skąd wynikają różnice.
4. Odszukaj w internecie strony, które świadczą usługi udostępniania danych na temat właścicieli domen internetowych (adresów symbolicznych). Odczytaj także inne dane, np. o serwerze, na którym znajduje się strona o danym adresie.
- 5*. Znajdź w sieci usługi, w których można zarezerwować i kupić domenę internetową (adresu symbolicznego). Sprawdź, czy domena zawierająca twoje nazwisko jest wolna i ile kosztuje jej wykupienie. Porównaj ceny domen `.pl`, `.com`, `.eu`. Opisz, czym się różnią z punktu widzenia użytkownika i odbiorcy strony internetowej spod danego adresu.

PODSUMOWANIE LEKCJI

- s. 204 • **Internet** jest siecią szkieletową, która łączy ze sobą sieci lokalne. Mogą one być niewielkie, np. w mieszkaniu, i rozległe, np. w dzielnicy lub wsi.
- s. 207 • Wszystkie urządzenia i programy sterujące pracą sieci muszą działać według zasad opisanych w specyfikacji protokołów sieciowych. Dwa podstawowe to **IP** – opisuje sposób łączenia komputerów, i **TCP** – reguluje zasady przesyłania informacji. Uwzględniając te zasady, działają pozostałe protokoły, np. **HTTP** – protokół stron internetowych, **FTP** – przesyłanie plików, **SMTP** – odpowiedzialny za przesyłanie poczty i inne.
- s. 208 • Każdy komputer wpięty bezpośrednio do internetu, w tym routery i serwery, ma swój adres IP, który przydziela **ICANN** (odpowiednikiem tej instytucji w Polsce jest **NASK**).
- s. 208 • Adres w sieci lokalnej przydziela **usługa protokołu DHCP**.
- s. 209 • Drogę połączenia z danym punktem mającym własne IP w internecie można śledzić za pomocą polecenia **tracert** wydanego w wierszu poleceń Windows lub odpowiedniego programu.
- s. 211 • Adresy symboliczne i odpowiadające im adresy IP są przechowywane w **serwerach DNS**. Przeglądarka wysyła zapytanie do DNS, przesyłając adres symboliczny, i otrzymuje odpowiadający mu IP.



28. Sieć to nie tylko internet, czyli poznajemy topologię sieci komputerowych

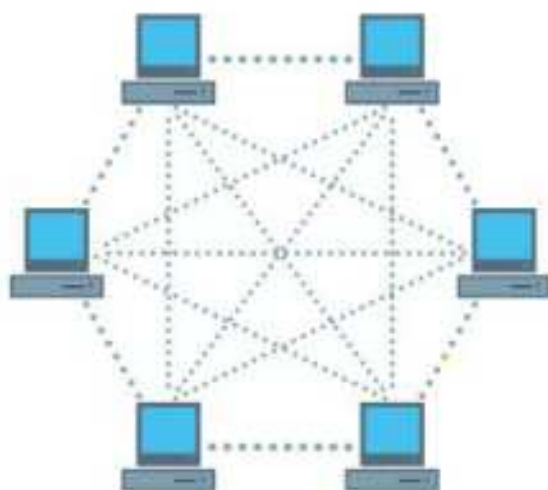
NA TEJ LEKCJI:

- dowiesz się, czym jest sieć P2P oraz topologia gwiazdy.

Internet to sieć szkieletowa łącząca mniejsze sieci. Ale jak są zbudowane mniejsze sieci? Czy rządzą nimi takie same prawa, jakie obowiązują w internecie? Jakich urządzeń potrzeba do uruchomienia małej sieci w domu lub biurze?

28.1. Komputer z komputerem, czyli sieć P2P

Najprostszą sieć mogą tworzyć dwa komputery połączone ze sobą w taki sposób, by możliwa była wymiana danych. Gdy łączysz się poprzez bluetooth lub WiFi z komputerem, by skopiować na jego dysk na przykład zdjęcia, tworzysz małą, lokalną, najprostszą sieć typu P2P (Peer-to-Peer, czytaj: pir-tu-pir). Może być ich oczywiście więcej. W tego rodzaju sieci komputery łączą się ze sobą bez pośrednictwa innych urządzeń i wszystkie mają takie same prawa (rys. 28.1.). Strukturę z rysunku 28.1. nazywa się także strukturą pełnej siatki.



Rys. 28.1. Komputery w sieci P2P, w której wszystkie komputery są ze sobą połączone



Rys. 28.2. Proces zestawienia połączenia P2P urządzeń z Bluetooth



W sieci P2P każdy komputer może pełnić funkcję hosta (komputera użytkownika) lub serwera. Wymiana danych odbywa się bezpośrednio pomiędzy nimi. Takie komputery nazywa się węzłami sieci. Z racji jej budowy P2P ma zastosowanie głównie do wymiany plików – podobnie jak we wspomnianym przykładzie z telefonem komórkowym i komputerem. Jest też stosowana do nielegalnego rozpowszechniania plików multimedialnych. Udostępnianie ich w ten lub inny sposób bez zezwolenia jest przestępstwem. Jeśli takie połączenie komputerów wykorzystano do budowy sieci lokalnej, może to zwiększyć niezawodność połączeń. Oczywiście warunkiem jest prawidłowe działanie wszystkich komputerów.

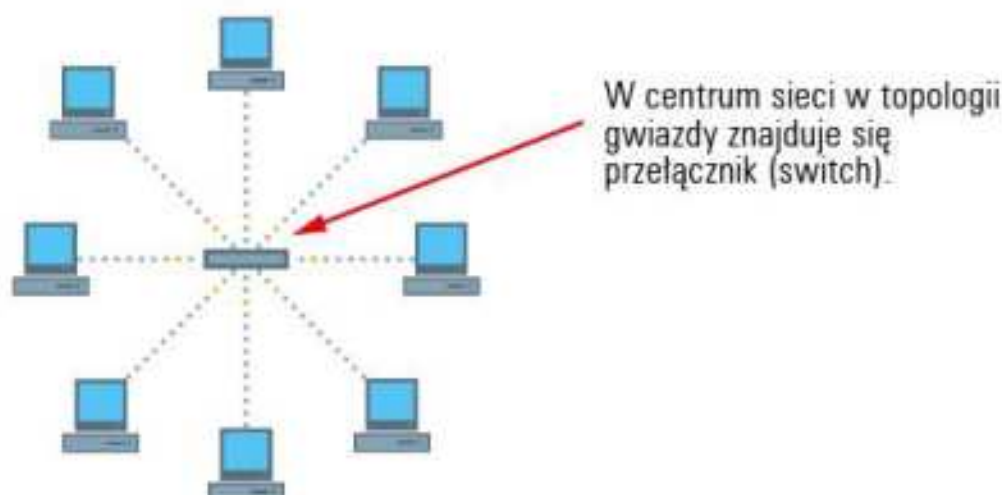
Jak nawiązać połączenie między smartfonem a komputerem? W pewnym stopniu zależy to od wersji ich systemów, ale ogólne zasady są niezmiennie (rys. 28.2.).

28.2. Przełączniki, czyli więcej o infrastrukturze sieci lokalnych

Sieci lokalne, bez względu na liczbę połączonych komputerów, muszą współdziałać w ramach pewnej infrastruktury. Najważniejsze w niej są łącza – przewodowe lub bezprzewodowe. Sposób połączenia komputerów (fizyczny i logiczny) nazywamy topologią. Istnieje wiele różnych rozwiązań technicznych.

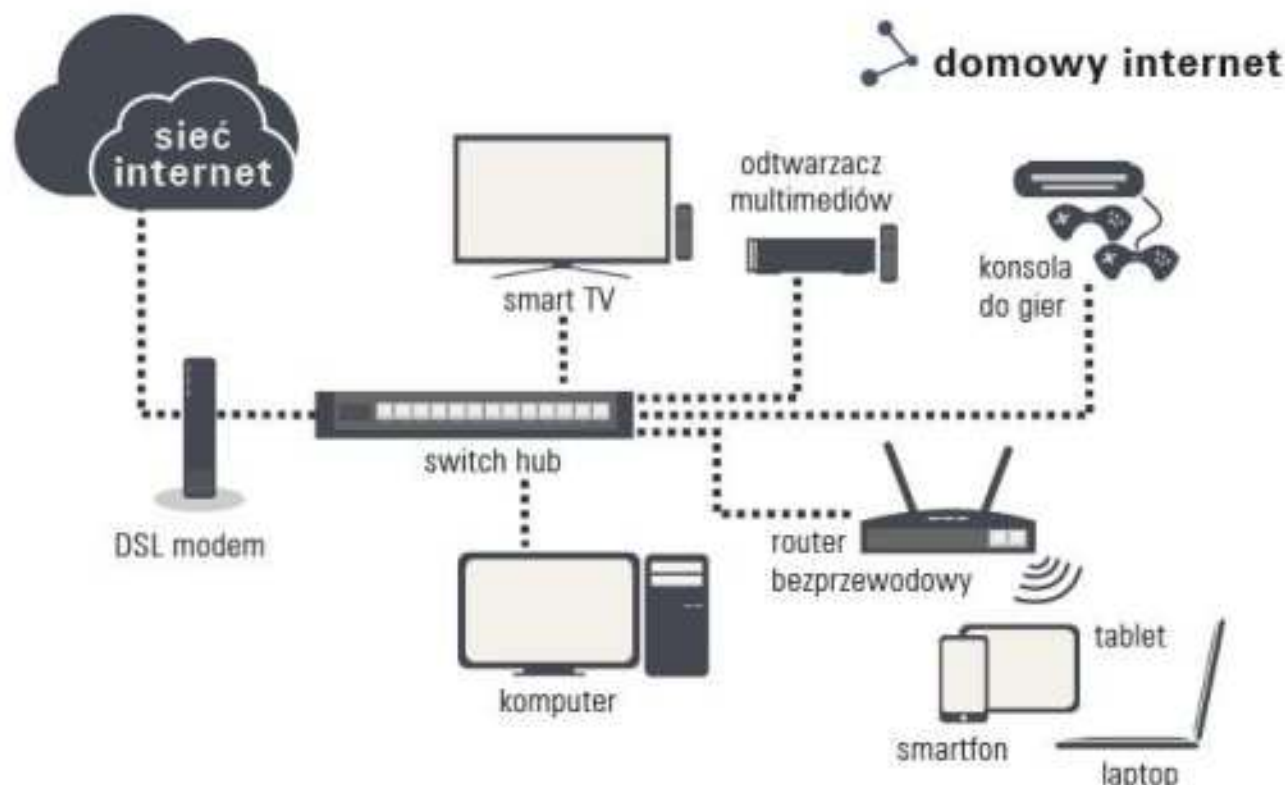
Najpopularniejsze obecnie spotykane w przedsiębiorstwach, biurach i mieszkaniach nosi nazwę **topologia gwiazdy** (rys. 28.3.).

topologia gwiazdy



Rys. 28.3. Sieciowa topologia gwiazdy

Z takim połączeniem zapewne masz do czynienia w swojej domowej sieci. To rozwiązanie zakłada, że głównym elementem infrastruktury jest przełącznik. W ten sposób łączy się nie tylko komputery, lecz także inne urządzenia (rys. 28.4.).



Rys. 28.4. Przykładowa domowa sieć łącząca różne urządzenia

Transfer danych w takiej sieci odbywa się za pośrednictwem przełącznika, z którym są połączone wszystkie urządzenia, w tym także router sieci bezprzewodowej. Dostawcy internetu często udostępniają urządzenia, które łączą w sobie modem, przełącznik sieciowy i router wraz z punktem dostępowym WiFi.

Do zalet topologii gwiazdy należą:

- duża odporność na awarie,
- awaria pojedynczego przewodu lub komputera nie zakłóca pracy sieci,
- większa przepustowość,
- łatwa lokalizacja uszkodzeń,
- duża wydajność,
- łatwa rozbudowa.

Wśród wad topologii gwiazdy można wymienić:

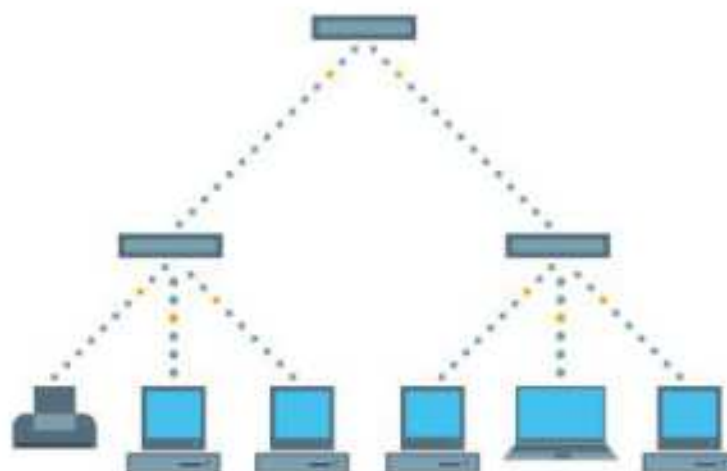
- zależność funkcjonowania sieci od jednego urządzenia centralnego,
- konieczność doprowadzenia przewodów od przełącznika do urządzeń podłączanych do sieci, co w niektórych przypadkach łączy się z koniecznością robienia przepustów przez ściany i stropy.

Topologia gwiazdy rozszerzonej pozwala na łączenie ze sobą sieci zbudowanych w topologii gwiazdy (rys. 28.5.). Można ją zastosować, gdy odległości pomiędzy poszczególnymi grupami komputerów, na przykład między biurami na różnych kondygnacjach, są duże. Prowadzenie kabli od przełącznika do każdego z komputerów byłoby niewskazane (grube, długie wiązki). Dlatego w takim

topologia gwiazdy
rozszerzonej



przypadku przełączniki z poszczególnych pięter łączy się jednym przewodem z głównym przełącznikiem. Można też pozostawić opcję wymiany danych między wszystkimi urządzeniami sieci. Niektóre rodzaje przełączników się konfiguruje, aby ograniczyć dostęp do podsieci pozostałym komputerom.



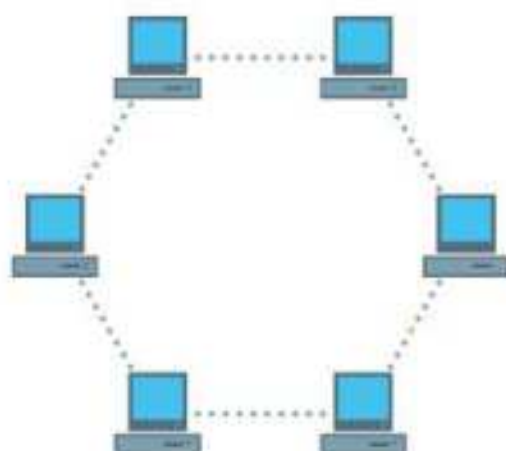
Rys. 28.5. Przykład sieci o topologii gwiazdy rozszerzonej

Zaletami topologii gwiazdy rozszerzonej są:

- możliwość łączenia ze sobą mniejszych sieci,
- ograniczenie liczby kabli między głównym przełącznikiem a znacznie oddalonymi grupami komputerów i urządzeń.

Do wad topologii gwiazdy rozszerzonej należą:

- konieczność rozbudowy infrastruktury o dodatkowe urządzenia,
- zwiększenie poboru energii przez infrastrukturę,
- konieczność przygotowania odpowiednich miejsc na dodatkowe przełączniki.



Rys. 28.6. Przykład sieci o topologii pierścienia

Topologia pierścienia to sposób połączenia komputerów, jak pokazano na rysunku 28.6. Liczba łączy i ich długość są ograniczone. Organizacja logiczna takiej sieci opiera się na tak zwanym żetonie. W danej sieci istnieje jeden komputer, który przejmie żeton i może transmitować dane. Żeton oznacza określoną sekwencję bitów.

W praktyce można spotkać jeszcze sieci o topologii mieszanej, mające cechy kilku topologii.



ZADANIA DO ROZWIĄZANIA

1. Zbadaj, jaką topologię zastosowano przy budowie sieci w twojej szkolnej pracowni komputerowej. Ustal, czy jest ona siecią jednego typu i czy łączy się z innymi sieciami lokalnymi w szkole.
2. Podaj założenia do budowy sieci w bloku mieszkalnym składającym się z 20 mieszkań położonych w 2 klatkach schodowych. Załóż, że w każdym mieszkaniu może działać więcej niż 1 komputer. Jakiego typu będzie to sieć?
3. Wykonaj swoim telefonem wspólne zdjęcie z kolegą lub koleżanką z klasy i połącz się za pośrednictwem Bluetooth z jego/jej telefonem. Po prawidłowym zestawieniu połączenia prześlij wykonane wcześniej zdjęcie. Rozłącz połączenie.
4. Ułóż algorytm w postaci listy kroków opisujący czynności podczas łączenia dwóch telefonów za pośrednictwem Bluetooth.
- 5*. Użyj jednego z narzędzi systemu Windows, aby sprawdzić, które z komputerów w pracowni są podłączone do sieci. Zwróć uwagę na ich nazwy. Spróbuj przesłać pliki na ich dysk. Jaki warunek musi być spełniony, by przesłanie się powiodło?

PODSUMOWANIE LEKCJI

- Połączenie twojego telefonu z komputerem za pośrednictwem Bluetooth jest siecią typu **P2P**. s. 213
- Najpopularniejszą topologią sieci jest **topologia gwiazdy**. s. 214
- Sieci zbudowane zgodnie z topologią gwiazdy mogą być łączone w **gwiazdę rozszerzoną**. Urządzenia komputerowe można łączyć w sieci na różne sposoby. s. 215



29. Kto tam, czyli identyfikujemy komputery w sieci

NA TEJ LEKCJI:

- nauczysz się odczytywać dane z karty sieciowej;
- dowiesz się, jak skonfigurować kartę sieciową.

Sieć komputerowa działa dzięki protokołom i infrastrukturze technicznej. W każdym urządzeniu sieciowym znajduje się karta sieciowa. Jej skonfigurowanie decyduje o poprawnym działaniu. Od protokołów zależy sprawne komunikowanie się urządzeń i działanie usług. Czym są protokoły i jak konfigurować kartę? Warto to wiedzieć, by świadomie korzystać z możliwości sieci.

29.1. Sprawdzamy konfigurację, czyli jak odczytać dane z karty sieciowej

Wiesz już, że jeśli komputer ma pracować w sieci, należy mu przydzielić adres IP. Musi tak być zarówno w sieci lokalnej, jak i internecie. Do sprawdzenia stanu połączenia i odczytania parametrów połączenia można użyć polecenia **ipconfig** z parametrem **all** (rys. 29.1.). Co otrzymaliśmy w wyniku działania tego polecenia? Odczytaj informacje z ekranu swojego komputera.

ipconfig/all

```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>ipconfig /all

Windows IP Configuration

Host Name . . . . . : komp
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No
DNS Suffix Search List. . . . . : home

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . : home
Description . . . . . : Qualcomm Atheros AR8611 PCI-E Gigabit Ethernet Controller (NDIS 6.30)
Physical Address. . . . . : 96-F4-34-6E-F4-5C
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::3d95:3dc4:b53d:1f1e%11(Preferred)
IPv4 Address. . . . . : 192.168.1.13(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Lease Obtained. . . . . : 27 października 2018 17:04:07
Lease Expires . . . . . : 28 października 2018 17:04:07
Default Gateway . . . . . : 192.168.1.1
DHCP Server . . . . . : 192.168.1.1
DHCPv6 IAID . . . . . : 131843188
DHCPv6 Client DUID. . . . . : 00-01-00-01-3F-55-96-A5-F4-5C-34-6E-F4-5C
DNS Servers . . . . . : 192.168.1.1
                        192.168.1.1
NetBIOS over Tcpip. . . . . : Enabled

C:\Users\Wojtek>
```

Rys. 29.1. Efekt działania polecenia „ipconfig/all”



Host Name (czytaj: host nejm) to w rzeczywistości nazwa twojego komputera nadana w trakcie instalacji systemu operacyjnego.

Description (czytaj: deskrypszyn) jest pełną nazwą modelu karty sieciowej. Z ilustracji można odczytać, że to karta Ethernet (czytaj: eternet; to karta przewodowa) o przepustowości 1 gigabita z układem Qualcomm Atheros (czytaj: kłalikom ateros).

Physical Address (czytaj: fizykal adres) to unikatowy, fizyczny adres karty sieciowej zwany także **MAC adresem** (Media Access Control Address, czytaj: midia akses kontrol adres). Każda karta sieciowa musi mieć niepowtarzalny adres fizyczny. Producenci wpisują je na stałe, wybierając z przydzielonej im puli.

Karta z ilustracji ma także włączone automatyczne pobieranie adresu IP (**DHCP Enabled**, czytaj: enejbled) oraz opcję automatycznej konfiguracji (**Autoconfiguration Enabled**, czytaj: autokonfiguurejszyn enejbled).

IPv4 Address oznacza adres IP komputera w sieci; w tym przypadku przydzielony przez router w ramach usługi DHCP.

Ponieważ nasz komputer łączy się z inną siecią (internetem), musi korzystać z bramy sieciowej (**Gateway**, czytaj: gejtłej).

DHCPv6 pełni podobną funkcję co DHCPv4. Wprowadzono go po modyfikacjach protokołu IP. To protokół komunikacyjny pozwalający na pobranie z serwera danych potrzebnych do konfiguracji połączenia.

Adres IP został przydzielony karcie (komputerowi) przez router, więc adresem urządzenia z usługą **DNS** jest router (**DNS Servers**, czytaj: serwery) pod swoim adresem.

29.2. Ping, czyli sprawdzamy poprawność połączenia

Jak sprawdzić, czy twój komputer może nawiązać kontakt z innym urządzeniem podłączonym do sieci? Załóżmy, że twój telefon łączy się z siecią za pośrednictwem karty WiFi. Oznacza to, że może pracować w tej samej sieci lokalnej, co twój komputer. Aby sprawdzić, czy rzeczywiście tak jest, użyj polecenia **ping** z **Wiersza polecenia** systemu Windows. Zanim jednak rozpoczniesz doświadczenie, odczytaj adres IP przydzielony w sieci karcie twojego telefonu (rys. 29.2.). W tym celu przytrzymaj chwilę ikonę połączenia WiFi i wybierz opcję konfiguracji karty.

ping



Adres IP przydzielony przez usługę DNS routera sieci lokalnej.

Rys. 29.2. Odczyt IP przydzielonego karcie telefonu z systemem Android



Otwórz **Wiersz polecenia (CMD)** i użyj polecenia **ping**. Ponieważ sprawdza ono połączenie z urządzeniem wyposażonym w kartę sieciową podłączoną do sieci, jako argumentu użyj odczytany z komórki IP (rys. 29.3.).

```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time=1021ms TTL=64
Reply from 192.168.1.10: bytes=32 time=7ms TTL=64
Reply from 192.168.1.10: bytes=32 time=205ms TTL=64
Reply from 192.168.1.10: bytes=32 time=419ms TTL=64

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 7ms, Maximum = 1021ms, Average = 413ms

C:\Users\Wojtek>
```

Nawiązanie połączenia powiodło się. Pakiety 32 bajtów wysyłane przez komputer wracały po upływie wyświetlonego czasu. Proces przebiegł bez błędów.

Rys. 29.3. Polecenie ping nawiązało kontakt z telefonem z kartą sieciową WiFi

W przypadku nienawiązania kontaktu z kartą sieciową telefonu otrzymasz komunikat jak na rysunku 29.4. Oczekiwanie na odpowiedź przekroczy czas ustalony dla testu.

```
Microsoft Windows [Version 10.0.17134.786]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.1.13: Destination host unreachable.
Reply from 192.168.1.13: Destination host unreachable.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),

C:\Users\Wojtek>
```

Rys. 29.4. Polecenie ping nie nawiązało kontaktu z telefonem z kartą sieciową WiFi

Czy tym samym poleceniem można sprawdzić poprawność podłączenia naszego komputera do internetu? Domyślasz się, jak to zrobić? Wykorzystaj polecenie **ping**. W miejscu adresu IP komputera wpisz domenę (rys. 29.5.).

```
Microsoft Windows [Version 10.0.17134.786]
(c) 2018 Microsoft Corporation. Wszelkie prawa zastrzeżone.

C:\Users\Wojtek>ping men.gov.pl

Pinging men.gov.pl [194.54.27.118] with 32 bytes of data:
Reply from 194.54.27.118: bytes=32 time=24ms TTL=57
Reply from 194.54.27.118: bytes=32 time=24ms TTL=57
Reply from 194.54.27.118: bytes=32 time=24ms TTL=57
Reply from 194.54.27.118: bytes=32 time=24ms TTL=57

Ping statistics for 194.54.27.118:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 24ms, Maximum = 24ms, Average = 24ms
```

Możesz odczytać adres IP hosta, na którym umieszczono stronę spod adresu men.gov.pl.

Zapis oznacza nawiązanie połączenia z hostem spod adresu przypisanego do domeny men.gov.pl.

Rys. 29.5. Sprawdzenie połączenia z internetem za pomocą polecenia ping



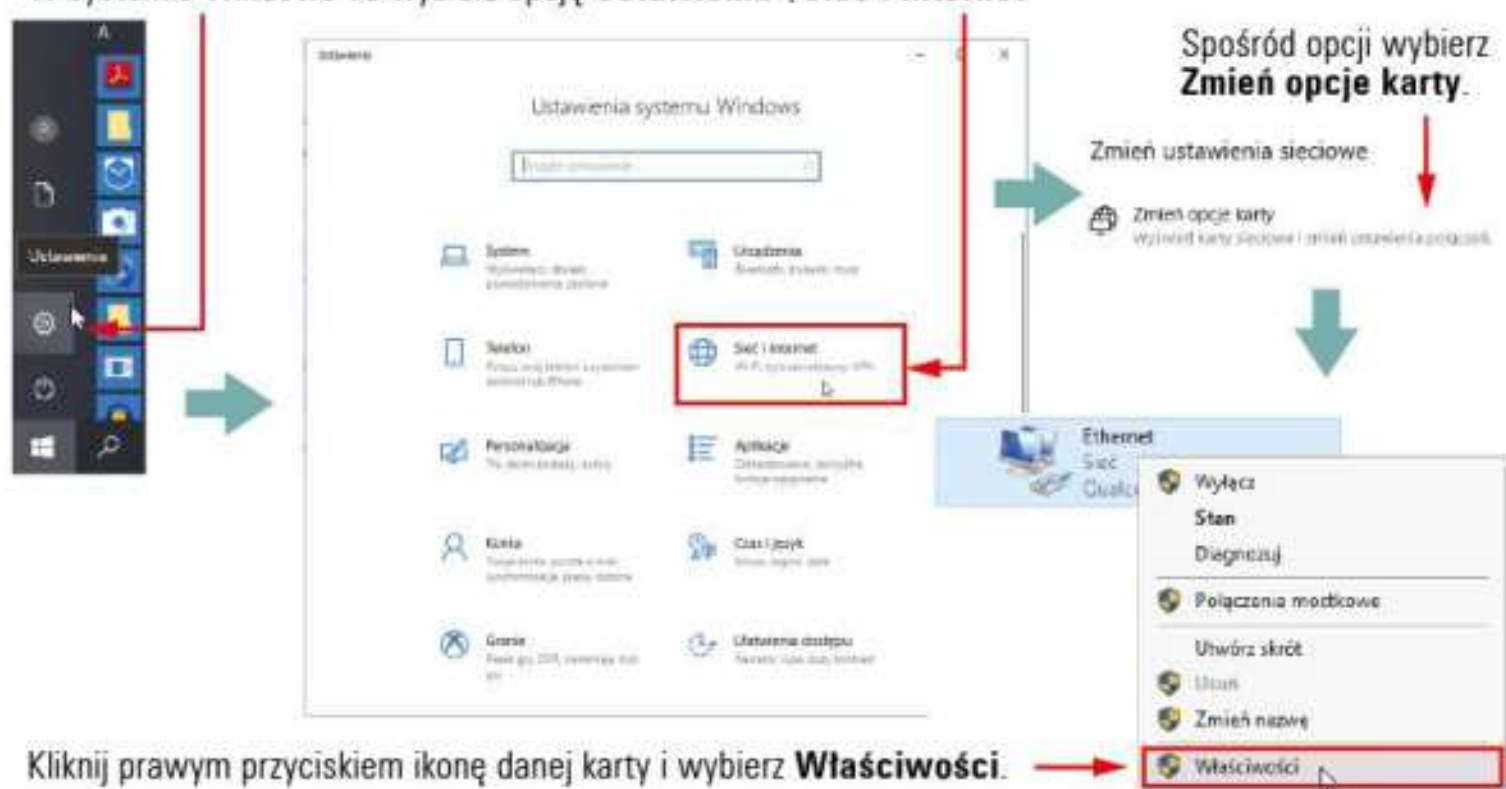
29.3. Kłopoty z siecią, czyli jak skonfigurować połączenie

Współczesne systemy operacyjne potrafią samodzielnie skonfigurować kartę sieciową, umożliwiając komputerowi pracę w sieci i w internecie. Może się jednak zdarzyć, że zajdzie potrzeba samodzielnego wykonania tych czynności.

konfiguracja karty sieciowej

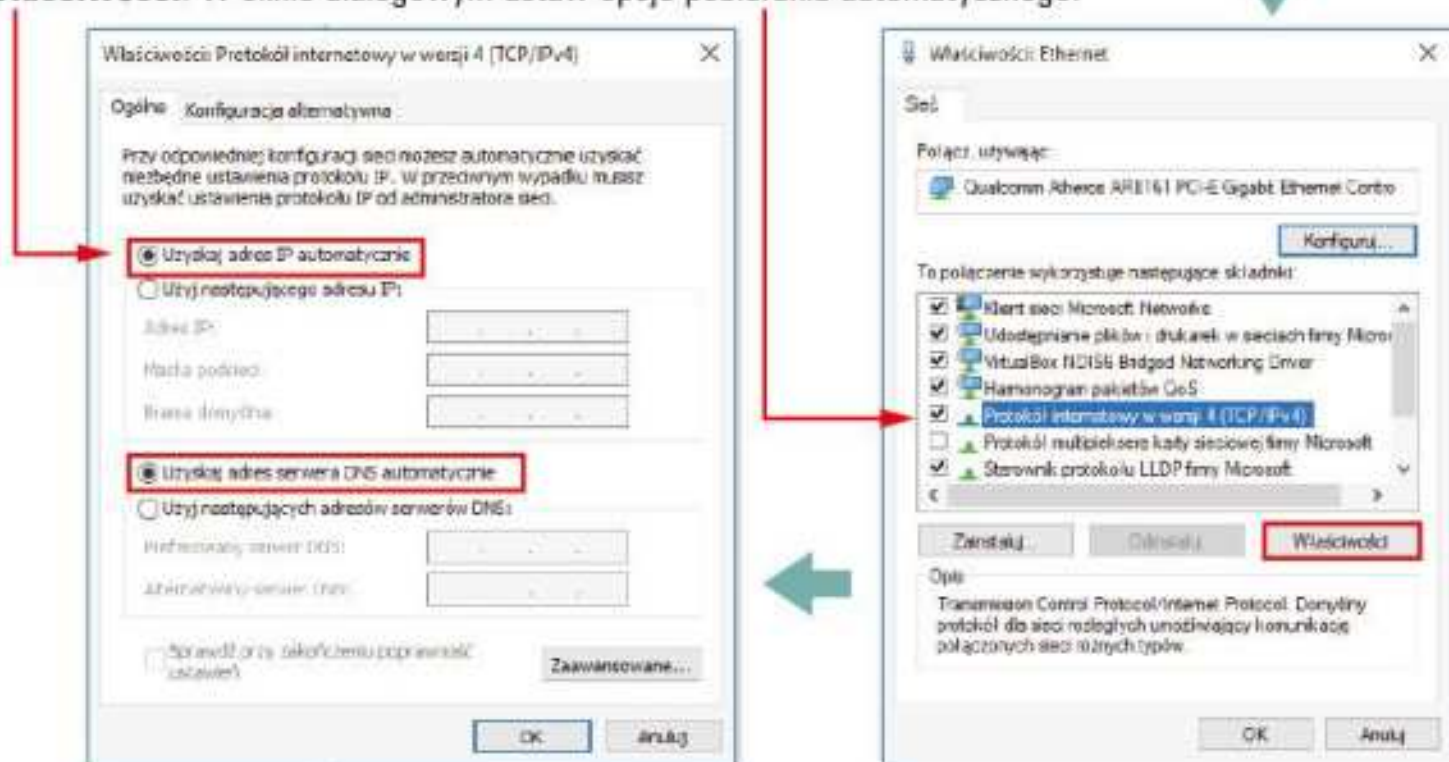
Zacznijmy od konfiguracji powodującej automatyczne wczytywanie adresu z pomocą protokołu DHCP (rys. 29.6.).

W systemie Windows 10 wybierz opcję **Ustawienia \ Sieć i Internet**



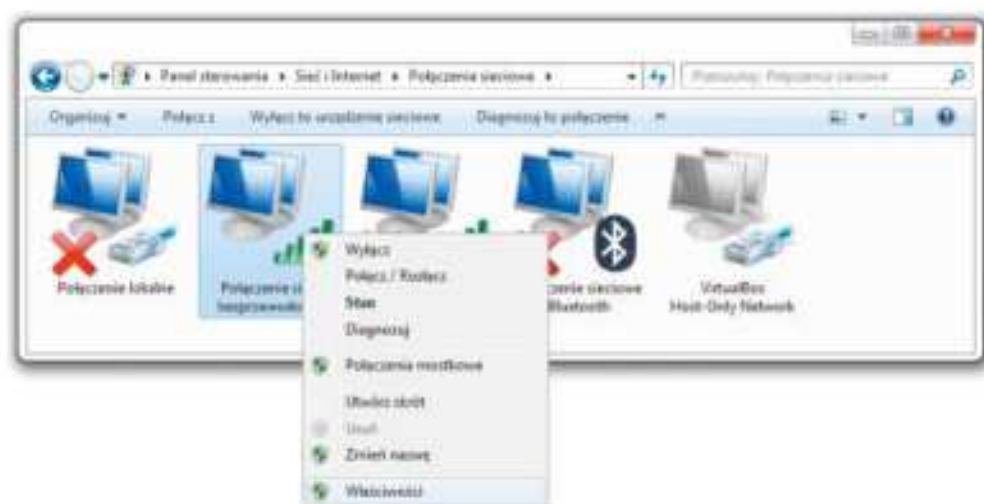
Kliknij prawym przyciskiem ikonę danej karty i wybierz **Właściwości**.

Zaznacz **Protokół internetowy w wersji 4** i wybierz przycisk **Właściwości**. W oknie dialogowym ustaw opcje pobierania automatycznego.



Rys. 29.6. Ustawienie opcji automatycznego pobierania adresów z routera

Jeśli pracujesz w systemie Windows 7, rozpocznij od **Panelu sterowania** (rys. 29.7).



Rys. 29.7. Opcja „Właściwości karty sieciowej” w systemie Windows 7

Gdy wybierzesz opcję ręcznego ustawienia parametrów połączenia, musisz znać parametry danej sieci (rys. 29.1.). Adresy serwerów DNS dla połączeń internetowych mogą być różne dla różnych dostawców internetu, na przykład dla Neostrady Orange są to: 194.204.152.34 i 194.204.159.1.

ZADANIA DO ROZWIĄZANIA

1. Sprawdź ustawienia połączenia sieciowego karty sieciowej twojego komputera. Wykorzystaj polecenie **ipconfig**. Porównaj je z danymi innego komputera pracującego w tej samej sieci, np. na sąsiednim stanowisku. Wskaż różnice i wyjaśnij, skąd się wzięły.
2. Za pomocą polecenia **ping** znajdź adresy IP kilku znanych ci stron internetowych. Porównaj czasy odpowiedzi ich serwerów. Doświadczenie powtórz kilka razy. Sprawdź, czy czasy odpowiedzi dla tych samych adresów się różnią. Jeśli tak, zastanów się nad przyczyną i zweryfikuj swoje przypuszczenia w wiarygodnych źródłach.
3. Odnajdź w swoim smartfonie lub tablecie dane karty sieciowej WiFi. Zanotuj jej MAC adres. Ponieważ jest on niepowtarzalny i na stałe wpisany w jej pamięć, może posłużyć do identyfikacji telefonu.
- 4*. Użyj opcji pomocy Wiersza polecenia dla polecenia **ipconfig**. Użyj argumentu **/?** i odczytaj komunikaty. Na podstawie dostępnych źródeł, np. wiarygodnych stron internetowych lub książek, ustal, jakie znaczenie mają poszczególne argumenty tego polecenia.
- 5*. Skorzystaj z opcji pomocy Wiersza polecenia dla polecenia **ping**. Użyj argumentu **/?** i odczytaj komunikaty. Korzystając z dostępnych źródeł, np. wiarygodnych stron internetowych lub książek, ustal, jakie znaczenie mają poszczególne argumenty tego polecenia.

PODSUMOWANIE LEKCJI

- s. 218 • Konfigurację połączenia i dane karty sieciowej poznasz, używając polecenia **ipconfig/all**.
- s. 219 • Za pomocą polecenia **ping** sprawdzisz połączenie pomiędzy komputerami w sieci lokalnej, używając ich adresów IP, a także poprawność połączenia z internetem, używając np. IP lub domeny znanej ci strony.
- s. 221 • Gdy zajdzie taka potrzeba, np. dla kamery sieciowej, która powinna mieć stałe IP w sieci, możesz **skonfigurować kartę sieciową**, wpisując ręcznie parametry sieci i adres IP.



30. Kupujemy świadomie, czyli poznajemy parametry urządzeń peryferyjnych

NA TEJ LEKCJI:

- poznasz parametry urządzeń peryferyjnych;
- dowiesz się, na co zwracać uwagę przy kupnie urządzenia.

Na półkach marketów i witrynach sklepów internetowych sprzedawcy oferują mnóstwo urządzeń peryferyjnych. Które wybrać? Jak kupić sprzęt dopasowany do naszych potrzeb? Na co zwracać uwagę? Czy droższe zawsze oznacza lepsze? Aby dokonać właściwego wyboru, musisz poznać podstawowe parametry, które decydują o przeznaczeniu i jakości sprzętu. Zapoznaj się z nimi i samodzielnie oceń wartość sprzętu.

30.1. DPI, czyli parametry drukarek

Czy zastanawiałeś/zastanawiałaś się kiedyś, co oznaczają skróty w nazwie danego parametru urządzenia peryferyjnego? Czym na przykład różni się PPI od DPI? Zacznijmy od nich.

Zasadniczą różnicą między tymi parametrami jest to, że DPI dotyczy obrazu wydrukowanego, a PPI – wyświetlanego na ekranie.

Wartość podana jako **DPI** (Dots Per Inch, czytaj: dots per incz) oznacza liczbę wydrukowanych punktów na długości jednego cala (2,54 cm). Ma to bezpośrednie przełożenie także na wielkość wydrukowanego obrazka. Przykładowo grafika przygotowana w edytorze zbudowana z 6000 punktów w pionie i 3000 punktów w poziomie wydrukowana w rozdzielczości 300 DPI ma wymiary 20 na 10 cali. Istnieje wiele programów pomagających przeliczać wartości związane z wydrukiem, w tym także on-line (rys. 30.1.). Ta sama grafika drukowana z rozdzielczością 150 DPI zajmie na wydruku obszar 40 na 20 cali. Z tego wniosek, że dla danego obrazu zapisanego w pliku uzyska się różne rozmiary wydruku w zależności od liczby DPI ustawionej dla drukarki. Wartość ta służy więc do określania rozmiaru obrazu na wydruku względem faktycznych wymiarów obrazu liczonych liczbą punktów powstałych podczas tworzenia danej grafiki.

DPI



Online Pixel DPI Calculator Converter Conversion -- PPI calculate

Online Pixel DPI Calculator Converter Conversion -- PPI calculate

Unit: Select the size (examples)

20 inc	6000 pix	20 inc
10 inc	3000 pix	3000 pix
300 dpi	300 dpi	
= 6000 x 3000 pix	= 20 x 10 inc	= 300 dpi
calculate	calculate	calculate

Rys. 30.1. Kalkulator on-line wspomagający obliczanie wielkości wydruku w zależności od wartości PPI

Czym w takim razie jest **PPI** (Pixels Per Inch, czytaj: piksels per incz)? Szybko to zrozumiesz, jeśli posłużysz się edytorem grafiki rastrowej, na przykład GIMP (rys. 30.2.). Ustawienia nowego projektu zawierają podstawowe jego parametry, w tym także wielkość w pikselach i rozdzielczość PPI.

Diagram illustrating the relationship between GIMP project settings and the online calculator results.

GIMP Project Settings (Left):

- Wymiary obrazu: Szerokość: 640, Wysokość: 400
- Zaawansowane opcje: Rozdzielczość: 72 DPI, Jednostka rozdzielczości: piksele na cal

Annotations:

- Projekt w skali 1:1 zajmie na ekranie obszar 640 na 400 pikseli.
- Rozdzielczość w obu osiach ustawiono na 72 piksele na cal, czyli 72 PPI.

Online Calculator Results (Right):

Unit: Select the size (examples)

226 mm	141 mm	72 dpi
141 mm	72 mm	400 dpi
72 mm	72 mm	300 dpi
= 226 x 141 mm	= 226 x 141 mm	= 72 dpi
calculate	calculate	calculate

Kalkulator wyliczył, że taki projekt po wydrukowaniu z rozdzielczością 72 DPI będzie miał wymiary 226 mm na 141 mm, a dla 300 DPI 54 na 34 milimetry.

Rys. 30.2. Wpływ ustawienia parametrów projektu graficznego na wielkość wydruku

cechy i parametry

Domyślasz się, że znajomość tych parametrów pomoże w wyborze drukarki (DPI) i monitora (PPI). Ale nie tylko one. Zanim podejmiesz decyzję o zakupie, odwiedź stronę producenta i pobierz pełną instrukcję obsługi, w której



znajdziesz także komplet parametrów. Nie opieraj swoich sądów na informacjach reklamowych. Na co zwrócić uwagę?

- **Technologia druku** – w powszechnym, domowym i biurowym użytkowaniu wykorzystuje się drukarki laserowe i atramentowe. Na podstawie wydajności i ceny materiałów eksploatacyjnych, tonera lub atramentu można wyliczyć cenę wydruku jednej strony. Technologia druku znacznie wpływa na pozostałe parametry.
- **Rozdzielczość** – ten parametr określa liczbę DPI. W przypadku drukarek laserowych często spotykanymi wartościami są 300, 600 lub 1200. 600 jest wystarczająca do uzyskania dobrej jakości druku i dlatego jest najczęściej używana. Większość drukarek atramentowych drukujących barwnie dla domowego lub biurowego użytku ma rozdzielczość na przykład 1200 x 4800 dpi. Tak duża liczba punktów wynika z tego, że druk powstaje z kropli emitowanych przez wiele dysz dla każdego z kolorów oddzielnie.
- **Szybkość druku** – jest ważna, gdy drukarka ma być użytkowana do druku dokumentów o wielu stronach. Dla drukarek laserowych może sięgać ponad 35 stron na minutę, co daje 1 kartkę co około 2 sekundy. Drukarki atramentowe tej samej klasy drukują wolniej. W czerni osiągają prędkość do 20 stron na minutę. Wydruk kolorowy trwa około dwukrotnie dłużej. Oczywiście dane te mogą być różne w zależności od modelu i producenta drukarki.
- **Format papieru** – najczęściej spotykanym formatem papieru, na którym można drukować dokumenty, jest A4. Większość drukarek drukuje także na formatach mniejszych, na przykład na różnej wielkości kopertach. Niektóre drukarki atramentowe drukują też na płytach CD lub DVD. Z formatem wiąże się dopuszczalna gramatura papieru, czyli jego waga na 1 metr kwadratowy. Typowa kartka do drukowania dokumentów ma gramaturę 80 gramów. Każda drukarka ma ponadto określoną maksymalną liczbę kartek, jaka mieści się w zasobniku, przykładowo 250 sztuk (pół ryzy). Jeśli planuje się drukowanie na grubszym papierze, należy zwrócić uwagę, czy drukarka ma możliwość podawania pojedynczych arkuszy.
- **Obciążalność** – ten parametr jest bardzo ważny, gdy drukarka pracuje w miejscu, w którym wykonuje się wiele wydruków w miesiącu. Drukarki laserowe przewyższają pod tym względem atramentowe. Ich dopuszczalna miesięczna liczba wydrukowanych stron może sięgać nawet do kilkudziesięciu tysięcy. Dla drukarek atramentowych liczba ta sięga kilkunastu tysięcy.
- **Sposób transmisji danych** – każda drukarka domowa i biurowa łączy się z komputerem za pośrednictwem magistrali USB. Wiele modeli ma także wbudowaną kartę sieciową. Dzięki temu staje się ona dostępna dla wielu użytkowników bez konieczności przenoszenia drukarki i przetaczania przewodów. Jeśli drukarka ma pracować w sieci firmowej, należy sprawdzić dostępne protokoły transmisji danych.



- **Koszt wydruku** – tego parametru nie podają oficjalne opisy drukarek. Producent materiałów eksploatacyjnych (tonerów lub atramentów) podaje średnią liczbę stron, jaką można wydrukować z użyciem danego materiału, a także normy określające poziom wydajności tych materiałów (laserowe – ISO/IEC 19752, atramentowe – ISO/IEC 24711). Ustalono, że takie dane określa się dla pięcioprocentowego pokrycia strony drukiem dla tekstu i dwudziestoprocentowego – dla grafiki (dla każdego koloru).

Technologia druku	Laserowa, monochromatyczna
Maksymalny format nośnika	A4
Podajnik papieru	250 arkuszy
Rodzaje podajników papieru	Kasetowy
Odbiornik papieru	150 arkuszy
Szybkość druku w mono	28 str./min
Maksymalna rozdzielczość druku	1200 x 1200 dpi
Miesięczne obciążenie	30000 str./miesiąc
Maksymalna gramatura papieru	163 g/m ²
Druk dwustronny (dupleks)	Automatyczny
Interfejsy	USB Wi-Fi LAN (Ethernet)
Wyświetlacz	Brak
Szerokość	371 mm
Wysokość	360 mm
Głębokość	625 mm
Waga	6,9 kg
Dołączone akcesoria	Kabel zasilający Kabel USB Toner startowy

Rys. 30.3. Przykładowe parametry biurowej drukarki laserowej

Przeznaczenie produktu	Dom Małe biuro
Technologia druku	Atramentowa, kolorowa
Maksymalny format nośnika	A3
Podajnik papieru	500 arkuszy
Rodzaje podajników papieru	Kasetowy + tacka
Szybkość druku w kolorze	10 str./min
Szybkość druku w mono	18 str./min
Maksymalna rozdzielczość druku	4800 x 2400 dpi
Miesięczne obciążenie	20000 str./miesiąc
Maksymalna gramatura papieru	256 g/m ²
Druk dwustronny (dupleks)	Automatyczny
Interfejsy	USB Wi-Fi LAN (Ethernet) NFC
Wyświetlacz	Wbudowany
Szerokość	567 mm
Wysokość	304 mm
Głębokość	424 mm
Waga	15,5 kg

Rys. 30.4. Przykładowe parametry biurowej drukarki laserowej formatu A3

30.2. 4:3 czy 16:10, czyli parametry monitorów

wybór monitora

Zanim wybierzesz monitor, musisz zapoznać się z kilkoma parametrami. Podobnie jak w przypadku drukarek ważne jest przeznaczenie urządzenia i oczekiwania użytkownika. Zaczniemy od rozdzielczości.

Rozdzielczość monitora określa się maksymalną liczbą pikseli matrycy w poziomie i w pionie, na przykład 1920 na 1080 (full HD). Rozdzielczość ekranu determinuje proporcje obrazu. Dla rozdzielczości określanej jako pełne HD (High Definition, czytaj: haj definyshyn) to 16:9, co łatwo wyliczyć, dzieląc 1920 przez 1080. Im większa rozdzielczość, tym więcej szczegółów na ekranie.

Dobór proporcji ekranu zależy od upodobań użytkownika i właściwości oprogramowania, którym się posługuje. Gracz wybierze zapewne monitor panoramiczny, na przykład 22:9 z jak największą rozdzielczością, użytkownikowi korzystającemu z arkuszy kalkulacyjnych lub edytorów tekstu zapewne wystarczy 16:9.



Rys. 30.5. Porównanie wielkości obrazu w zależności od rozdzielczości

Przekątna ekranu, podobnie jak w przypadku telewizorów, jest podawana w calach. Wartość ta powinna być brana pod uwagę w ścisłym połączeniu z proporcjami ekranu.

Rodzaj matrycy określa technologię, w jakiej została ona wykonana. Wszystkie obecnie stosowane matryce są podświetlane diodami LED i wykonane na bazie ciekłych kryształów.

Czas reakcji decyduje o jakości obrazu (efekt smużenia) w czasie wyświetlania ruchomych obrazów, na przykład w grach. Jego wartość zależy od technologii użytej do budowy matrycy ekranowej. Według norm czas reakcji to czas, jaki upłynął w czasie przejścia pojedynczego piksela ze stanu całkowitego zgaszenia (czarny) poprzez całkowicie rozjaśniony do ponownego wygaszenia. Jednak w niektórych monitorach czas reakcji mierzy się według innych norm. Czas od 1 ms do 5 ms można uznać za wystarczający, by nie odczuwać efektu smużenia.

Kąt widzenia ma znaczący wpływ na wrażenia wizualne. Zbyt mały może spowodować zmianę kolorów i jasności podczas obserwowania obrazu z pozycji innej niż prostopadła do ekranu. Widać to szczególnie w osi y. Kąt ten mierzy się dla osi x i y.

Matryca TN (Twisted Nematic, czytaj: twisted nematic) jest szybka, na przykład czas reakcji może osiągnąć 1 ms, jednak nie zapewnia szerokich kątów widzenia. Można to zauważyć zwłaszcza na osi y.

Matryca VA charakteryzuje się podobną szybkością jak TN, jednak znacznie lepszymi kątami widzenia.

Matryca IPS (in-plane switching, czytaj: inplane sitching) zapewnia duże, zbliżone do 180° kąty widzenia w pionie i poziomie, dużą liczbę kolorów (do ponad miliarda), wysoki kontrast i czas reakcji kilku milisekund.

**Uwaga!**

Producenci nieustannie pracują nad nowymi technologiami wyświetlania obrazu w monitorach. Zazwyczaj są one najpierw wykorzystywane w produkcji telewizorów. Ekrany QLED, OLED lub te, które jeszcze do produkcji nie weszły, mogą niebawem trafić także do monitorów.

Częstotliwość odświeżania to liczba klatek (obrazów) wyświetlanych w ciągu sekundy. Standardem jest 60 Hz (60 na sekundę). Niektóre monitory mogą wyświetlać obrazy z wyższą częstotliwością, na przykład 75 Hz lub większą.



Rys. 30.6. Wejścia sygnały wideo w monitorze komputerowym

Interfejsy wejściowe (rys. 30.6.) muszą być zgodne z wyjściami karty grafiki komputera. Coraz rzadziej spotyka się wejście analogowe VGA. Wśród cyfrowych najpopularniejszy jest HDMI (używany także w domowym sprzęcie telewizyjnym). Oprócz niego można spotkać DVI i DisplayPort. Ten ostatni pozwala na dołączenie do karty grafiki (jeśli ma odpowiednią liczbę portów) kilku monitorów jednocześnie.

Inne parametry są w większości zależne od producentów i zazwyczaj mają poprawić jakość pracy przy monitorze. Redukcja migotania ekranu (Flicker free, czytaj: fliker fri) czy regulacje kolorów mogą znacznie ograniczyć negatywny wpływ wyświetlanego obrazu na wzrok.

Specyfikacja	
Przekątna ekranu	27"
Powłoka matrycy	Matowa
Rodzaj matrycy	LED IPS
Rozdzielczość ekranu	2560 x 1440 (WQHD)
Format ekranu	16:9
Częstotliwość odświeżania ekranu	60 Hz
Technologia ochrony oczu	Redukcja migotania (Flicker free)
Wielkość pikseli	0,233 mm
Jasność	350 cd/m ²
Kontrast statyczny	1 000:1
Kontrast dynamiczny	3 000 000:1
Kąt widzenia w poziomie	178 stopni
Kąt widzenia w pionie	178 stopni
Czas reakcji	4 ms
Liczba wyświetlanych kolorów	1,07 mld
Rodzaje wejść / wyjść	HDMI - 1 szt. DisplayPort - 1 szt. DC in (wejście zasilania) - 1 szt.

Rys. 30.7. Przykładowe parametry monitora komputerowego



Czy wiesz, czym jest **rozdzielczość natywna** monitora? To rozdzielczość wyświetlacza określona fizycznie występującą liczbą pikseli w osi pionowej i poziomej. Ustawiając w sterowniku karty grafiki rozdzielczość natywną, uzyskuje się najlepszy efekt wyświetlania obrazu.

30.3. Z papieru na dysk bez problemów, czyli jak wybierać skaner

Wiesz już, jak działają skanery i do czego służą. Podobnie jak w przypadku drukarek i monitorów przed wyborem odpowiedniego sprzętu należy przeanalizować potrzeby użytkownika i wymagania oprogramowania.

Od domowych skanerów wymaga się prostoty obsługi, niewielkich rozmiarów i rozdzielczości wystarczającej do otrzymania dobrych cyfrowych kopii fotografii. Najczęściej w takich przypadkach wybiera się tańsze skanery CIS. Przeanalizuj specyfikację przykładowego skanera (rys. 30.8.). Przykładowe urządzenie zgodnie ze specyfikacją ma bardzo dobre parametry. Jednak tabelka z danymi nie przedstawia wszystkich zalet tego skanera. Jedną z nich jest to, że model nie wymaga dodatkowego zasilacza, a energię czerpie ze złącza USB. Inne cechy można odczytać z oprogramowania dołączonego do urządzenia. W wielu skanerach możliwe jest skanowanie uruchamiane wciśnięciem klawisza w obudowie, automatyczne tworzenie wielostronicowych dokumentów PDF, automatyczna korekta, a także uzyskanie osobnych plików dla zdjęć zeskanowanych w czasie jednego przebiegu. Warto więc wczytać się w opis udostępniany przez producenta lub sprzedawcę.

Specyfikacja

Typ skanera	Płaski	
Typ sensora	CIS	
Interfejs	USB	
Rozdzielczość optyczna	4800 x 4800 dpi	← Wysoka rozdzielczość.
Prędkość skanowania w kolorze	6 str./min	
Prędkość skanowania w czerni	10 s	← Dokument A4 z rozdzielczością 300 dpi zostanie zeskanowany w czasie 10 s.
Głębokość koloru (wejście)	48-bitowa	
Głębokość koloru (wyjście)	48-bitowa	
Skala szarości (wejście)	16 bitów	← Każdy z kolorów RGB jest skanowany i zapisywany z rozdzielczością 16 bitów.
Skala szarości (wyjście)	8 bitów	
Obsługiwane nośniki	A4 A5 A6 B5 B6 Pocztówki Wizytówki	← Można skanować dokumenty o dowolnych wymiarach równych lub mniejszych od A4.

Rys. 30.8. Parametry przykładowego skanera CIS



Powyższe rozważania dotyczyły głównie skanerów płaskich. Istnieją także skanery ręczne, które można zabrać chociażby do biblioteki i skanować bez udziału komputera. Niektóre skanują jednocześnie całą powierzchnię, na przykład albumu A4, inne pojedyncze wiersze tekstu, co w połączeniu z programem OCR pozwala zamienić obraz na tekst edytowalny w komputerze.

ZADANIA DO ROZWIĄZANIA

1. Określ rozmiary rysunku wydrukowanego z rozdzielczością 300 dpi, jeśli jego rzeczywisty rozmiar to 6000 na 1500 punktów. Jaką rozdzielczość dpi należy dobrać, by wydruk miał rozmiar poziomy 10 cali? Wyniki swoich obliczeń sprawdź w internecie (poszukaj strony z kalkulatorem rozdzielczości).
2. Ustal, dla jakiej rozdzielczości pionowej proporcje ekranu będą równe 16:10, jeśli pozioma jest równa 1920 pikseli.
3. Oblicz, dla jakiej rozdzielczości poziomej proporcje ekranu będą równe 4:3, jeśli pionowa jest równa 768 pikseli.
4. Załóż, że do realizacji swojego hobby potrzebujesz monitora o minimalnej wielkości ekranu 27 cali i rozdzielczości WQHD. Znajdź najkorzystniejszą ofertę w sklepach internetowych. Uzasadnij swój wybór, biorąc pod uwagę wiarygodność sprzedawcy (sprawdź opinie), cenę i sposób płatności oraz dostawy.
5. Oblicz liczbę kolorów, jaką można otrzymać, skanując dokument z 48-bitową głębią kolorów. Oceń wynik i określ, jaki ma to wpływ na jakość skanowanych zdjęć.
6. Ile odcieni szarości można uzyskać, skanując dokument z 16-bitową głębią? Czy zapisanie go po przetworzeniu na 8-bitową głębię pogorszy jakość powstałego obrazu?

PODSUMOWANIE LEKCJI

- s. 223 • Wartość podana jako **DPI** oznacza liczbę wydrukowanych punktów na długości jednego cala.
- s. 224 • **Przy wyborze urządzenia** zwróć uwagę przede wszystkim na jego istotne cechy i parametry. Następnie oceń dodatkowe zalety, np. kolor, kształt, dołączone oprogramowanie. Znajomość **parametrów urządzeń peryferyjnych** pozwala na dokonanie świadomego wyboru urządzenia odpowiedniego do potrzeb i preferencji użytkownika. **Dane techniczne** można znaleźć na stronach producentów i sprzedawców sprzętu.
- s. 226 • Zanim dokonasz **wyboru monitora** sprawdź, czy twoja karta grafiki ma **wymagany interfejs** i czy potrafi wygenerować obraz o natywnej rozdzielczości monitora.



PODSUMOWANIE PRZED SPRAWDZIANEM

27. Nie wszystko jest takie oczywiste, czyli jak działa internet

- **IP, TCP, HTTP, FTP i SMTP** to protokoły, bez których sieci i internet nie mogłyby funkcjonować. s. 207
- **Adres symboliczny**, np. strony internetowej, w rzeczywistości służy jedynie wygodzie użytkowników. W sieci funkcjonuje jako adres **IP**. Za ich zamianę odpowiadają **DNS**. s. 211



28. Sieć to nie tylko internet, czyli poznajemy topologię sieci komputerowych

- Połączenie twojego **telefonu** z komputerem za pośrednictwem **Bluetooth** jest siecią typu **P2P**. s. 213
- Najpopularniejsza topologia sieci to **topologia gwiazdy**. s. 214

29. Kto tam, czyli identyfikujemy komputery w sieci

- **Poprawność połączenia** sieciowego można sprawdzić poleceniem **ping**. s. 219
- **ipconfig/all** to polecenie systemowe wyświetlające stan sieci i karty sieciowej. s. 218
- **Konfigurację połączenia sieciowego** w Windows wykonuje się w opcji **Ustawienia**. s. 221

30. Kupujemy świadomie, czyli poznajemy parametry urządzeń peryferyjnych

- Wybór **drukarki** zależy od wielu czynników, np. **rozdzielczości, techniki druku**, szybkości. s. 223
- **TN, VA, IPS** to różne rodzaje wyświetlaczy LCD monitorów komputerowych. s. 226
- **VGA, HDMI, DISPLAYPORT, DVI** oznaczają podstawowe rodzaje złączy dla **monitorów**. s. 228
- Trafny wybór odpowiedniego urządzenia peryferyjnego zależy od znajomości jego parametrów. s. 224