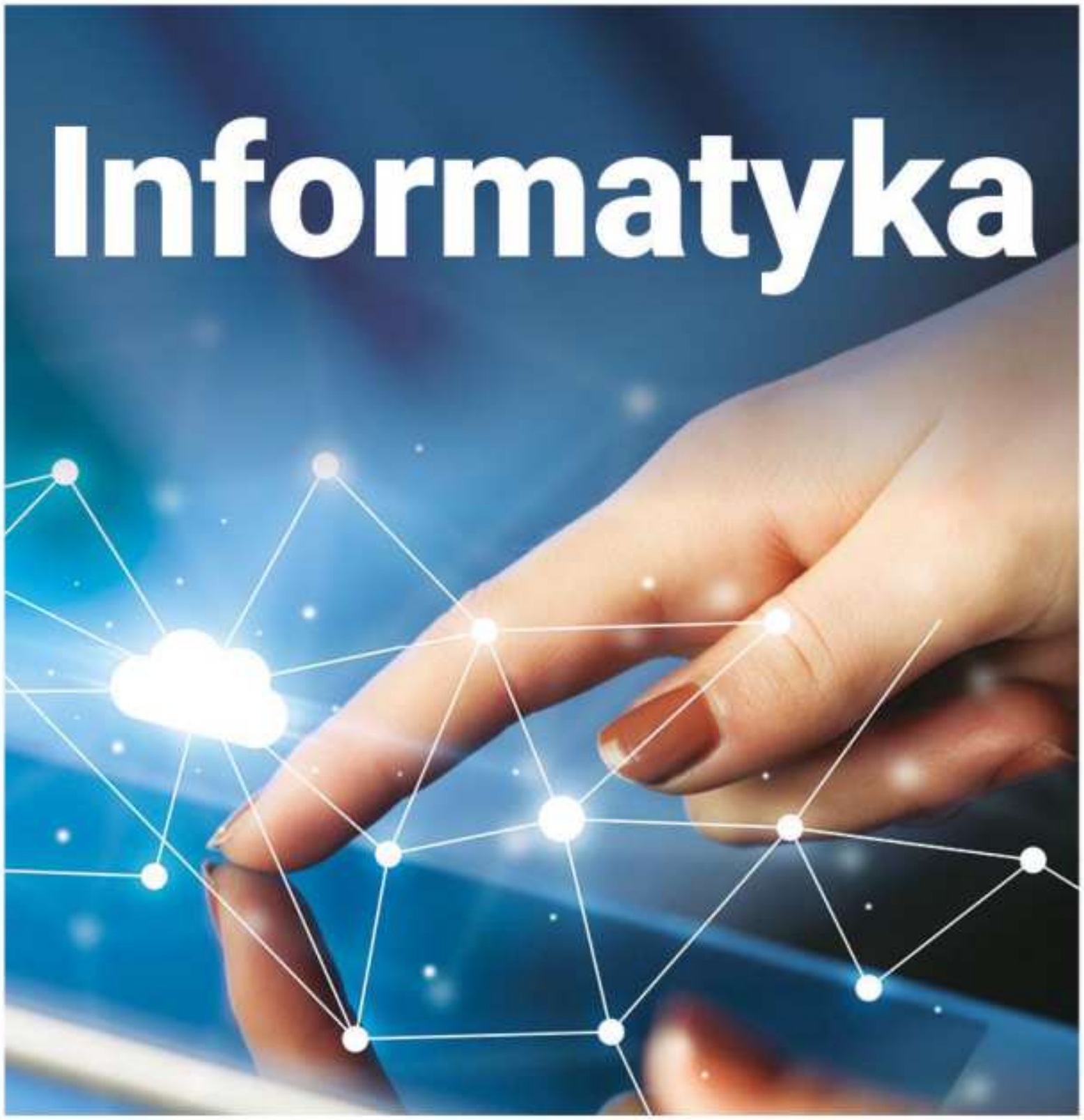
OPERON
Edukacja jest podróżą

ZAKRES ROZSZERZONY

Informatyka





2

Podręcznik dla **szkół ponadpodstawowych**

Wojciech Hermanowski
Sławomir Sidor

Informatyka

Podręcznik dla **szkół ponadpodstawowych**

2

ZAKRES ROZSZERZONY

OPERON
Edukacja jest podróżą

2024

Projekt okładki: Paweł Kwacz
Redaktor prowadzący: Sebastian Przybyszewski
Redakcja językowa: zespół
Redakcja graficzna i skład: zespół
Korekta: Eliza Perkowska
Fotoedycja: Marta Jasińska
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 rozszerzonym na poziomie szkoły ponadpodstawowej, na podstawie opinii rzeczoznawców: dr. Tomasza Huka, dr. Przemysława Ogonowskiego, mgr Teresy Kosyry-Cieślak.

Numer dopuszczenia: 1048/2/2020

Etap edukacyjny: III
Typ szkoły: szkoła ponadpodstawowa
Rok dopuszczenia: 2020

© Copyright by OPERON Sp. z o.o. & Wojciech Hermanowski, Sławomir Sidor
Gdynia 2020
Wszelkie prawa zastrzeżone. Kopiowanie w całości lub we fragmentach bez zgody wydawcy zabronione.
N7195

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 978-83-66365-05-6

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

Wstęp 5

I. Sieci komputerowe

1. Host i serwer, czyli o działaniu sieci 8
2. Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej 15
3. Nie tylko Wi-Fi, czyli bezprzewodowy dostęp do internetu 20

Podsumowanie przed sprawdzianem 24

II. Opracowania rozwiązań problemów za pomocą wybranych aplikacji

4. Zespół, czyli realizujemy projekty w chmurze 26
5. Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi 32
6. Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu 36
7. Broszura, czyli jak projektować duże dokumenty 42
8. Recenzja, czyli proponujemy poprawki w tekście 50
9. OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu 55

Podsumowanie przed sprawdzianem 60

III. Rozwijanie kompetencji społecznych

10. Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony 62
11. Platformy uczą, czyli rola e-learningu w nauce i pracy 69
12. Możesz być administratorem, czyli jak zarządzać platformą e-learningową 75
13. Zasoby i testy, czyli wypełniamy kursy treścią 80
14. Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach do e-learningu 86

Podsumowanie przed sprawdzianem 92

IV. Bazy danych

15. Jak gromadzić informacje, czyli komputerowe bazy danych 94
16. Tabele i formularze, czyli jak utworzyć bazę danych 97
17. Relacje i pytania, czyli jak uczynić bazę użyteczną 103
18. Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access 107
19. Formularze, czyli ułatwiamy sobie wprowadzanie informacji do bazy danych 115
20. Czy to możliwe, czyli baza danych w arkuszu Excel 127
21. Wiedza w sieci, czyli internet mądrych ludzi 134
22. Wirtualne serwery, czyli instalujemy Apache 145
23. Język zapytań, czyli poznajemy język SQL 150
24. Zadajemy pytania, czyli kwerendy w języku SQL 153
25. Więcej pytań, czyli jeszcze o kwerendach w języku SQL 162
26. Podzapytania, czyli instrukcje modyfikujące dane w języku SQL 169
27. Modyfikacje, czyli zarządzanie bazą danych 174
28. Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe 179

Podsumowanie przed sprawdzianem 183

V. Programowanie stron internetowych

29. HTML, czyli przeglądarka interpretuje język programowania stron 188
30. Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony 193
31. Grafika, czyli kolejna składowa stron internetowych w HTML 200
32. Składnia stylów, czyli jak CSS pomaga w programowaniu strony 205
33. Pliki stylów, czyli CSS w akcji 210
34. Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej 219
35. Widoczna w internecie, czyli jak opublikować stronę 225
36. CMS, czyli system zarządzania treścią strony internetowej 231
37. Panel i skórki, czyli tworzymy stronę w CMS 236
38. Szybko i łatwo, czyli programy do tworzenia stron internetowych 245
39. Pierwsze skrypty, czyli poznajemy język PHP 255
40. Stałe i zmienne, czyli typy danych w języku PHP 258
41. Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP 262
42. Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP 265
43. Funkcje w języku PHP, czyli jak wzbogacić naszą stronę 268
44. Tablice, czyli jak odzyskać dane w języku PHP 271
45. Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP 275

Podsumowanie przed sprawdzianem 277

VI. Algorytmy i programowanie

46. Powtarzanie pętli i wywoływanie siebie, czyli iteracja i rekurencja w algorytmach 282
47. Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa ... 287
48. Przez wstawianie, czyli jeszcze o porządkowaniu liczb 293
49. Komputer porządkuje liczby, czyli układamy programy sortujące 297
50. Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu 302
51. Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania 306
52. Przecinając oś, czyli wyznaczanie miejsc zerowych funkcji metodą połowienia 309
53. Prawie robi różnicę, czyli obliczanie przybliżonej wartości pierwiastka kwadratowego ... 311
54. Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym 313
55. Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak inaczej obliczyć potęgę o wykładniku naturalnym 316
56. Suma binarna, czyli dodawanie w systemie binarnym 319
57. Różnica binarna, czyli odejmowanie w systemie binarnym 321
58. Iloczyn binarny, czyli mnożenie w systemie binarnym 324
59. Iloraz binarny, czyli dzielenie w systemie binarnym 326
60. Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby pi 328

Podsumowanie przed sprawdzianem 331

Wstęp

Jak jest zbudowany nasz podręcznik

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



Każdy dział podręcznika otwiera spis tematów z danego obszaru.



29. HTML, czyli przeglądarka interpretuje język programowania stron

NA TEJ LEKCJI:

- dowiesz się, czym jest HTML i CSS,
- poznasz podstawowe znaczniki i budowę dokumentu HTML,
- zrozumiesz, jakie znaczenie dla wyglądu strony ma CSS.

HTML i CSS. Każdy słyszał o nich słyszał, ale czy wiesz, czym tak naprawdę są? Czy to języki programowania, a jeśli tak, to jak uruchamia się programy? A może jest inaczej, ale co się wydaje? Ten rozdział rozwieje twoje wątpliwości.

29.1. Co interpretuje przeglądarka, czyli czym jest dokument HTML

Nasze rozważania zaczniemy od definicji.

HTML – HTML (ang. Hypertext Markup Language) to hipertekstowy język znaczników. Nie jest językiem programowania, ponieważ nie układa się w nim programów komputerowych. Służy do opisu stron internetowych za pomocą znaczników i tekstu. Przeglądarka interpretuje język znaczników i na tej podstawie wyświetla stronę.

Pierwsze wątpliwości rozwiane. HTML nie jest językiem programowania, chociaż często w języku potocznym tak się o nim mówi. W uproszczeniu można powiedzieć, że opisuje on za pomocą tekstu wygląd i zawartość stron internetowych. Często mówi się także, że ten tekst to program stron. Stąd nieporozumienia związane z interpretacją nazwy.

znaczniki – Skoro jest to język znaczników, warto wiedzieć, czym one są. W takcie tekst je odnaleźć, bo wpisuje się je pomiędzy znaki < >. Rozróżniamy dwa ich rodzaje: **znaczniki otwierające** i **znaczniki zamykające** . Jeśli porównasz użyć którego z nich, np. w odniesieniu do tekstu, to przed nim umieścisz odpowiedni znacznik w < >, a na jego końcu w </ >. Dzięki znacznikom można określać sposób wyświetlania obiektów na stronie.

Przykład

Tekst który ma się pojawić pogrubiony. Należy więc go umieścić pomiędzy znacznikami: tekst. Inne znaczniki poznasz w dalszej części podręcznika.

W nazwie HTML występuje też słowo hipertekstowy. Dlaczego? Zapewne wiesz, czym są linki (odsyłacze) na stronach internetowych. To, że dzięki, jest

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

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

Przystępny język publikacji ułatwi wam zrozumienie treści.

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

Schematy i rysunki pomogą wam usystematyzować wiedzę.

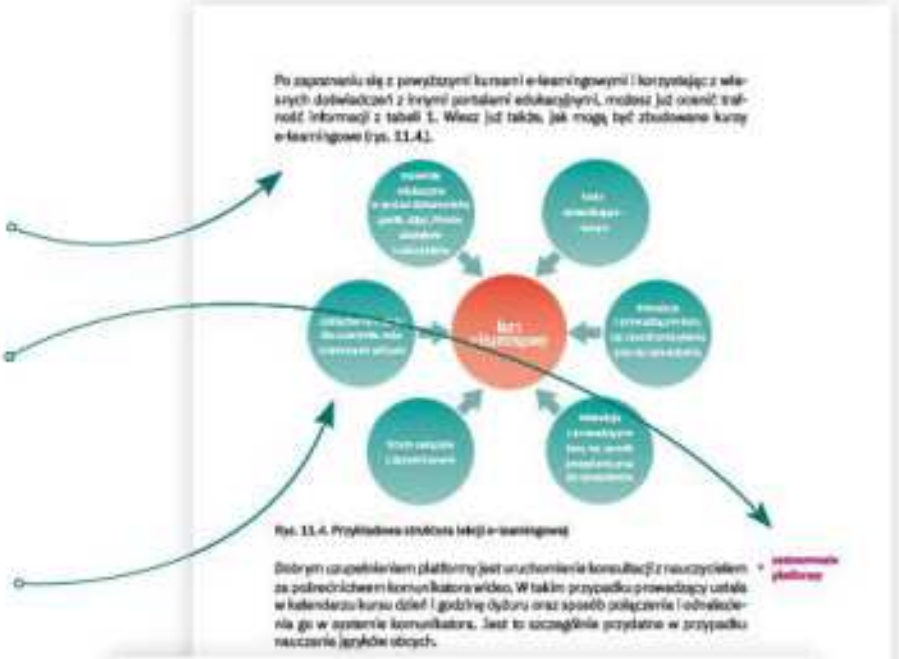
Zrzuty ekranu pełnią funkcję tutoriali (samouczków), dzięki którym będziecie je mogli szybko zastosować w praktyce.

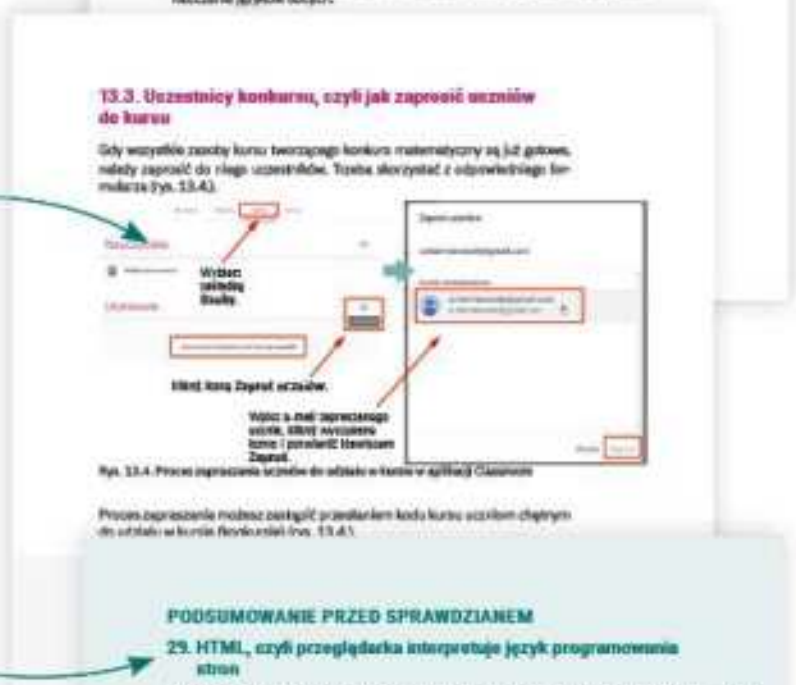
Podsumowanie przed sprawdzaniem zawiera najważniejsze treści do zapamiętania z działu. Jeśli ich nie pamiętacie, wróćcie na stronę wskazaną obok i poszukajcie hasła na marginesie.

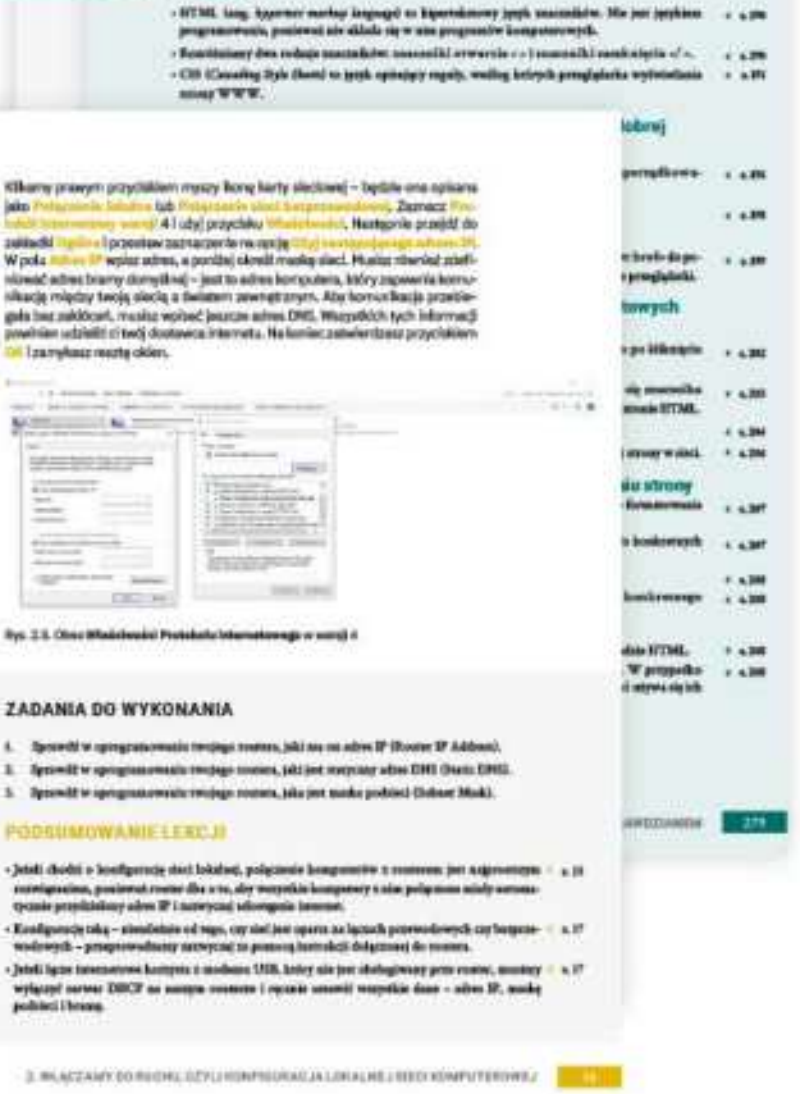
UWAGA! W podręczniku nie należy umieszczać pisemnych odpowiedzi do zadań.

Zadania do wykonania przećwiczycie w toku lekcji lub wykorzystacie do nauki w domu albo przed kartkówką czy sprawdzianem. Gwiazdką oznaczono trudniejsze zadania.

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. Sieci komputerowe

Host i serwer, czyli o działaniu sieci

Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej

Nie tylko Wi-Fi, czyli bezprzewodowy dostęp do internetu

1. Host i serwer, czyli o działaniu sieci

NA TEJ LEKCJI:

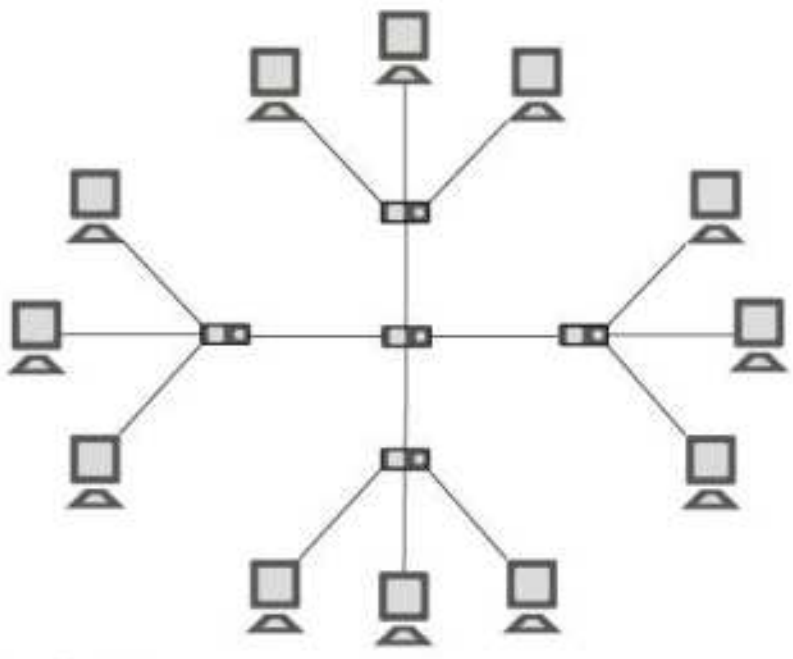
- dowiesz się, co to jest sieć komputerowa;
- zapoznasz się z urządzeniami potrzebnymi do jej funkcjonowania;
- dowiesz się, czym są media transmisyjne i jakie są ich rodzaje;
- zapoznasz się z układem i topologią sieci komputerowych.

Nie wyobrażamy sobie dzisiejszego świata bez otaczających nas zewsząd telefonów, komputerów oraz wielu innych urządzeń elektronicznych, ułatwiających życie czy umożliwiających porozumiewanie się ze sobą. Komunikujemy się za pomocą różnego rodzaju aplikacji głosowych, tekstowych czy wideo. Wysyłamy miliony wiadomości dziennie. Największą rolę w procesie komunikowania się odgrywają dzisiaj – oprócz wspomnianych urządzeń – sieci komputerowe.

1.1. Pojęcie sieci komputerowej

sieć komputerowa

Sieć komputerowa to system urządzeń, takich jak: komputery, telefony, drukarki czy telewizory, połączonych ze sobą w celu wymiany i przesyłania danych (rys. 1.1.). Odbywa się to za pomocą kanałów komunikacyjnych oraz odpowiedniego oprogramowania. Sieć komputerowa umożliwia szybkie publikowanie danych, a także otwiera techniczne możliwości tworzenia wspólnych zasobów informacji i korzystania z nich.



Rys. 1.1. Sieć komputerowa

Obecnie do sieci komputerowej podłączamy nie tylko komputery i urządzenia peryferyjne, lecz także telefony, tablety, telewizory, a nawet lodówki i inne urządzenia AGD – są to tzw. urządzenia klienckie. Oczywiście podstawy działania sieci się nie zmieniły. Różnica polega jedynie na tym, że korzysta z niej

wiele więcej urządzeń niż kiedyś. Abyśmy mogli sprawnie korzystać z sieci, potrzebne nam są tzw. urządzenia pośredniczące, takie jak: switch (czyt. stícz), router (czyt. ruter) czy access point (czyt. akses pojnt).

1.2. Najważniejsze pojęcia związane z siecią komputerową

Host – urządzenie końcowe sieci komputerowej, stanowiące źródło lub cel przesyłania danych w sieci. Hostem jest każde urządzenie w sieci, któremu przypisano adres IP.

Serwer – komputer, na którym zainstalowano specjalistyczne oprogramowanie, oferujący usługi innym komputerom (WWW, poczta elektroniczna, zasoby plikowe).

Klient – każde urządzenie korzystające z usług udostępnionych przez serwery, może to być komputer (stacjonarny lub laptop) ale również telefon, zegarek, lodówka itp.

Klient – Serwer – architektura sieci komputerowej, w której występuje komputer udostępniający usługi (serwer) oraz komputery korzystające z tych usług (klienci).

Peer to Peer (czyt. pir tu pir; P2P, czyt. pi tu pi) – architektura sieci komputerowej, w której nie występuje jeden komputer udostępniający usługi – wszystkie komputery w tej sieci działają na równorzędnych prawach.

Medium transmisyjne – element sieci komputerowej, za pomocą którego urządzenia wzajemnie się komunikują: na przykład kabel miedziany czy światłowodowy lub fale radiowe (Wi-Fi).

Protokół komunikacyjny – określony sposób (język) komunikacji, dzięki któremu jest możliwa wymiana danych pomiędzy urządzeniami w sieci.

LAN (ang. *local area network*, czyt. lokal erja netłork) – lokalna sieć komputerowa, obejmująca swoim zasięgiem pomieszczenie, piętro, budynek lub zbiór budynków.

MAN (ang. *metropolitan area network*, czyt. metropolitan erja netłork) – miejska sieć komputerowa, obejmująca swoim zasięgiem miasto lub aglomerację miejską.

WAN (ang. *wide area network*, czyt. łajd erja netłork) – rozległa sieć komputerowa, komunikująca ze sobą odległe sieci LAN.

Bezprzewodowa sieć lokalna, WLAN (od ang. *wireless local area network*, czyt. łajrles lokal erja netłork) – sieć lokalna, w której połączenia między urządzeniami sieciowymi zrealizowano bez użycia przewodów. W Polsce termin Wi-Fi (chociaż pierwotnie był nazwą tylko jednego produktu używającego określonego standardu WLAN) jest używany jako synonim określenia WLAN.

Karta sieciowa (ang. *network interface card*, czyt. netłork interfejs kard, NIC) – adapter instalowany w urządzeniach (komputerach, telefonach itp.), dzięki któremu jest możliwe podłączenie tego urządzenie do sieci komputerowej.

Ruter (ang. *router*) – urządzenie sieciowe, którego głównym zadaniem jest łączenie ze sobą różnych sieci w celu umożliwienia im wzajemnej komunikacji oraz określanie ścieżki przepływu danych pomiędzy sieciami na podstawie adresu IP (ang. *routing*, czyt. rutin).

Przełącznik (ang. *switch*) – urządzenie sieciowe, które łączy urządzenia w sieci lokalnej oraz decyduje o przesyle danych pomiędzy urządzeniami na podstawie adresu MAC; podstawowym zastosowaniem przełącznika jest kontrola dostępu do medium transmisyjnego (nośnika używanego do transmisji sygnałów telekomunikacyjnych) oraz dzielenie sieci na segmenty w celu minimalizacji ruchu.

Koncentrator sieciowy (także z ang. *hub*, czyt. hab) – urządzenie pozwalające na przyłączenie wielu urządzeń sieciowych do sieci komputerowej o topologii gwiazdy. Najczęściej spotykane w wersji 4-, 8-, 16- lub 24-portowej.

Punkt dostępu, punkt dostępowy (ang. *access point*, AP) – urządzenie zapewniające hostom dostęp do sieci komputerowej za pomocą bezprzewodowego nośnika transmisyjnego, jakim są fale radiowe. Punkt dostępowy jest zazwyczaj mostem łączącym bezprzewodową sieć lokalną (WLAN) z siecią lokalną (LAN).

1.3. Jednostki przesyłu danych w sieciach komputerowych

bit per second ▸ Wymiana danych w sieciach komputerowych odbywa się z pewną prędkością, której wartość informuje, jaka ilość danych jest przesłana w ciągu konkretnego czasu. Podstawową jednostką służącą w informatyce do zapisu pojemności danych jest 1 bit [b]. Do określania prędkości przesyłania danych zastosujemy zatem liczbę bitów przesłanych w ciągu jednej sekundy. Zatem podstawowa jednostka to bit na sekundę (b/s bądź bps – *bit per second*, czyt. bit per sekend).

Aby nie podawać zbyt dużych liczb, stosujemy ich wielokrotności. Są one wam znane między innymi z pojemności dysków – kilo, mega, tera itd.:

1 kilobit (1 Kb/s lub 1 Kbps) – 1024 b/s (2^{10})
1 megabit (1 Mb/s lub 1 Mbps) – 1024 Kb/s – 1048576 b/s ($2^{10}*2^{10}$)

Często możemy znaleźć informacje, że dana sieć ma przepustowość podawaną w megabajtach lub gigabajtach. Należałoby przypomnieć sobie, jak przeliczamy takie wielkości. 1 bajt (1 B) równa się 8 bitom (8 b), do tego dochodzą jeszcze jednostki czasu, które też przeliczamy.

Ćwiczenie 1.

Obliczamy, ile danych prześlemy w ciągu godziny, jeżeli będziemy dysponować łączem o przepustowości (czyli prędkości przesyłu) 120 Mb/s. W rzeczywistości przepustowości są dużo niższe, ponieważ wysyłane są dane nadmiarowe jak adres (gdzie dane mają być wysłane) oraz inne usługi sieciowe (np. sprawdzanie poczty w tle).

Dane:
czas: 1 godzina
przepustowość łącza: 120 Mb/s

Obliczenia:
1. 1 godzina to 3600 sekund.
2. Zamieniamy jednostki danych – 120 megabitów/s to 15 megabajtów/s (120 Mb dzielimy na 8, ponieważ 1 bajt to 8 bitów).
3. Mnożymy teraz przepustowość: 15 MB/s*3600 s = 54000 MB, czyli około 52,7 GB (ponieważ 1 GB to 1024 MB).

Odpowiedź:
W ciągu jednej godziny łączem o przepustowości 120 Mb/s jesteśmy w stanie przesłać około 52,7 GB danych.

Ćwiczenie 2.

Obliczmy teraz, w jakim czasie prześlemy dane, które na dysku zajmują 1,5 GB pojemności. Dysponujemy łączem o przepustowości 15 Mb/s.

Dane:
wielkość pliku: 1,5 GB
przepustowość łącza: 15 Mb/s

Obliczenia:
1. Zamieniamy jednostkę przesyłu danych: 15 MB/s to 1,875 MB/s (15 Mb dzielimy przez 8, ponieważ 1 bajt to 8 bitów).
2. Zamieniamy jednostkę wielkości pliku z GB na MB: 1,5 GB to 1536 MB.
3. Obliczamy czas przesyłu pliku 1536 MB/1,875 MB/s = 819,2 s, czyli około 14 minut.

Odpowiedź:
Plik o pojemności 1,5 GB będzie przesłany łączem o przepustowości 15 Mb/s przez około 14 minut.

1.4. Media transmisyjne

media transmisyjne

Aby powstał jakikolwiek ruch w sieci, to urządzenia, które pozwalają nam się komunikować, muszą być połączone ze sobą w jakiś sposób. Urządzenia sieciowe łączy się za pomocą mediów transmisyjnych przewodowych i bezprzewodowych. W tym rozdziale skupimy się przede wszystkim na mediach przewodowych. Zaliczamy do nich kable symetryczne (skrętkę), współosiowe (kabel koncentryczny), światłowodowe i energetyczne.

Kabel symetryczny, nazywany inaczej skrętką, składa się z ośmiu żył (czterech par żył). Są one ze sobą splecione parami. Każda para skrętki ma jedną żyłę do przenoszenia napięcia i drugą uziemioną. Istnieją trzy typy skrętek:

- **nieekranowana** (*unshielded twisted pair,* czyt. anszilded tłisted per, UTP), która jest stosowana w większości sieci,
- **ekranowana** (*shielded twisted pair,* czyt. szilded tłisted per, STP), która ma specjalną warstwę (ekran) chroniącą ją przed zakłóceniami elekromagnetycznymi,
- **foliowana** (*foiled twisted pair,* czyt. fojld tłisted per FTP), która jest okręcona metalową tasiemką.

Skrętka jest stosowana do przesyłania danych zarówno analogowo, jak i cyfrowo. Ze względu na przydatność do transmisji danych wyróżnia się następujące kategorie skrętek:

- CAT 3 – przepustowość 10 Mb/s,
- CAT 5 – przepustowość do 100 Mb/s,
- CAT 5e – skrętka kategorii 5, tak jak CAT 5, ale z poprawioną jakością transmisji; jej przepustowość wynosi również do 100 Mb/s,

- CAT 6 – przepustowość do 1 Gb/s,
- CAT 7 – przepustowość powyżej 1 Gb/s,
- CAT 7a – różni się ona od poprzednich klas stosowaniem kabli typu S/FTP (każda para w ekranie oraz ekran obejmujący cztery pary) łączonych ekranowanymi złączami; dla tej klasy okablowania jest możliwa realizacja systemów transmisji danych z prędkościami przekraczającymi 1 Gb/s (40 Gb/s przy długości mniejszej niż 50 m i 100 Gb/s przy długości mniejszej niż 15 m),
- CAT 8.1 – wykorzystująca pasmo częstotliwości 1600–2000 MHz; prędkość transmisji ponad 40 Gbit/s (parametry kabli kategorii 8.1 nie są formalnie sprecyzowane),
- CAT 8.2 - wykorzystująca pasmo częstotliwości 1600–2000 MHz; prędkość transmisji ponad 40 Gbit/s.

Kabel współosiowy, nazywany również koncentrycznym, składa się z dwóch przewodników – wewnętrznego i zewnętrznego – czyli z żyły podstawowej i ekranu. Oba składniki są od siebie oddzielone warstwą izolacyjną (dielektrykiem), która ma chronić przewód przed zakłóceniami. Obecnie kabel współosiowy jest bardzo rzadko stosowany, a wcześniej był powszechnie wykorzystywany do połączeń AV, łączenia anten oraz w sieciach komputerowych. Kable współosiowe (koncentryczne) dzielimy na:

- **gruby Ethernet** (czyt. eternet) – ma grubość około 1 cm i pozwala na transmisję danych bez zakłóceń na maksymalną odległość do 500 m,
- **cienki Ethernet** – ma grubość około 0,5 cm i pozwala na transmisję danych na maksymalną odległość do 185 m.

Maksymalna prędkość przesyłu danych za pomocą kabla koncentrycznego wynosi 10 Mb/s.

Światłowód to obecnie najnowocześniejsze medium stosowane w transmisji przewodowej. Składa się z cienkiego włókna szklanego. Rdzeń kabla jest wykonany ze szkła kwarcowego i otoczony płaszczem oraz warstwą ochronną. Informacje są przesyłane za pomocą wiązki światła generowanej przez diodę elektryczną lub laser. Światłowód wyróżnia się całkowitą odpornością na zakłócenia elektromagnetyczne i uniemożliwia podsłuch transmisji.

Jeszcze do niedawna ze względu na wysoki koszt światłowód był stosowany tylko do połączeń między dostawcami internetu. Dzisiaj jest doprowadzany do indywidualnych gospodarstw domowych.

Kable energetyczne oferują najmniejszą jakość techniczną, mimo że teoretyczne są w stanie przesyłać informacje z prędkością 200 Mb/s. Niska jakość to wynik braku ochrony przed szumami zakłócającymi, które pochodzą z innych urządzeń niż nadajnik. W praktyce te kable prawie w ogóle nie są wykorzystywane, a jeżeli już, to na niewielkich odległościach.

1.5. Topologie sieci komputerowych

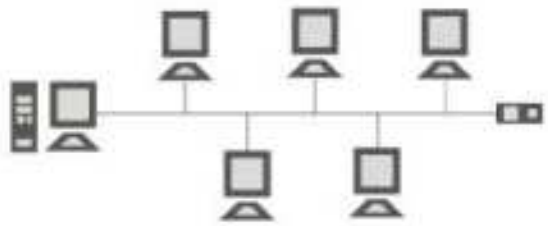
topologia sieci komputerowej

Sieć komputerowa to system połączonych ze sobą urządzeń. Połączenie stanowi swoisty model układu różnych elementów sieci. Nazywamy go topologią sieci. Topologia sieci może się odnosić do konstrukcji fizycznej i logicznej.

Topologia fizyczna sieci to fizyczny układ rozmieszczenia urządzeń w sieci oraz model połączeń między nimi.

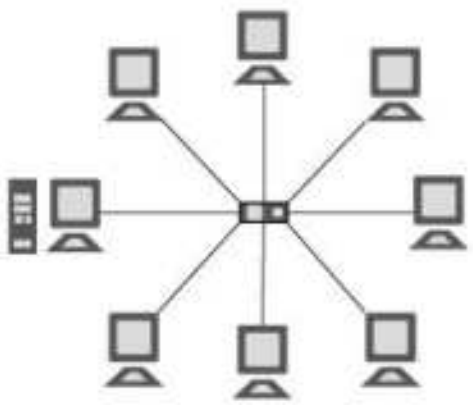
W topologii fizycznej można wyróżnić siedem modeli połączeń:

1. **Topologia magistrali** (rys. 1.2.) jest zbudowana w taki sposób, że wszystkie elementy sieci są podłączone do jednej magistrali, którą stanowi kabel koncentryczny. To jeden z najstarszych modeli połączeń sieci, dzisiaj praktycznie nie jest spotykany. Stosuje się w niej mało wydajne medium transmisyjne (maksymalna przepustowość do 10 Mb/s). Topologia ta ma miano bardzo awaryjnej, ponieważ uszkodzenie przewodu w jednym miejscu powoduje brak sieci we wszystkich urządzeniach do niej podłączonych. Zastosowanie topologii gwiazdy rozwiązuje ten problem.



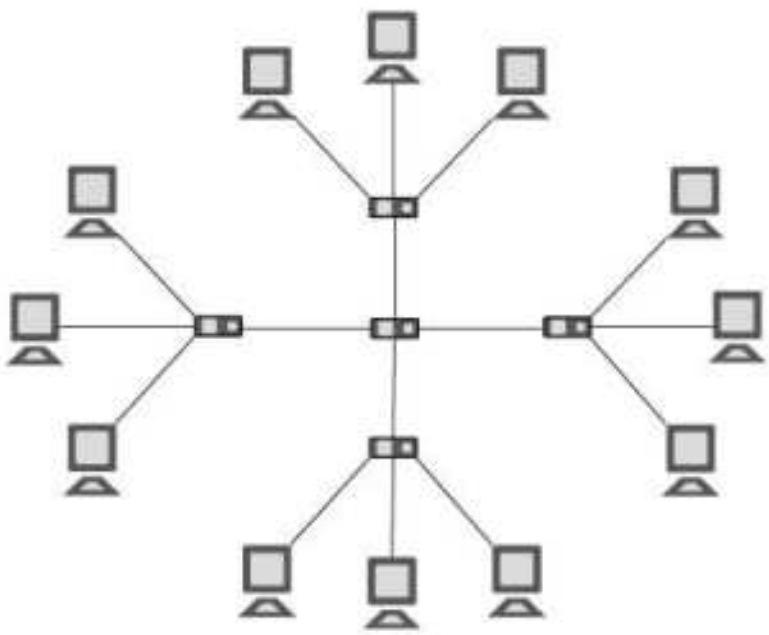
Rys. 1.2. Magistrala

2. W **topologii gwiazdy** (rys. 1.3.) urządzenia końcowe są podłączone do jednego punktu centralnego (koncentratora lub przełącznika). W tym układzie medium transmisyjne stanowi skrętka lub kabel światłowodowy. Każdy z kabli jest wykorzystywany do połączenia z siecią dokładnie tylko jednego urządzenia końcowego.



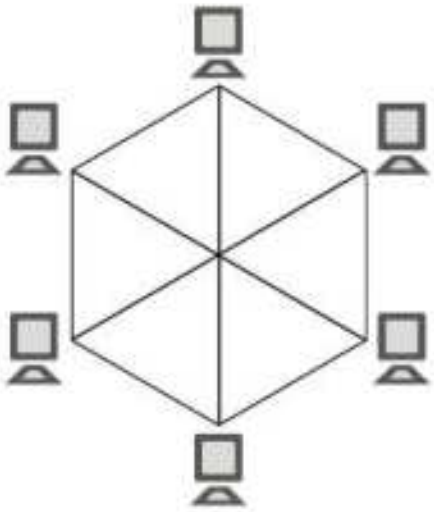
Rys. 1.3. Topologia gwiazdy

3. **Topologia gwiazdy rozszerzonej** (rys. 1.4.) jest swoistą odmianą topologii gwiazdy. Sieć skonstruowana na jej planie ma punkt centralny i punkty boczne. Taka sieć była chętnie wykorzystywana z zastosowaniem Ethernetu.



Rys. 1.4. Topologia gwiazdy rozszerzonej

4. **Topologia siatki** (rys. 1.5.) to układ sieci zbudowany w ten sposób, że elementy łączą się bezpośrednio, dynamicznie z jak największą liczbą urządzeń, które współpracują ze sobą w celu jak najbardziej efektywnego trasowania danych. Zaletą tej topologii jest jej niska zawodność uzyskana poprzez możliwość przesyłania danych różnymi ścieżkami.



Rys. 1.5. Topologia siatki

Topologia logiczna opisuje sposób, w jaki dane przepływają między urządzeniami. Wyróżniamy dwa rodzaje topologii logicznych:

A. **Topologia rozgłoszenia** polega na tym, że host wysyła dane do wszystkich innych hostów podłączonych do medium transmisyjnego. Kolejność korzystania z medium funkcjonuje na zasadzie „kto pierwszy, ten lepszy". Technologia ta jest z powodzeniem wykorzystywana w sieciach Ethernet.

B. **Topologia przekazywania tokena**, zwana też topologią przekazywania żetonu, polega na kontrolowaniu przez urządzenia dostępu do sieci poprzez przekazywanie elektronicznego tokena (żetonu). Host mający w danym momencie żeton może skorzystać z medium transmisyjnego. Host tak długo korzysta z tokena, aż wykona wszystkie swoje zadania, a następnie przekazuje go następnemu hostowi.

ZADANIA DO WYKONANIA

1. Oblicz, ile czasu zajmie przesłanie informacji zapisanych na płycie CD o pojemności 700 MB. Transfer łącza danych wynosi 10 Mb/s.
2. Oblicz, ile danych możemy przesłać w ciągu 30 minut łączem o przepustowości 300 Mb/s.
3. Kolega poprosił cię o pomoc w zaprojektowaniu sieci domowej i chce utworzyć sieć z wykorzystaniem połączenia przewodowego z modemu do dwóch komputerów stacjonarnych. Wymień i omów topologie fizyczne i logiczne, jakich należy użyć do stworzenia najlepszej struktury sieciowej. Uzasadnij swój wybór.

PODSUMOWANIE LEKCJI

- **Sieć komputerowa** jest zbiorem komputerów i urządzeń (telefony, drukarki czy telewizory) połączonych ze sobą w celu wymiany i przesyłania danych. s. 8
- Wymiana danych w sieciach komputerowych odbywa się z określoną prędkością. Podstawową jednostką danych w informatyce jest 1 bit [b]. Do określania prędkości przesyłania danych zastosujemy zatem liczbę bitów przesłanych w ciągu jednej sekundy. Podstawową jednostką jest więc bit na sekundę (b/s bądź bps – *bit per second*). s. 10
- Urządzenia sieciowe są ze sobą połączone za pomocą mediów transmisyjnych przewodowych i bezprzewodowych. s. 11
- **Topologia sieci komputerowej** to relacja pomiędzy urządzeniami w sieci, połączeniami między nimi oraz sposób, w jaki między nimi przepływają dane. s. 12

2. Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej

NA TEJ LEKCJI:

- dowiesz się, jak skonfigurować router w sieci lokalnej opartej na połączeniach kablowych oraz połączeniach bezprzewodowych z użyciem serwera DHCP;
- nauczysz się konfigurować sieć z routerem bez serwera DHCP.

Konfiguracja sieci lokalnej z routerem nie jest skomplikowana. Producent wraz z urządzeniem dostarcza instrukcję, jak krok po kroku uruchomić w domu lokalną sieć. Niestety, nie zawsze udaje nam się przejść przez ten proces bez kłopotów. Pojawiają się one, gdy musimy zrezygnować z serwera DHCP.

2.1. Konfiguracja sieci z routerem

konfiguracja sieci z routerem

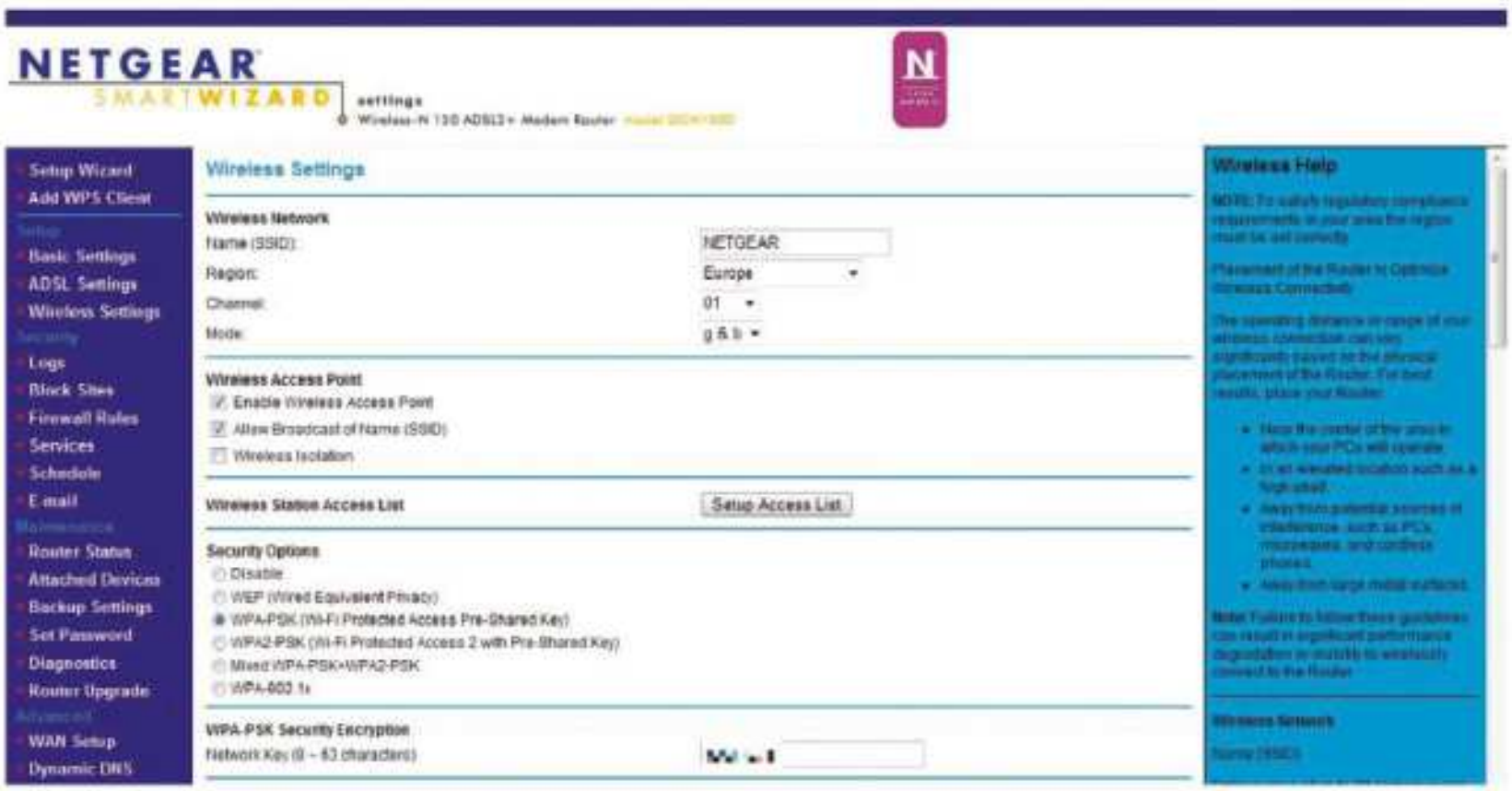
Jeżeli zamierzamy skonfigurować sieć lokalną, powinniśmy ją oprzeć na kilku komputerach i routerze. Tego typu konfiguracja jest najprostszym rozwiązaniem, ponieważ router dba o to, aby wszystkie komputery z nim połączone miały automatycznie przydzielony adres IP i zazwyczaj udostępnia internet. Podczas konfiguracji musimy pamiętać, aby ustawienia, które wprowadzimy, zapewniały jak najwyższy poziom bezpieczeństwa. Dotyczy to przede wszystkim połączeń bezprzewodowych, ale także tych tradycyjnych opartych na kablu.

Na początku odłącz od komputera modem i połącz go kablem sieciowym z routerem. Uruchom przeglądarkę internetową i wpisz adres administracyjny urządzenia. Znajdziesz go w instrukcji instalacji routera. Na stronie, która się otworzy, podaj nazwę konta administratora i hasło. Te informacje również odszukasz w instrukcji lub na obudowie routera. Wszystkie ustawienia konfiguracyjne zawsze zależą od wybranego przez ciebie modelu routera, nazewnictwo jednak jest zbliżone w większości z nich.

Do instalacji sieci wykorzystamy urządzenie, które będzie posiadało wszystkie prezentowane na lekcji funkcje. O zasadach doboru sprzętu uczyłeś się w klasie pierwszej. Wybierając router, musisz przeanalizować swoje potrzeby oraz zorientować się, jakie modele dostępne są na rynku. W ten sposób dobierzesz sprzęt najbardziej odpowiadający twoim wymaganiom.

Jeżeli nie korzystasz z połączeń bezprzewodowych, ważne dla ciebie ustawienia znajdziesz w sekcji **LAN Setup** (czyt. lan set ap) w grupie **Adanced** (czyt. edwensd). Od razu zauważysz, że masz zaznaczone pole **Use Router as DHCP Server** (czyt. juz ruter es diejczcipi serwer) oraz widoczny zakres adresów, które będą używane. W górnym rogu jest wyświetlony adres routera.

Jeżeli chcielibyśmy przeprowadzić konfigurację sieci bezprzewodowej (rys. 2.1.), to powinniśmy przejść do sekcji **Wireless Settings** (czyt. łajerles setins) w grupie **Setup**. Tutaj musimy podać nazwę sieci, ale przede wszystkim włączyć szyfrowanie transmisji danych. Niezależnie od tego, jaki model routera wybierzemy, ta opcja zawsze powinna być włączona. Zazwyczaj zaleca się szyfrowanie WPA2-PSK. Kolejnym krokiem będzie ustawienie odpowiednio silnego hasła – powinno się składać z co najmniej 8 znaków, w tym dużej litery, cyfry i znaku specjalnego. Nie używaj nigdy nazw własnych, a tym bardziej nazw, które można z tobą skojarzyć.



Rys. 2.1. Konfiguracja sieci bezprzewodowej

instrukcja dołączona do routera

Sieć, która bazuje na routerze, jest bardzo łatwa w konfiguracji. Wszyscy producenci routerów, które wykorzystujemy w domu, dołączają instrukcję jego instalacji i konfiguracji w formie elektronicznej lub papierowej.

Jeżeli nasze urządzenie zapewnia połączenie z internetem, powinniśmy przejść do ustawień w sekcji **Basic Setting** (czyt. bejzik setin). To w tym miejscu wpisujemy wszystkie dane, które dostarczył nam dostawca internetu – *internet service provider* (czyt. internet serwis prowajder, ISP).

Jedną z najważniejszych informacji, jaką powinniśmy otrzymać od dostawcy internetu, są reguły wbudowanej zapory. Dotyczą one wszystkich rodzajów połączeń, jakie oferuje nam dostawca – klikamy w tym celu **Firewall rules** (czyt. fajerłol ruls).

Reguły zabezpieczające są zazwyczaj podzielone na dwie grupy: zabezpieczanie połączeń przychodzących oraz wychodzących. W pierwszej grupie blokowana jest cała transmisja, a w drugiej – tylko jej część. Jeżeli chcesz zmienić te reguły i dostosować je do swoich potrzeb, użyj przycisku **Add** (czyt. ad) w odpowiedniej grupie. Pozycję zmieniasz za pomocą przycisku **Move** (czyt. muw). Wszelkie wprowadzone zmiany zatwierdź przyciskiem **Apply** (czyt. aplaj).

Ostatnim etapem będzie wskazanie typu sieci, z której chcemy korzystać – będzie to *Dom* lub *Sieć domowa* (w zależności od wersji systemu Windows).

Jeżeli używasz połączenia bezprzewodowego, kliknij prawym przyciskiem myszy ikonę sieci i wybierz **Połącz z siecią**. Następnie kliknij dwa razy w nazwę sieci – będzie ona taka sama jak w polu **Name (SSID)** routera. W polu **Klucz zabezpieczeń** wpisz hasło WPA2 i kliknij **Połącz**.

2.2. Konfiguracja sieci z routerem bez DHCP

konfiguracja sieci z routerem bez DHCP

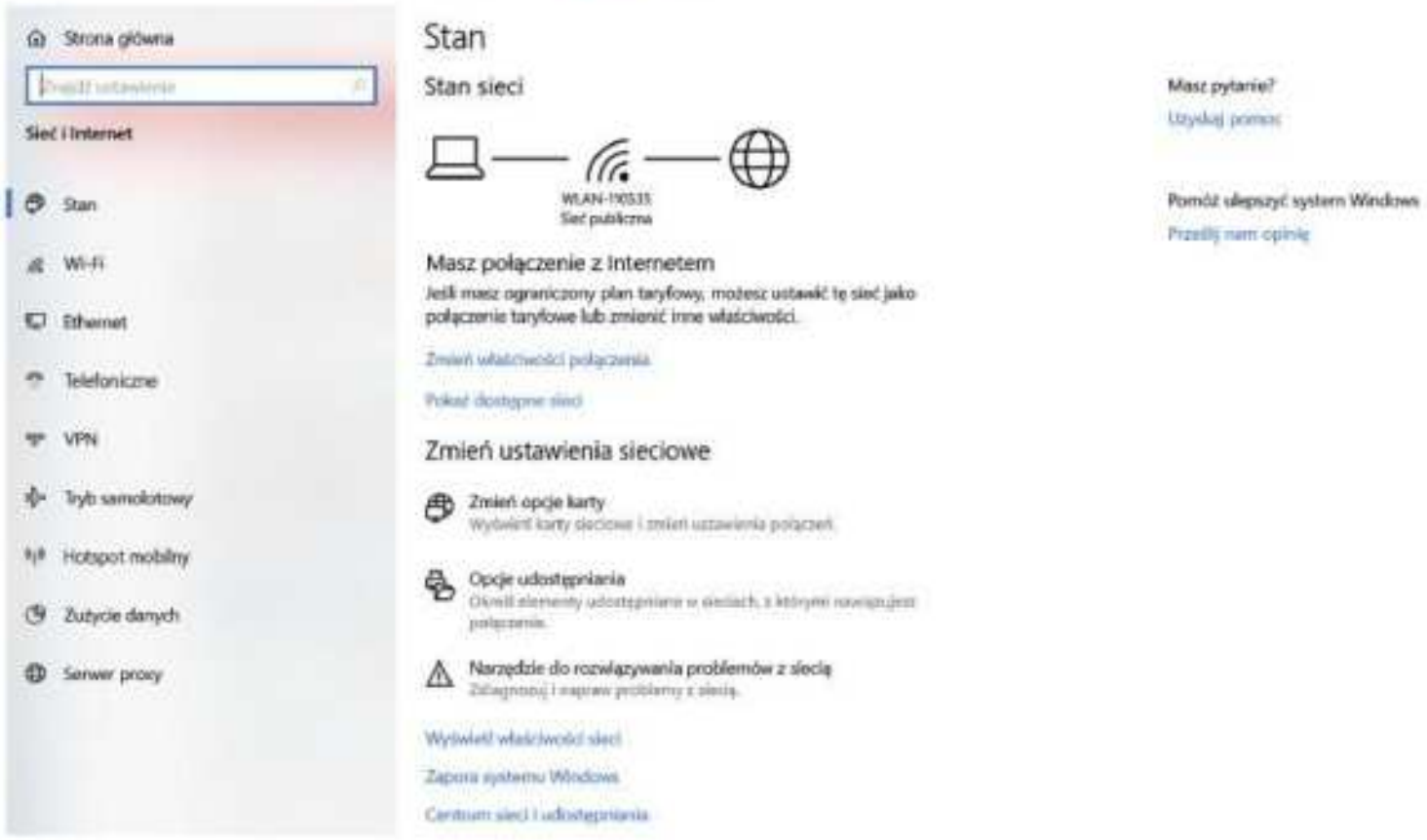
Podczas konfiguracji sieci może się okazać, że nasze łącze internetowe korzysta z modemu USB, który nie jest obsługiwany przez router. Będziemy wówczas zmuszeni wyłączyć serwer DHCP na naszym routerze i ręcznie zdefiniować adresy IP.

Ręczne przypisywanie adresów IP może również poprawić nasze bezpieczeństwo w sieci. Router funkcjonuje wówczas tylko jako przełącznik, który zapewnia transmisję danych między naszymi komputerami.

Aby rozpocząć ręczne definiowanie adresów, musimy zacząć od wyłączenia serwera DHCP na naszym routerze. W tym celu należy otworzyć stronę konfiguracyjną. W sekcji **LAN Setup** w grupie **Adanced** usuwamy zaznaczenie **Use Router as DHCP Server**.

Wracamy do części **LAN TCP/IP Setup** i w polu **IP Address** rezerwujemy nasz adres IP, którym będzie się posługiwał router. Powinniśmy również wpisać maskę. Akceptujemy wprowadzone zmiany.

Teraz klikamy ikonę sieci na naszym komputerze (w prawym dolnym rogu paska stanu) i z menu kontekstowego wybieramy **Otwórz ustawienia sieci i internetu**. Otworzy nam się okno **Stan** (rys. 2.2.).



Rys. 2.2. Okno **Stan**, otwarte po wybraniu **Otwórz** ustawienia sieci i internetu

Teraz z listy, która się znajduje po lewej stronie, wybieramy **Ethernet**. Otworzy nam się okno, w którym z prawej strony wybieramy **Zmień opcje karty** (rys. 2.3.).



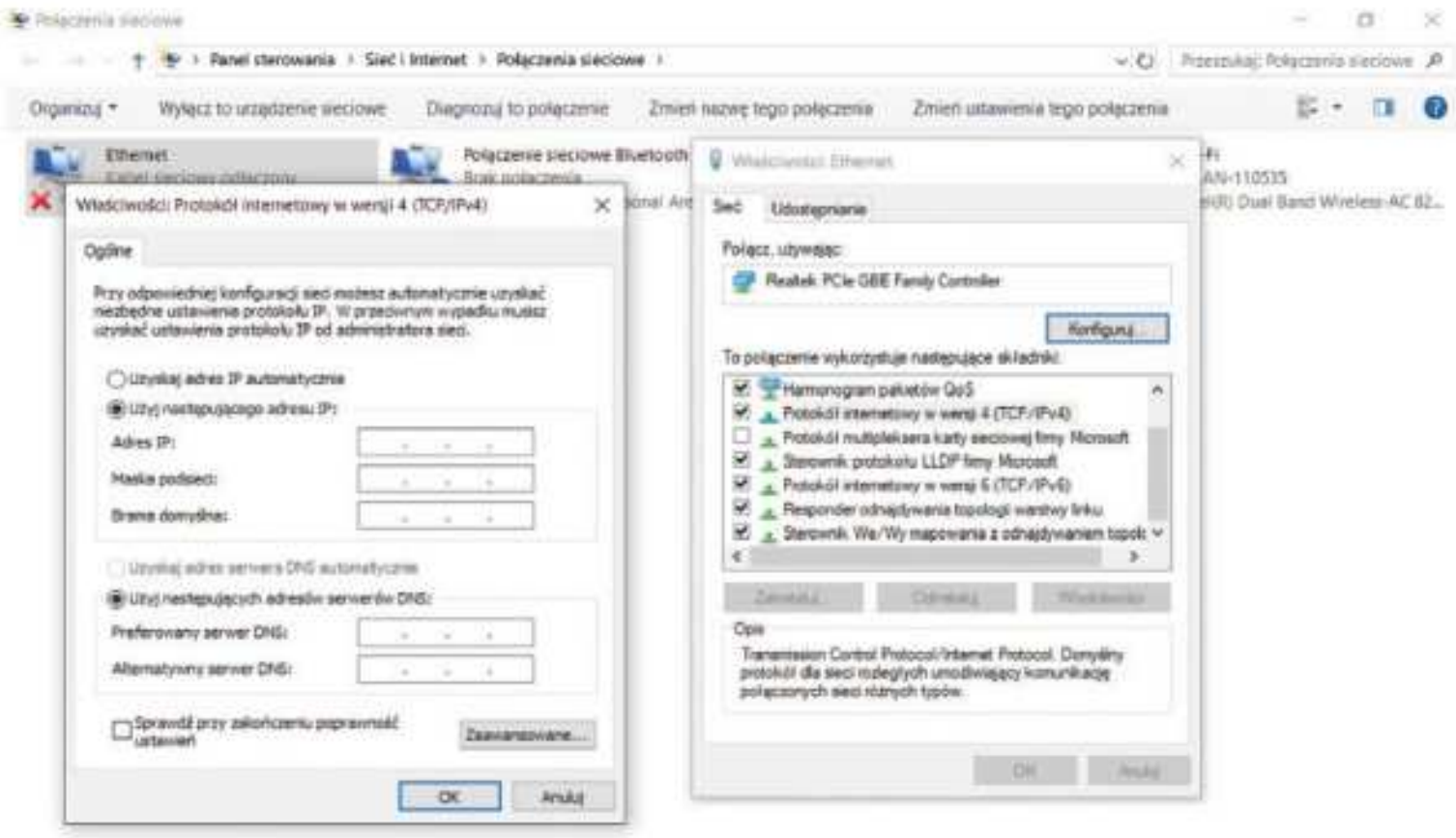
Rys. 2.3. Okno, w którym wybieramy **Zmień opcje karty**

Następnie pojawi się okno, w którym możemy zmienić opcje karty sieciowej (rys. 2.4.).



Rys. 2.4. Okno połączeń sieciowych

Klikamy prawym przyciskiem myszy ikonę karty sieciowej – będzie ona opisana jako **Połączenie lokalne** lub **Połączenie sieci bezprzewodowej**. Zaznacz **Protokół internetowy wersji 4** i użyj przycisku **Właściwości**. Następnie przejdź do zakładki **Ogólne** i przestaw zaznaczenie na opcję **Użyj następującego adresu IP**. W polu **Adres IP** wpisz adres, a poniżej określ maskę sieci. Musisz również zdefiniować adres bramy domyślnej – jest to adres komputera, który zapewnia komunikację między twoją siecią a światem zewnętrznym. Aby komunikacja przebiegała bez zakłóceń, musisz wpisać jeszcze adres DNS. Wszystkich tych informacji powinien udzielić ci twój dostawca internetu. Na koniec zatwierdzasz przyciskiem **OK** i zamykasz resztę okien.



Rys. 2.5. Okno **Właściwości Protokołu internetowego** w wersji 4

ZADANIA DO WYKONANIA

1. Sprawdź w oprogramowaniu twojego routera, jaki ma on adres IP (Router IP Address).
2. Sprawdź w oprogramowaniu twojego routera, jaki jest statyczny adres DNS (Static DNS).
3. Sprawdź w oprogramowaniu twojego routera, jaka jest maska podsieci (Subnet Mask).

PODSUMOWANIE LEKCJI

- Jeżeli chodzi o konfigurację sieci lokalnej, połączenie komputerów z routerem jest najprostszym rozwiązaniem, ponieważ router dba o to, aby wszystkie komputery z nim połączone miały automatycznie przydzielony adres IP i zazwyczaj udostępnia internet. s. 15
- Konfigurację taką – niezależnie od tego, czy sieć jest oparta na łączach przewodowych czy bezprzewodowych – przeprowadzamy zazwyczaj za pomocą instrukcji dołączonej do routera. s. 17
- Jeżeli łącze internetowe korzysta z modemu USB, który nie jest obsługiwany prze router, musimy wyłączyć serwer DHCP na naszym routerze i ręcznie ustawić wszystkie dane – adres IP, maskę podsieci i bramę. s. 17

3. Nie tylko Wi-Fi, czyli bezprzewodowy dostęp do internetu

NA TEJ LEKCJI:

- zapoznasz się z bezprzewodowymi mediami transmisyjnymi;
- poznasz, czym jest WLAN;
- dowiesz się, co to jest Wi-Fi;
- uzyskasz informacje o standardach IrDa, Bluetooth i WiMAX.

Gdy słyszymy „sieć bezprzewodowa", natychmiast na myśl przychodzi nam Wi-Fi. Problem zaczyna się wtedy, gdy próbujemy zgłębić zasady jej funkcjonowania. Wówczas się okazuje, że sieć bezprzewodowa to nie tylko Wi-Fi. Istotą jej jest nawiązanie połączenia między urządzeniami bez użycia przewodów. Sieci bezprzewodowe najczęściej wykorzystują mikrofale (te same, które emituje kuchenka mikrofalowa) lub podczerwień.

3.1. Bezprzewodowe media transmisyjne

media bezprzewodowe

Media bezprzewodowe to druga grupa mediów transmisyjnych. Stosuje się w nich przede wszystkim fale radiowe wykorzystywane przez Wi-Fi, o którym dowiesz się więcej w następnym rozdziale.

fale radiowe

Fale radiowe to promieniowanie elektromagnetyczne o częstotliwościach od częstotliwości słyszalnych (15 kHz) do częstotliwości fal widzialnych ($3x\ 10^{14}$ Hz). Obecnie najpopularniejszymi częstotliwościami używanymi do transmisji bezprzewodowej są 2,4 GHz i wyższe. Najnowszym standardem w sieciach bezprzewodowych jest standard „Gigabit Wi-Fi". Jest to już piąta już generacja sieci bezprzewodowych, której głównym celem jest zapewnienie prędkości połączenia przekraczającej 1 Gb/s.

podczerwień IR

Fale elektromagnetyczne z zakresu podczerwieni **IR** (InfraRed) są stosowane na otwartym terenie bądź w pomieszczeniach. Jako źródło promieniowania fal elektromagnetycznych wykorzystuje się diody LED (*light emitting diode*, czyt. lajt emitin dajod) lub diody laserowe. Największym ograniczeniem podczerwieni jest niewielki zasięg – wynoszący do kilku metrów.

Główną zaletą mediów bezprzewodowych jest to, że mogą przenieść duże ilości danych przy odpowiednio wysokich częstotliwościach. Natomiast największe wady to tłumienie i dyfrakcja (nakładanie się) sygnału powodowane przez różnego rodzaju przeszkody znajdujące się na drodze fali, która niesie sygnał.

3.2. WLAN

WLAN

WLAN (patrz s. 9) to lokalna sieć komputerowa, w której połączenia między urządzeniami są bezprzewodowe. Urządzenia tak połączone pozostają w niewielkiej

odległości od siebie. Tego typu sieci stosuje się zazwyczaj w domach, szkołach czy instytucjach.

Urządzenia, które tworzą sieć WLAN, wykorzystują do połączenia mikrofale w charakterze medium przenoszącego sygnał. Oprócz fal radiowych WLAN wykorzystuje również fale podczerwieni. Sieci WLAN są projektowane w zgodności ze standardami IEEE 802.11. Jest to grupa standardów IEEE 802, która opisuje warstwę fizyczną i podwarstwę MAC w sieciach bezprzewodowych. Standard ten był też pierwszym standardem określającym dwie prędkości transmisji – 1 oraz 2 Mb/s. Podczerwień w tym standardzie się nie przyjęła, natomiast dalsze prace nad transmisją bezprzewodową szybko doprowadziły do rozbudowy standardu IEEE 802.11.

Na infrastrukturę sieci WLAN składają się standardowe urządzenia sieciowe, takie jak: karty sieciowe, punkty dostępu, rozdzielacze, przejściówki oraz anteny.

Do komunikacji za pomocą mikrofal wykorzystuje się zazwyczaj pasmo 2,4 GHz lub 5 GHz. W praktyce na tych częstotliwościach mogą pracować jedynie trzy sieci, aby wzajemnie się nie zakłócały. Pasmo każdej z nich wynosi około 20 MHz.

Szyfrowanie danych w sieciach bezprzewodowych to szyfrowanie WPA lub WPA2. Użytkownik chcący korzystać z sieci bezprzewodowej musi znać hasło, na podstawie którego jest generowany klucz dostępu.

3.3. Wi-Fi

Wi-Fi

Wi-Fi to znak towarowy (rys. 3.1.). Powszechnie używa się go do sygnowania urządzeń opartych na standardach IEEE 802.11, które są wykorzystywane przez sieci bezprzewodowe. Jest on własnością firmy Wi-Fi Alliance. Na każdym urządzeniu wykorzystującym standard IEEE 802.11 znajdziemy czarno-białe logo. Obecnie znakomita większość domowych urządzeń sieci bezprzewodowej działa na takich samych zasadach, więc potocznie przyzwyczailiśmy się je określać mianem Wi-Fi.



Rys. 3.1. Logo Wi-Fi

Obecnie korzystamy z wielu różnych standardów IEEE 802.11 sieci bezprzewodowych:

- 802.11a – oferuje połączenia o prędkości do 54 Mb/s i częstotliwości 5 GHz,
- 802.11b – oferuje połączenie do 11 Mb/s i częstotliwość 2,4 GHz, swoim zasięgiem obejmuje około 30 m we wnętrzu i do 120 m w przestrzeni otwartej,
- 802.11g – prędkością transferu dorównuje standardowi 802.11a, a częstotliwością 802.11b, w praktyce osiąga jednak słabsze wyniki, ponieważ jest bardziej podatny na zakłócenia, zwłaszcza gdy korzystamy ze sprzętu starszej generacji,

- 802.11n – najczęściej spotykany standard, który oferuje transfer do 300 Mb/s przy częstotliwości 5 GHz – wymaga bardzo mocnego i stabilnego sygnału,
- 802.11ac – specyfikacja standardu zakłada prace wyłącznie w paśmie 5 GHz, a więc kompatybilność wsteczna będzie zapewniona tylko dla urządzeń działających w tym paśmie (802.11a/n); głównym celem tego standardu jest zapewnienie prędkości połączenia przekraczającej 1 Gb/s,
- 802.11ax – to standard Wi-Fi, znany także pod nazwą Wi-Fi 6, który został zaprojektowany do działania we wszystkich pasmach ISM od 1 do 6 GHz, oprócz już przydzielonych pasm 2,4 i 5 GHz; urządzenia umożliwiają łączność z prędkością 11 Gb/s; w typowych warunkach bez większych problemów nowy standard Wi-Fi powinien zaoferować prędkości na poziomie 1–1,5 Gb/s,
- 802.11ay – standard ten będzie używał częstotliwości 60 GHz, a jego przewidywaną prędkość szacuje się na zakres między 20 a 40 Gb/s z zasięgiem nawet 300 metrów.

Technologią stworzoną na bazie Wi-Fi jest Wi-Fi Direct (czyt. łajfaj dajrekt), podobna do Bluetooth (czyt. blutuf). Wykorzystuje się ją w telekomunikacji. Jej główną zaletą jest znacznie większy zasięg. Od Wi-Fi różni się przede wszystkim tym, że komunikacja między urządzeniami zachodzi bezpośrednio, bez pośrednictwa punku dostępowego czy routera. Dzięki takiemu rozwiązaniu mamy możliwość szybszego wydrukowania, synchronizacji, udostępniania zawartości itp. Technologia ta nie ogranicza się tylko do telefonów. Umożliwia także przesyłanie obrazów, dźwięku czy plików wideo między telewizorem a tabletem czy laptopem. Technologia Wi-Fi Direct jest również wykorzystywana w grach.

3.4. IrDa, Bluetooth i WiMAX

IrDa

Wi-Fi to nie jedyna technologia komunikacji bezprzewodowej. W 1993 roku powstała **IrDa** (*Infrared Data Association*, czyt. infrared dejta asosijejszyn) – grupa skupiająca kilkudziesięciu producentów sprzętu komputerowego. Zamierzała stworzyć i kontrolować standardy transmisji danych z wykorzystaniem fal podczerwieni. IrDa opracowała swój system bezprzewodowej transmisji danych – nazywany potocznie IrDą. Głównym założeniem tego sytemu było tworzenie sieci tymczasowych, w których znajdowały się urządzenia przenośne – laptopy, plamtopy itp. IrDa oferowała możliwość przesyłania danych między urządzeniami, drukowanie i sterowanie urządzeniami telekomunikacyjnymi. Swego czasu cieszyła się również popularnością w telefonii komórkowej. Charakteryzowała się małym poborem mocy oraz tym, że przesył danych odbywał się punkt w punkt. Niestety, transfer był bardzo wolny. Obecnie system ten praktycznie nie jest już spotykany – został wyparty przez Bluetooth i Wi-Fi.

Bluetooth

Bluetooth (rys. 3.2.) jest standardem bezprzewodowej komunikacji krótkiego zasięgu, opartym na otwartym standardzie IEEE 802.15.1 – jego specyfikacja obejmuje trzy klasy mocy nadawczej: 100, 10 i 1 m otwartej przestrzeni. Bluetooth wykorzystuje fale radiowe w paśmie częstotliwości 2,4 GHz. Obecnie mamy wersję Bluetooth 5.0 – urządzenia te oferują nam transfer o prędkości 2 Mb/s dla urządzeń typu wearables (czyt. łerejbls; ubrania oraz akcesoria

przeznaczone do noszenia, zawierające w sobie komputer lub zaawansowane technologie elektroniczne, np. smartwatch, czyt. smartłocz) i 50 Mb/s dla pozostałych – realny zasięg działania wynosi około 140 m.



Rys. 3.2. Logo bluetooth

WiMAX (*Worldwide interoperability for microwave access*, czyt. łorldłajd interoperabiliti for majkrołejw akses) jest techniką bezprzewodowej transmisji danych powstałej na bazie standardów IEEE 802.16, Europejskiego Instytutu Norm Telekomunikacyjnych (ETSI) i HiperLAN (*high performance radio local area network*, czyt. haj performens redijoł lokal erja netłork) – europejskiego standardu transmisji radiowej w sieciach lokalnych). Standardy te zostały stworzone dla szerokopasmowego, radiowego dostępu na dużych obszarach. W 2009 roku największe światowe sieci komórkowe po kilkuletnich testach zaczęły odchodzić od standardów WiMAX na rzecz LTE. Kilka lat później cena urządzeń telekomunikacyjnych uległa obniżeniu na tyle, że standard WiMAX zaczął ponownie cieszyć się powodzeniem wśród operatorów sieci komórkowych. Przepustowość tego standardu zbliża się do 175 Mb/s. Prędkość ta jest dostępna do 10 km od nadajnika. Obecnie w Polsce przepustowość w tym standardzie to tylko 4 Mb/s, ale za to może ona działać w odległości 50 km od nadajnika.

WiMAX

ZADANIA DO WYKONANIA

1. Poszukaj w internecie informacji na temat standardów IEEE 802.11c i IEEE 802.11ax.
2. Wyszukaj w internecie informacje na temat standardu bezprzewodowego przesyłu danych LTE.

PODSUMOWANIE LEKCJI

- Media bezprzewodowe to grupa mediów transmisyjnych, która przesył informacji opiera na falach radiowych i podczerwieni. s. 20
- WLAN to lokalna sieć komputerowa, która do łączności między urządzeniami wykorzystuje bezprzewodowe media transmisyjne. Urządzenia połączone w sieci WLAN pozostają w niewielkiej odległości od siebie. s. 20
- **Wi-Fi** to znak towarowy stworzony i wypromowany przez firmę Wi-Fi Alliance na urządzeniach wykorzystujących standardy IEEE 802.11 do przesyłu danych. s. 21
- **IrDa** to grupa skupiająca producentów urządzeń komputerowych, która stworzyła standard przesyłu informacji za pomocą fal podczerwieni. s. 22
- **Bluetooth** jest standardem komunikacji bezprzewodowej krótkiego zasięgu. s. 22
- Połączenie **WiMAX** to szerokopasmowy standard komunikacji bezprzewodowej. s. 23

PODSUMOWANIE PRZED SPRAWDZIANEM

1. Host i serwer, czyli o działaniu sieci

s. 8
- **Sieć komputerowa** jest zbiorem komputerów i urządzeń (telefony, drukarki czy telewizory) połączonych ze sobą w celu wymiany i przesyłania danych.

s. 10
- Wymiana danych w sieciach komputerowych odbywa się z określoną prędkością. Podstawową jednostką danych w informatyce jest 1 bit [b]. Do określania prędkości przesyłania danych zastosujemy zatem liczbę bitów przesłanych w ciągu jednej sekundy. Podstawową jednostką jest więc bit na sekundę (b/s bądź bps – *bit per second*).

s. 11
- Urządzenia sieciowe są ze sobą połączone za pomocą mediów transmisyjnych przewodowych i bezprzewodowych.

s. 12
- **Topologia sieci komputerowej** to relacja pomiędzy urządzeniami w sieci, połączeniami między nimi oraz sposób, w jaki między nimi przepływają dane.

2. Włączamy do ruchu, czyli konfiguracja lokalnej sieci komputerowej

s. 15
- Jeżeli chodzi o konfigurację sieci lokalnej, połączenie komputerów z routerem jest najprostszym rozwiązaniem, ponieważ router dba o to, aby wszystkie komputery z nim połączone miały automatycznie przydzielony adres IP, i zazwyczaj udostępnia internet.

s. 17
- Konfigurację taką – niezależnie od tego, czy sieć jest oparta na łączach przewodowych, czy bezprzewodowych – przeprowadzamy zazwyczaj za pomocą instrukcji dołączonej do routera.

s. 17
- Jeżeli łącze internetowe korzysta z modemu USB, który nie jest obsługiwany prze router, musimy wyłączyć serwer DHCP na naszym routerze i ręcznie ustawić wszystkie dane – adres IP, maskę podsieci i bramę.

3. Nie tylko Wi-Fi, czyli bezprzewodowy dostęp do internetu

s. 20
- Media bezprzewodowe to grupa mediów transmisyjnych, która przesył informacji opiera na falach radiowych i podczerwieni.

s. 20
- WLAN to lokalna sieć komputerowa, która do łączności między urządzeniami wykorzystuje bezprzewodowe media transmisyjne. Urządzenia połączone w sieci WLAN pozostają w niewielkiej odległości od siebie.

s. 21
- **Wi-Fi** to znak towarowy stworzony i wypromowany przez firmę Wi-Fi Alliance na urządzeniach wykorzystujących standardy IEEE 802.11 do przesyłu danych.

s. 22
- **IrDa** to grupa skupiająca producentów urządzeń komputerowych, która stworzyła standard przesyłu informacji za pomocą fal podczerwieni.

s. 22
- **Bluetooth** jest standardem komunikacji bezprzewodowej krótkiego zasięgu.

s. 23
- Połączenie **WiMAX** to szerokopasmowy standard komunikacji bezprzewodowej.

II. Opracowania rozwiązań problemów za pomocą wybranych aplikacji

- Zespół, czyli realizujemy projekty w chmurze
- Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi
- Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu
- Broszura, czyli jak projektować duże dokumenty
- Recenzja, czyli proponujemy poprawki w tekście
- OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu

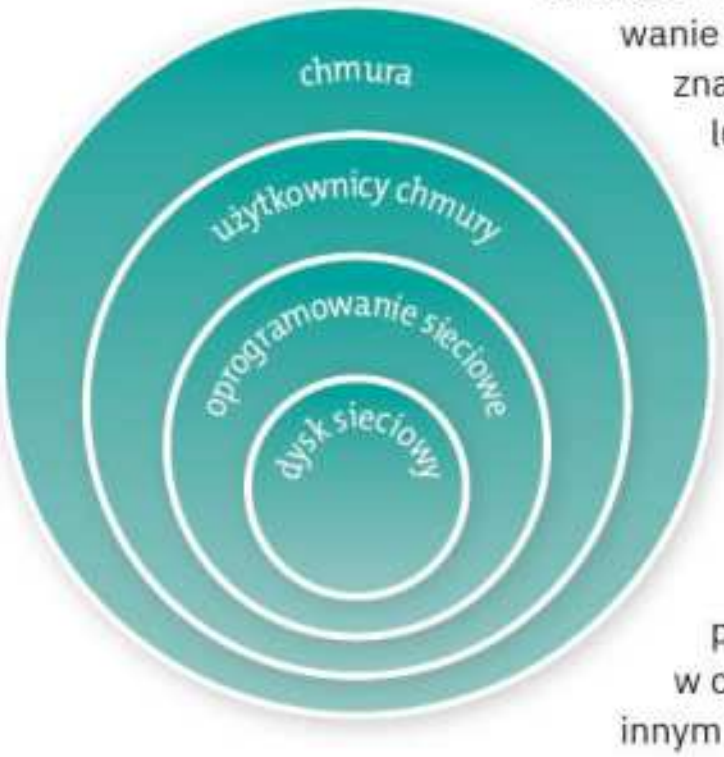
4. Zespół, czyli realizujemy projekty w chmurze

NA TEJ LEKCJI:

- dowiesz się, jak wykorzystać właściwości i usługi chmury do zarządzania zespołem pracującym nad wspólnym projektem;
- poznasz nowe właściwości i aplikacje chmury.

Znasz już niektóre właściwości i aplikacje chmury. Umiesz posługiwać się jej dyskiem oraz udostępniać i edytować dokumenty. Czas na praktyczne wykorzystanie tych umiejętności do planowania lub udoskonalenia pracy zespołu.

4.1. Zespół dostaje narzędzia, czyli dobieramy chmurę do wymagań projektu



Chmura informatyczna udostępnia oprogramowanie uruchamiane w serwerze, dlatego bez znaczenia jest moc twojego komputera lub smartfona. Wystarczy, że sprawnie pracuje przeglądarka internetowa. Podstawowym elementem chmury jest dysk sieciowy. Każdy dokument możesz udostępniać innym użytkownikom lub współdzielić z nimi. W chmurze funkcjonuje wiele usług i programów, między innymi komunikator, dzięki któremu można rozmawiać podczas wspólnej pracy nad projektem. Udostępniając dokument w chmurze, możesz nadawać uprawnienia innym użytkownikom.

Rys. 4.1. Budowa przykładowej chmury

przygotowania do konfiguracji chmury

Załóżmy, że wraz z koleżankami i kolegami chcesz wziąć udział w konkursie na temat historii twojego miasta. Na spotkaniu ustaliliście podział pracy zgodny ze zdolnościami i umiejętnościami poszczególnych osób. Tobie przypadła rola administrowania zasobami informatycznymi i koordynacja prac. To ważne zadanie, w którym musisz odpowiednio **skonfigurować chmurę** – założyć konta użytkowników, stworzyć strukturę folderów i zdecydować o wyborze aplikacji (rys. 4.2.). Wasz projekt wymaga utworzenia wielu dokumentów tekstowych, prezentacji,

grafik, zdjęć i filmów. Zaplanuj zadania, które powinny być wykonane w chmurze (tab. 4.1.). Załóż, że korzystasz z chmury Google. Na jej przykładzie sprawdzisz rozmaite mechanizmy, które później z łatwością wykorzystasz w chmurach komercyjnych lub firmowych.



Rys. 4.2. Przykładowy schemat struktury chmury dla zespołu pracującego nad projektem

Lp.	Cecha	Znaczenie
1.	Przechowywanie plików tekstowych, graficznych, dźwiękowych itp.	Dyski użytkowników. Udostępnianie plików pozostałym członkom zespołu.
2.	Edycja tekstu	Edytor tekstu z możliwością zapisywania dokumentów w różnych formatach, w tym PDF, lub eksportu do postaci HTML (strony internetowej).
3.	Edycja grafiki i fotografii	Edytor grafiki pozwalający na drobne korekty w zdjęciach i tworzenie nieskomplikowanej grafiki, np. strzałek i innych elementów strony internetowej, prezentacji itp.
4.	Edycja prezentacji	Edytor zgodny z najpopularniejszymi formatami prezentacji.
5.	Komunikacja	Chmura powinna udostępniać przynajmniej jedną formę komunikacji bezpośredniej, np. czat, wideoczat itp.
6.	Wspólna edycja dokumentów	Wszystkie zastosowane programy powinny mieć możliwość udostępniania i jednoczesnej edycji przez wielu użytkowników.
7.	System powiadamiania o terminach i kalendarz	Aplikacja kalendarza z możliwością wpisywania wydarzeń i powiadomień.
8.	Baza kontaktów	Aplikacja do zebrania danych uczestników projektu: adresów e-mail, numerów telefonów itp.

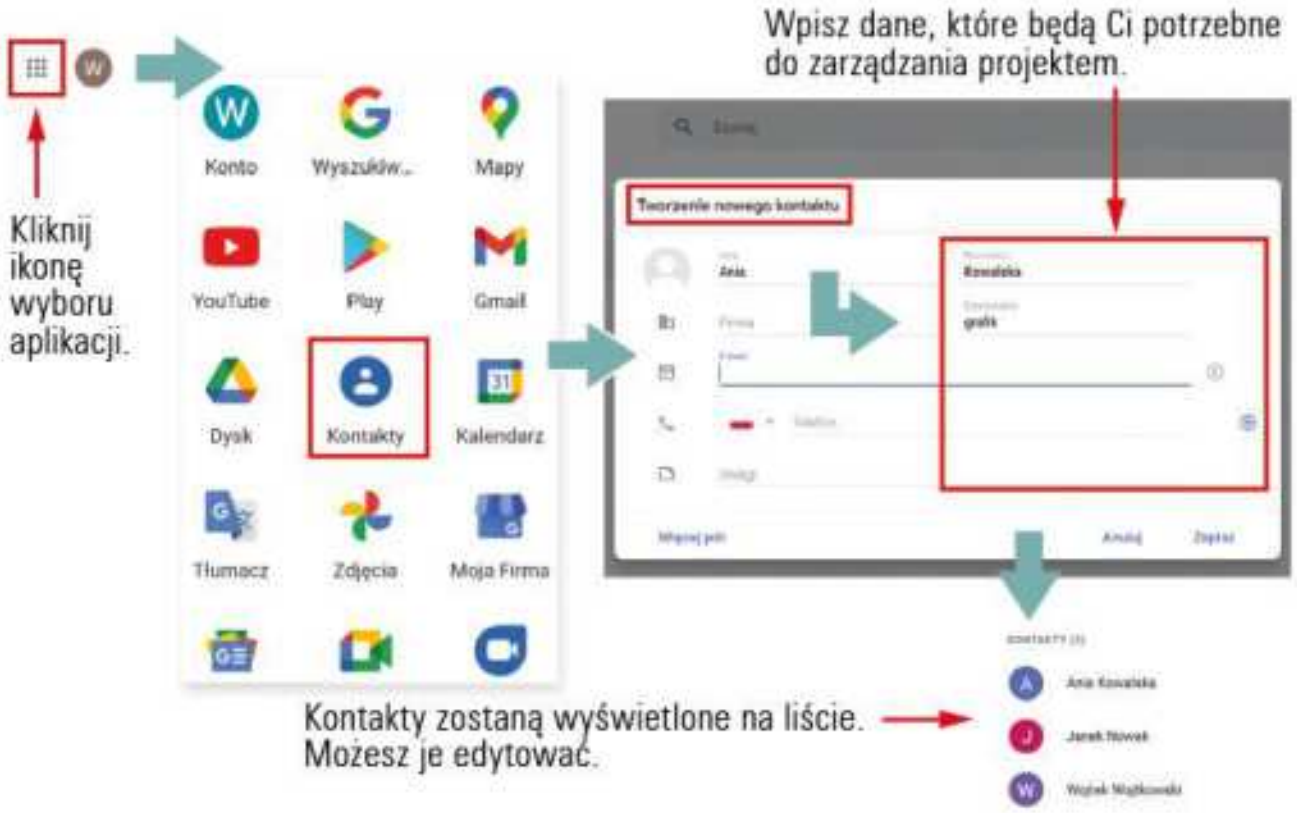
Tab. 4.1. Przykładowe zadania chmury w projekcie zespołowym

Po przejrzeniu możliwości chmury Google można przyjąć, że będzie ona spełniała wymagania pracy zespołu projetowego wynikające z zadań wskazanych w tabeli 4.1.

4.2. W kontakcie, czyli narzędzia do komunikacji w chmurze

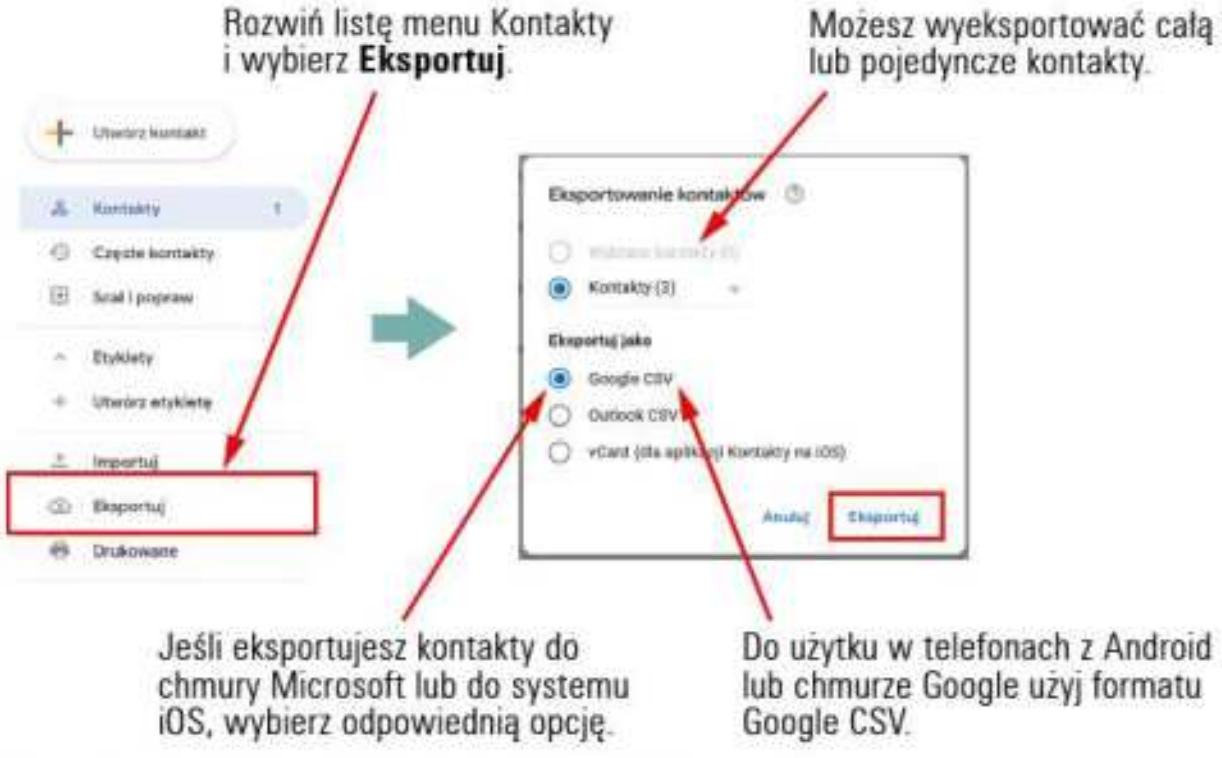
tworzenie kontaktów dla chmury

Przechowywanie, tworzenie i udostępnianie dokumentów nie stanowi już dla ciebie problemu. Możemy więc zająć się innymi zadaniami chmury. Rozpocznij od utworzenia **bazy kontaktów**. Będzie to niezbędne nie tylko podczas pracy zespołu, lecz także w trakcie korzystania z aplikacji komunikacyjnych (rys. 4.3.).



Rys. 4.3. Proces tworzenia nowego kontaktu w chmurze Google

Dzięki wpisanym do chmury kontaktom możesz bez problemu udostępniać dokumenty członkom zespołu. Kiedy wykonasz te czynności, twoje kontakty będą widoczne wyłącznie na twoim koncie, a przecież każdy z członków zespołu powinien mieć na liście wszystkie tworzące go osoby. Nie trzeba jednak niczego przepisywać. Wystarczy wyeksportować dane (rys. 4.4.).



Rys. 4.4. Eksport kontaktów w chmurze Google

Jeśli zdecydujecie się korzystać także z telefonów komórkowych, możesz zsynchronizować kontakty ze smartfonem. Wtedy listy z telefonu i z chmury będą identyczne i nie trzeba będzie przepisywać kontaktów. Taka umiejętność przyda się także podczas wymiany karty SIM lub telefonu. Wykorzystując sieć i konto Google, możesz się uchronić przed utratą listy kontaktów.

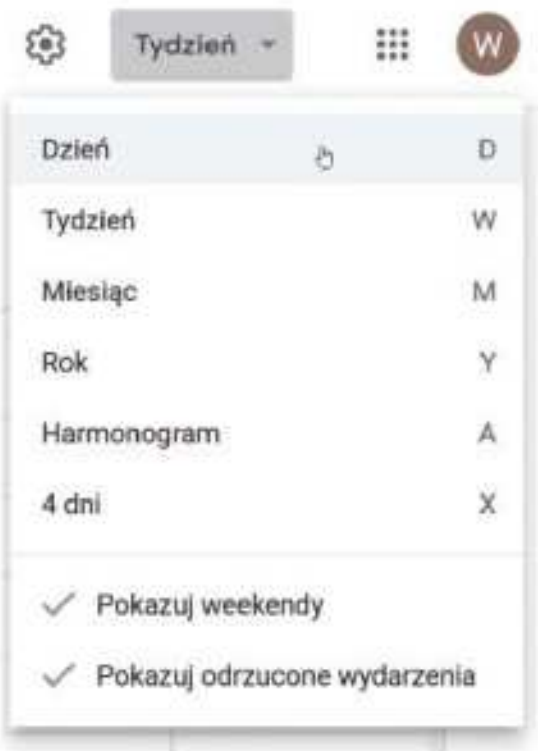
Jeśli twoja lista została już powielona na kontach członków zespołu, możecie wypróbować różne sposoby komunikacji. Najbardziej efektywne jest skorzystanie z aplikacji **Hangouts** (czyt. hengałts; rys. 4.5.). Udostępnia ona różne formy kontaktu, w tym także wideokonferencje kilkorga uczestników.



Rys. 4.5. Uruchomienie aplikacji komunikacyjnej Hangouts

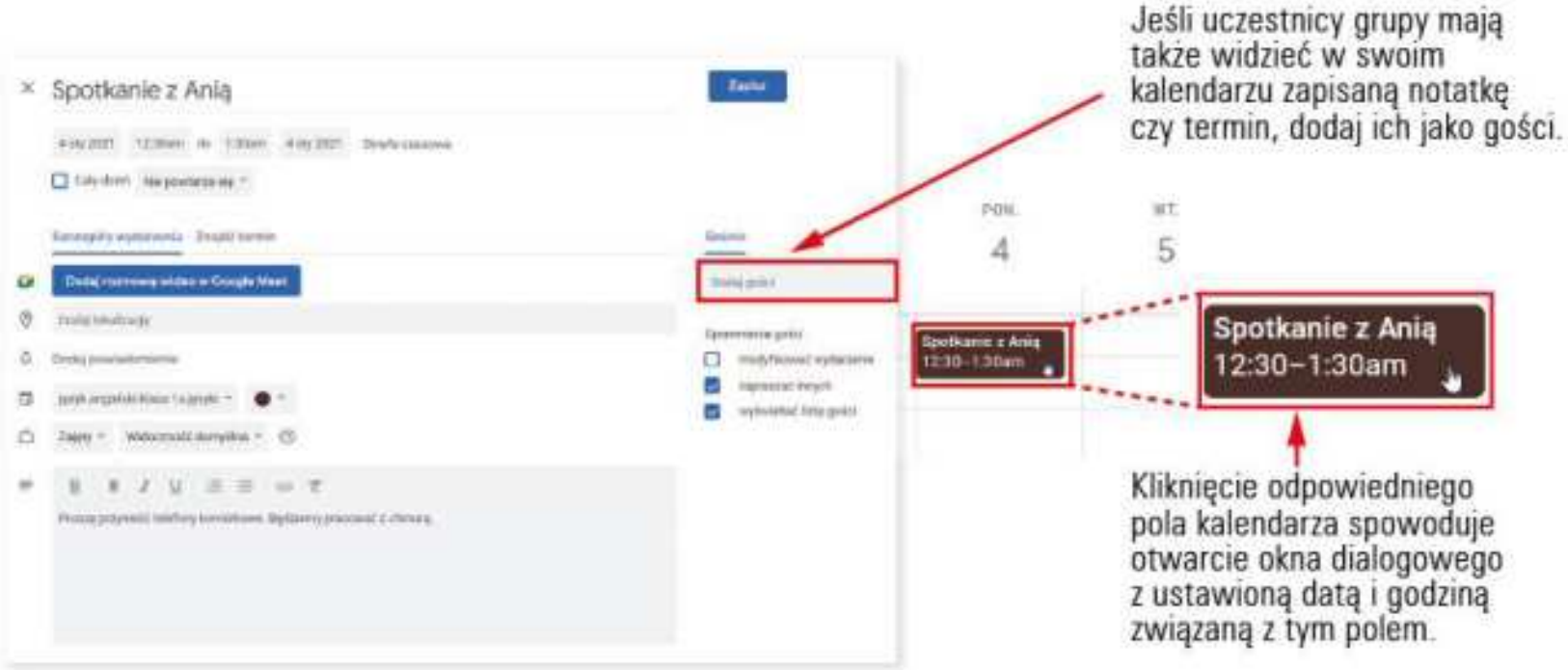
4.3. Nie zapomnieć, czyli chmura pilnuje terminów

W natłoku wielu obowiązków łatwo zapomnieć o terminie spotkania, wykonaniu jakiejś pracy lub wielu innych sprawach. Każdy, kto ma telefon komórkowy, może skorzystać z rozbudowanych funkcji aplikacji Kalendarz. Chmura Google, a także inne chmury również są wyposażone w taką aplikację. **Kalendarz** można wyświetlać na różne sposoby – od widoku całego roku aż do rozkładu jednego dnia (rys. 4.6.).



Rys. 4.6. Ustawienia widoku aplikacji **Kalendarz** w chmurze Google

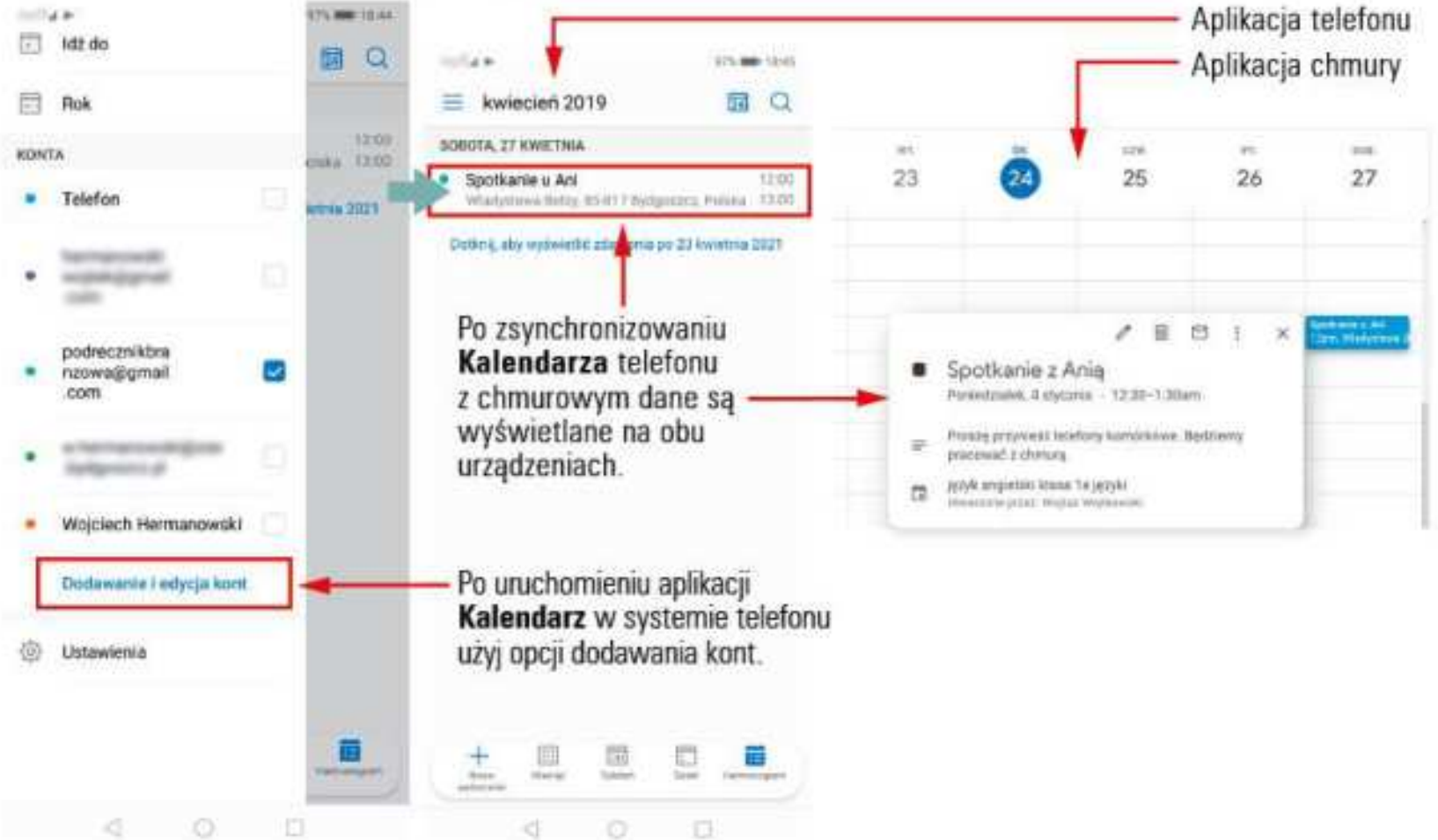
Wpisywanie notatek i terminów do aplikacji jest bardzo łatwe. Wystarczy kliknąć dane pole tabelki kalendarza i rubryki w oknie dialogowym (rys. 4.7.).



Rys. 4.7. Przykład wypełniania **Kalendarza**

synchronizacja z kalendarzem

W Google mamy możliwość **zsynchronizowania kalendarza chmury** z kalendarzem zainstalowanym w telefonie (rys. 4.8.). To ważna cecha, ponieważ telefon zawsze nosimy przy sobie.



Rys. 4.8. Przykładowa aplikacja **Kalendarz** w systemie Android i sposób jej synchronizacji z chmurą

Aplikacje takie jak Kalendarz są zazwyczaj łatwe w obsłudze. Wystarczy kilka ćwiczeń, by korzystać z ich wszystkich możliwości. Poświęć czas na poznanie najważniejszych opcji. W razie potrzeby skorzystaj z tutoriali dostępnych w sieci.

4.4. Wszystko gotowe, czyli zespół pracuje w chmurze

Jeśli chmura po konfiguracji, założeniu odpowiedniej liczby kont i synchronizacji pomiędzy nimi spełnia wymagania z tabeli 4.1., możesz uznać, że jesteś gotowy/gotowa do zarządzania techniczną stroną projektu. Pracę powinieneś/powinnaś rozpocząć od krótkiego szkolenia swoich koleżanek i kolegów, by sprawnie posługiwali się aplikacjami i dyskiem chmury. Przygotuj dla nich proste ćwiczenia, podczas których zalogują się do chmury, wspólnie będą edytować dokumenty, wysyłać, pobierać i udostępniać pliki, a także nauczą się komunikować i korzystać z kalendarza. Od tej chwili już tylko od inwencji twórczej i umiejętności pracy w zespole zależy sukces projektu. Jeśli mieszkacie daleko od siebie, ta odległość nie będzie miała już znaczenia. Każdy może po lekcjach pracować w domu, a jednocześnie wspólnie z koleżankami i kolegami tworzyć dokumenty i gromadzić pliki. Projekt edukacyjny jest ciekawym przedsięwzięciem. Ma on jednak jeszcze inne znaczenie. W trakcie jego realizacji uczysz się organizowania zespołu, współpracy, a także posługiwania się dyskami i aplikacjami chmury. Takie doświadczenie zaowocuje w przyszłości, gdy staniesz przed podobnym wyzwaniem. Bez względu na to, jaki zawód chcesz wykonywać, zdobyte umiejętności są bardzo ważne.

ZADANIA DO WYKONANIA



1. Korzystając z edytora chmury, sporządź dokument, w którym omówisz działanie i przeznaczenie aplikacji Jamboard (czytaj: dżembord). Dokument ilustruj zrzutami ekranowymi. Zastanów się i odpowiedz, czy jest to narzędzie przydatne w każdej sali lekcyjnej.
2. Utwórz kilka kontaktów w chmurze Google, a następnie wykonaj ich kopie (eksportuj na dysk lokalny w formacie Google), pomijając te wcześniej istniejące. Usuń te konta z kontaktów. Upewnij się, że przestały istnieć, a następnie importuj je z pliku. Sprawdź, czy ponownie znalazły się na liście.
3. Na dysku chmury umieść plik ze zdjęciem. Udostępnij go właścicielowi innego konta w chmurze Google poprzez przesłanie mu bezpośredniego odnośnika. Sprawdź, czy go otrzymał i bez problemów pobrał plik. Wypróbuj także inne sposoby udostępniania plików w chmurze Google.
4. Korzystając z odpowiedniego widoku kalendarza chmury Google, zaplanuj swój dzień. Uwzględnij pobyt w szkole, poszczególne lekcje, czas powrotu do domu itp. Wydrukuj tę stronę kalendarza z użyciem odpowiedniej opcji aplikacji.

5*. Zsynchronizuj swój telefon z kontem Google i połącz ze sobą aplikacje Kalendarz. Sprawdź, czy w obu aplikacjach znajdą się te same terminy, np. z zadania 4.

PODSUMOWANIE LEKCJI

- Do każdego zadania chmurę należy odpowiednio **skonfigurować** – założyć konta i sprawdzić, czy jej właściwości i zasoby są wystarczające do realizacji projektu. s. 26
- Bardzo ważnym elementem dla **zespołu projektowego** jest utworzenie wspólnej **bazy kontaktów**. s. 28
- Do **synchronizacji** terminów i przypominania o pracach do wykonania można wykorzystać kalendarz chmury. Można go także synchronizować z kalendarzem z telefonu komórkowego, co znacznie ułatwia powiadomienia. s. 30

5. Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi

NA TEJ LEKCJI:

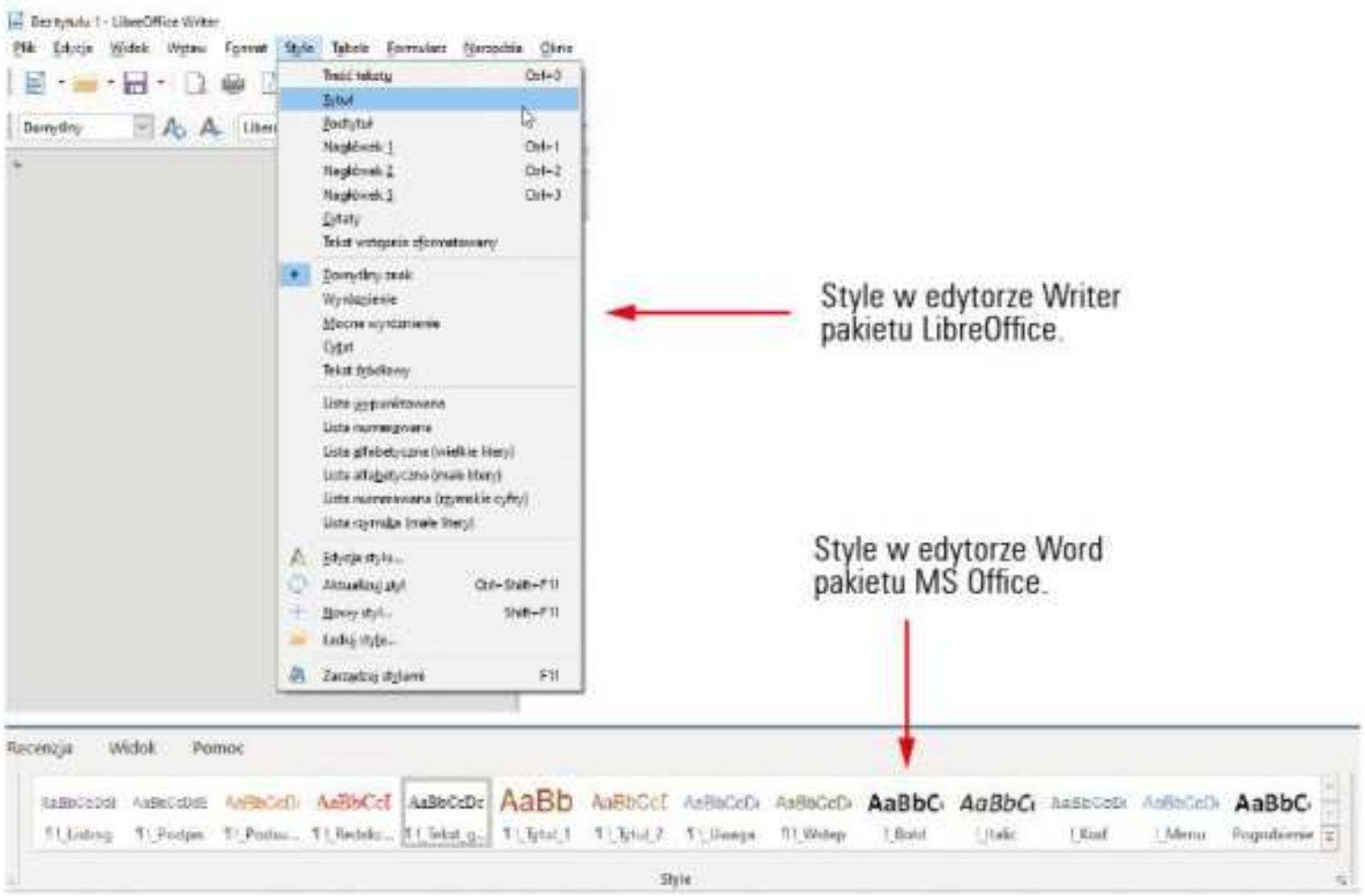
- utworzysz własny styl;
- zredagujesz dokument zawierający grafikę i style oraz na jego podstawie utworzysz własny szablon.

Wiesz już, jak korzystać z chmury informatycznej i jej programów. Umiesz edytować teksty z użyciem podstawowych narzędzi i opcji. Możesz więc tworzyć dokumenty o rozmaitym zastosowaniu. Jednym z rodzajów dokumentów tekstowych jest tutorial, czyli instrukcja opisująca kolejne kroki prowadzące do celu.

5.1. Własny styl, czyli własny pomysł na formatowanie dokumentów

tworzenie stylów

Style w edytorach tekstu są bardzo przydatnym mechanizmem edycyjnym. Pozwalają na formatowanie zaznaczonego tekstu lub akapitu z użyciem jednocześnie kilku różnych parametrów. Przydają się także w tworzeniu automatycznych spisów treści, konspektów itp. Popularne edytory pakietów biurowych, takie jak MS Office (czyt. emes ofis), LibreOffice (czyt. libreofis) itp., oferują gotowe style (rys. 5.1.).



Rys. 5.1. Opcje zmiany stylu w popularnych edytorach

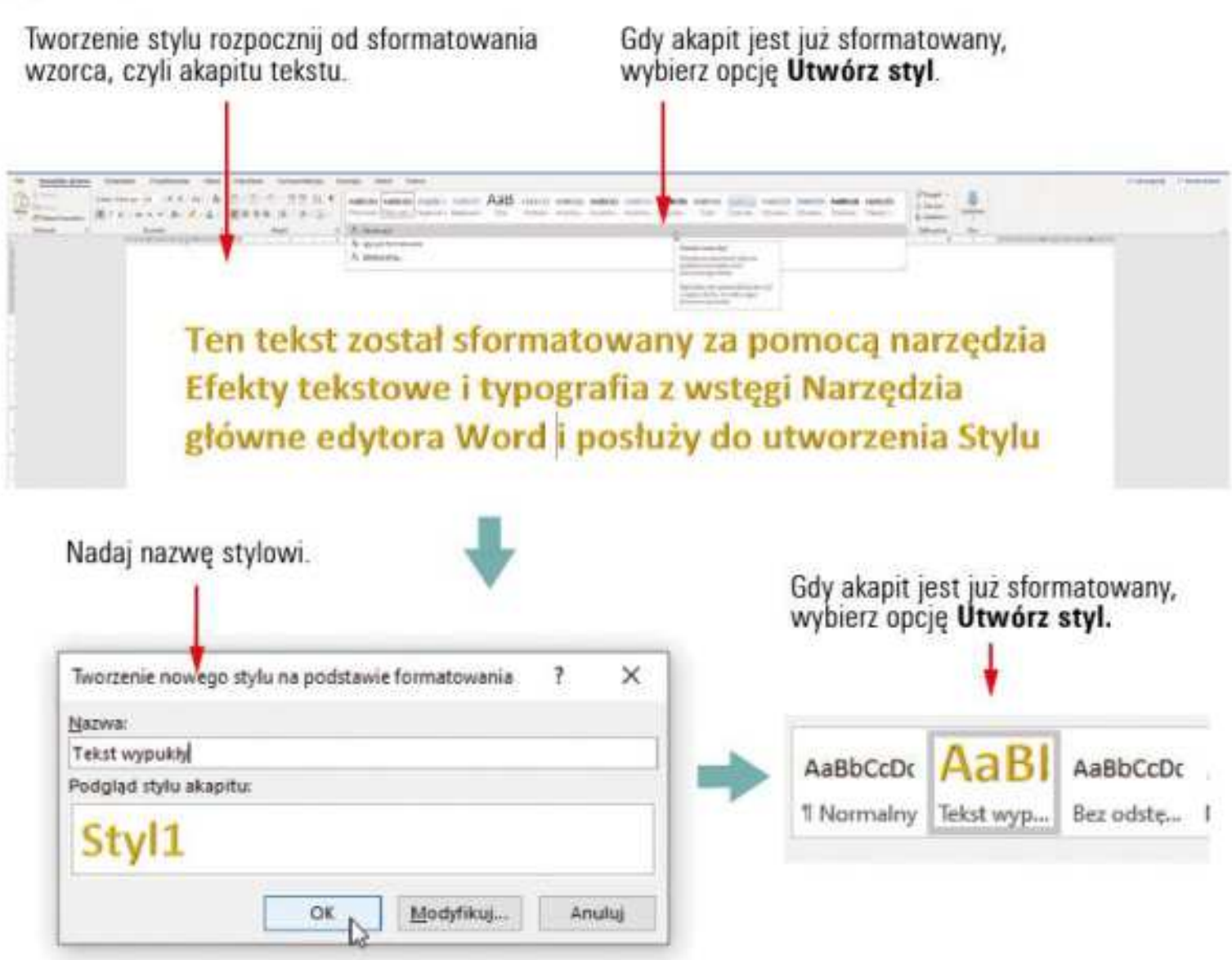
Użycie odpowiedniego stylu pomaga w utrzymaniu estetycznego wyglądu dokumentu, a także w zachowaniu takiego samego formatowania odpowiadających sobie fragmentów tekstu (rys. 5.2.).



Rys. 5.2. Przykłady użycia różnych stylów w edytorze Word

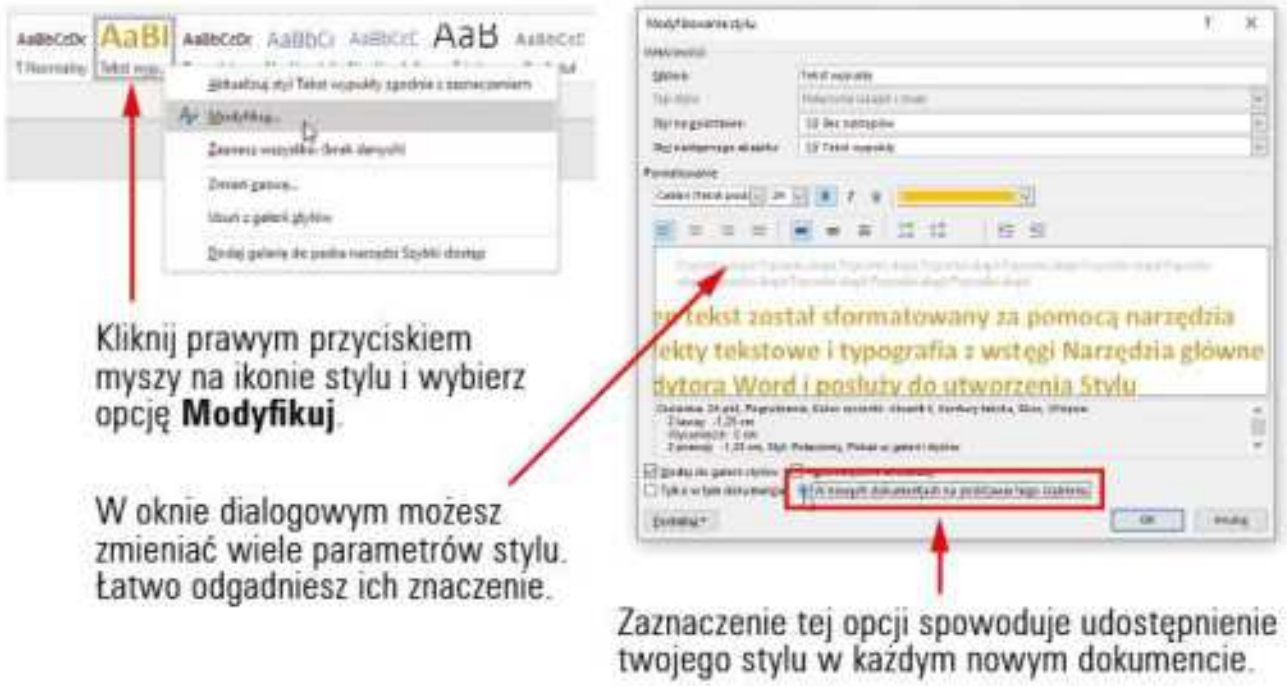
Użycie odpowiednich stylów ułatwia także import dokumentu tekstowego do innych programów, np. edytorów składu do druku.

Style udostępnione w edytorach mogą jednak nie spełniać wszystkich wymagań autora tekstu. Dlatego udostępniono opcję projektowania własnego stylu (rys. 5.3.).



Rys. 5.3. Proces tworzenia stylu w edytorze Word

Styl zaraz po utworzeniu jest dostępny jedynie w dokumencie, w którym powstał. Jeśli chcesz, by znajdował się w zestawie stylów w każdym dokumencie, użyj opcji **Modyfikuj** (rys. 5.4.).



Rys. 5.4. Proces udostępniania własnego stylu w każdym nowym dokumencie w MS Word

Tworzenie własnych stylów w innych edytorach jest równie intuicyjne i zapewne poradzisz sobie z tym wyzwaniem. Dzięki utworzeniu własnych stylów każdy dokument o podobnym przeznaczeniu może być formatowany w taki sam sposób. Skoro wiesz już, jak tworzyć własne style, czas na zaprojektowanie szablonu.

5.2. Własne szablony, czyli jak zaprojektować wygląd dokumentu

tworzenie szablonów

Jeśli planujesz tworzenie wielu dokumentów o podobnym przeznaczeniu, np. pomocy w rozwiązywaniu problemów (tutoriali), i zależy ci na rozpoznawalności twojej marki, pomocne będą własne szablony. Dzięki nim każdy z nich będzie miał oryginalny wygląd, a dokument zostanie wstępnie sformatowany. Edytory często mają wbudowane **szablony** pasujące do różnych tematów dokumentów (rys. 5.5.).

Szablony w Wordzie pogrupowano według tematyki dokumentów.
Można także pobrać szablony z zasobów sieciowych lub stworzyć własne.



Rys. 5.5. Przykłady szablonów edytora Word

Załóżmy, że nauczyciel polecił wam ułożenie programu dodającego ułamki zwykłe w języku C++. Koledzy poprosili o pomoc. Stanąłeś/stanęłaś przed wyborem: albo tłumaczyć każdemu z osobna, albo napisać tutorial dla wszystkich. Przy okazji wpadł ci do głowy pomysł polegający na pisaniu plików pomocy pokazujących rozwiązania różnych problemów informatycznych. Chcesz, by twoje dokumenty były charakterystyczne i jednoznacznie wskazywały na autora – wszak o swoje prawa i popularność trzeba dbać. Aby za każdym razem nie formatować dokumentu od nowa, należy utworzyć **szablon**.

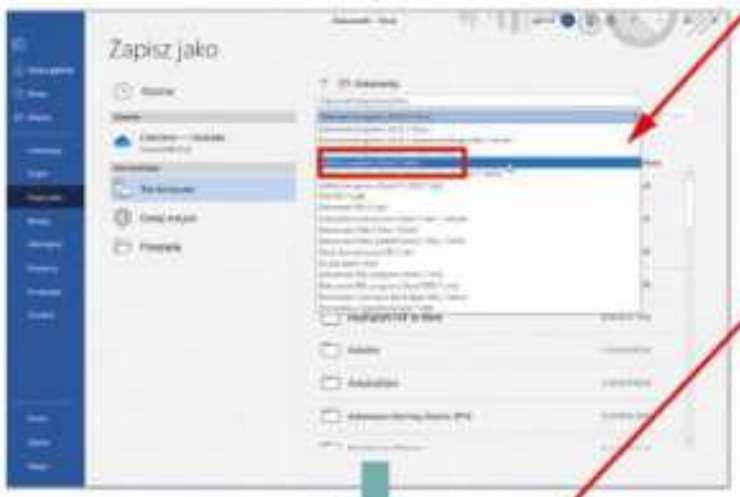
Uwaga!

Szablon jest modelem dokumentu, strony internetowej, rysunku itp., który można wypełniać treścią z zachowaniem zapisanego w nim formatowania.

Jeśli projektanci edytora Word byli konsekwentni, to nowy szablon powinno dać się utworzyć z gotowego dokumentu, podobnie jak w przypadku stylu i akapitu. Sprawdź to (rys. 5.6.) i wykonaj szablon dla swoich tutoriali.

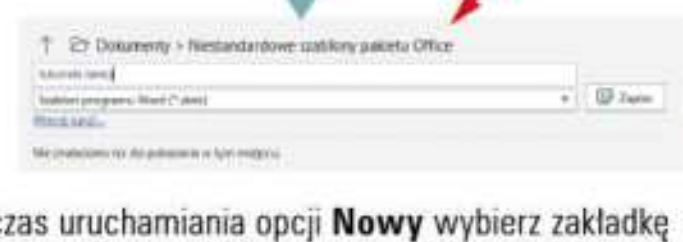


Przygotuj dokument, który stanie się podstawą utworzenia szablonu. Możesz wykorzystać nagłówek i style dostępne w edytorze.



Gdy twój wzorzec będzie gotowy, wybierz opcję **Zapisz jako** oraz format dokumentu **Szablon dokumentu Word (*.dotx)**.

Po wybraniu formatu dokumentu **(*.dotx)** automatycznie ustawi się ścieżka dostępu do folderu przeznaczonego na szablony użytkownika. Możesz ją zmienić w opcji **Opcje** i sekcji **Zapisywanie**. Wpisz nazwę swojego szablonu.





Podczas uruchamiania opcji **Nowy** wybierz zakładkę **Osobiste**. Tam znajdziesz swój szablon. Będzie on również dostępny wśród szablonów z zakładki Office.

Rys. 5.6. Proces tworzenia własnego szablonu w edytorze Word

ZADANIA DO WYKONANIA

1. Zaprojektuj własny styl, którego możesz używać do pisania prac z języka polskiego. Postaraj się dobrać odpowiednią czcionkę, jej kolor i wielkość. Wskaż, które ze stylów edytora mogą być jeszcze przydatne do pisania wypracowań.
2. Korzystając z pomocy edytora Word, znajdź opcję, w której można zmienić domyślny folder dla nowych szablonów. Zmień go na „moje szablony".
3. Utwórz dokument, który będzie podstawą do utworzenia szablonu dla tutoriali informatycznych dotyczących arkusza programu Excel. Dobierz odpowiednie ilustracje tła nagłówka i nadaj mu tytuł. Zapisz dokument w formacie szablonu i sprawdź, czy możesz z niego skorzystać, by utworzyć nowy dokument.
4. Zredaguj dokument z ilustracjami opisujący proces tworzenia stylu w edytorze LibreOffice.
5. Zredaguj dokument z ilustracjami opisujący proces tworzenia szablonu w edytorze LibreOffice. Sprawdź ścieżkę dostępu do pliku z szablonem oraz zidentyfikuj rozszerzenie jego nazwy (format pliku).

PODSUMOWANIE LEKCJI

s. 32 • **Style** tworzy się na podstawie formatowania akapitów, w których został ustawiony kursor.

s. 34 • **Szablony** tworzy się poprzez zapisanie odpowiednio przygotowanego dokumentu w formacie *.dotx.

6. Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu

NA TEJ LEKCJI:

- dowiesz się, jak korzystać z zaawansowanych opcji edytora tekstu;
- utworzysz konspekt dokumentu i spis treści, podzielisz tekst na sekcje i kolumny oraz ponumerujesz ilustracje.

Wielokrotnie korzystałeś/korzystałaś z instrukcji obsługi różnych urządzeń, programów, gier itp. Oprócz schematów, rysunków i opisów instrukcje powinny mieć odpowiednią strukturę, by łatwo było dotrzeć do interesującej nas informacji. Być może w swojej pracy napotkasz zadanie opracowania takiego dokumentu. Poćwiczyć możesz już teraz.

6.1. Konspekt, czyli pomysł na dobrą publikację

konspekt

Każdy dokument, np. wypracowanie z języka polskiego lub sprawozdanie z ćwiczenia w pracowni, powinno mieć odpowiednią strukturę i właściwą treść. By obszerny dokument był zrozumiały i precyzyjny oraz przedstawiał treści w odpowiedniej kolejności, powinien opierać się na **konspekcie**. Jest to skrócony, chronologiczny zapis najważniejszych treści przyszłego dokumentu, np. rozdziałów, podrozdziałów oraz akapitów.

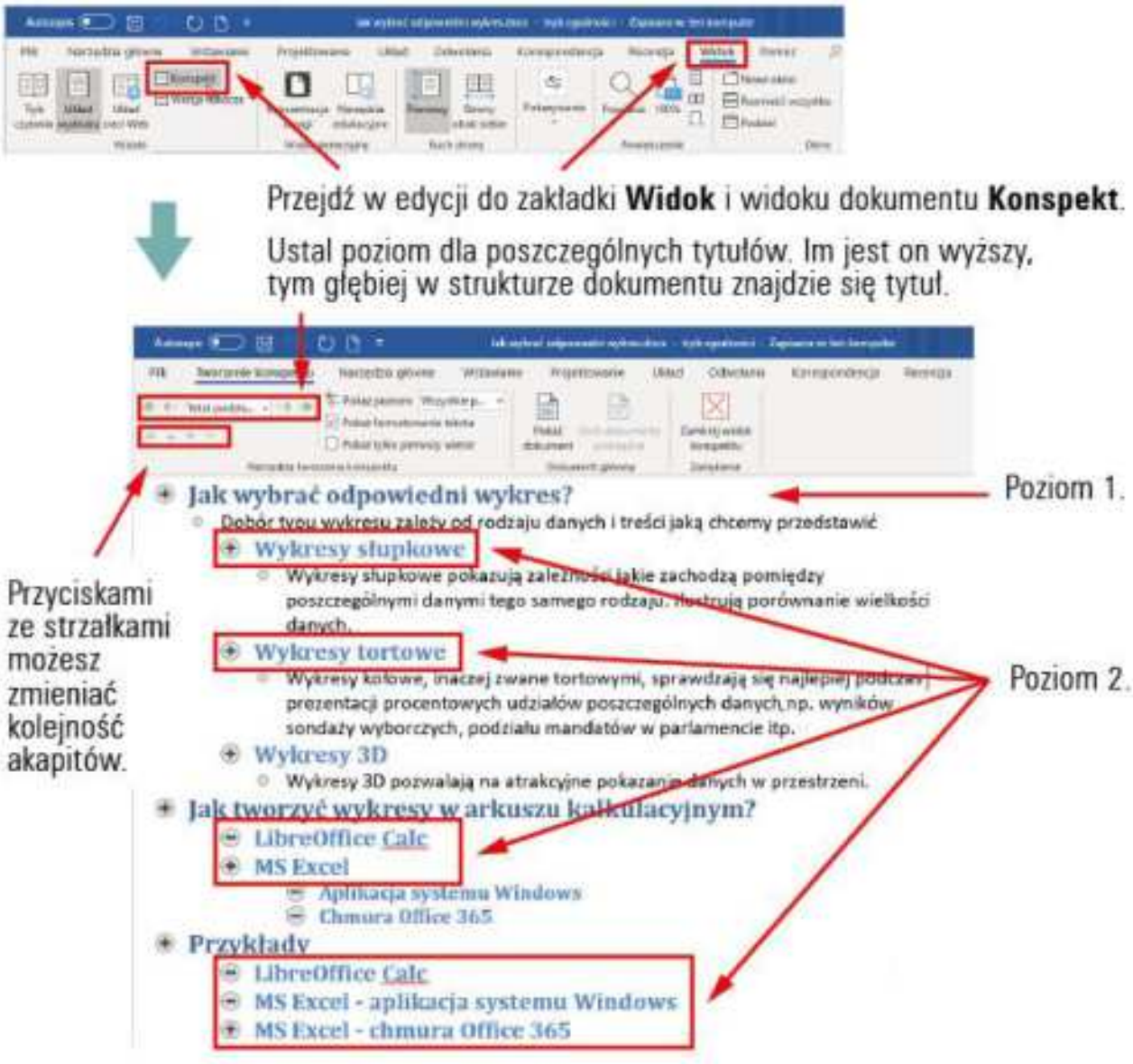
Uwaga!

Akapit często jest mylony z wcięciem w tekście. W rzeczywistości to obszar edytowanego tekstu pomiędzy wciśnięciami klawisza Enter.

akapit

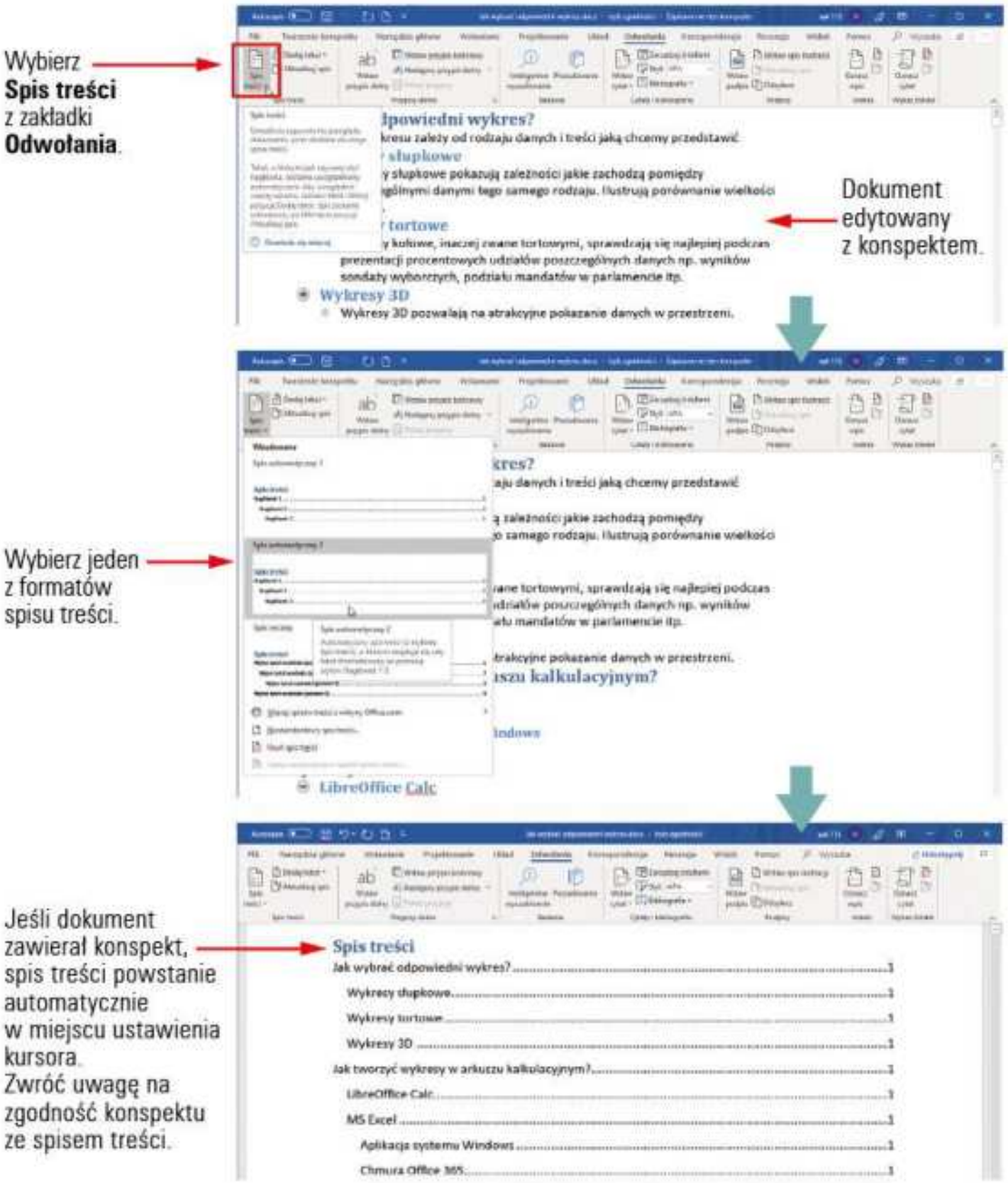
Na papierze konspekt najczęściej zapisuje się w punktach. Może być planem całej publikacji. W takim przypadku powinien powstać w czasie pracy nad koncepcją dokumentu. W edytorze można go również tworzyć w trakcie lub po zakończeniu edycji.

Załóżmy, że twoim zadaniem jest napisanie tutorialu opisującego tworzenie i wykorzystanie wykresów w arkuszu kalkulacyjnym. Zanim zabierzesz się za edycję, przemyśl jego strukturę. Zastanów się, czy będą rozdziały, podrozdziały itp. Następnie zapisz to w postaci konspektu dokumentu (rys. 6.1.). W przykładzie użyto edytora MS Word. W każdej jego wersji działania są podobne.



Rys. 6.1. Edycja konspektu dokumentu w programie Word

Tworzenie konspektów ma jeszcze jedną zaletę. Dzięki niemu można automatycznie utworzyć spis treści (rys. 6.2.). Spis powstanie w miejscu ustawienia kursora. Po każdej edycji powodującej zmianę strony, na której znajduje się tytuł, należy użyć przycisku **Aktualizuj spis**. Numery stron w spisie zostaną zaktualizowane. Niektóre edytory aktualizują spis automatycznie.



Rys. 6.2. Tworzenie spisu treści na podstawie konspektu dokumentu

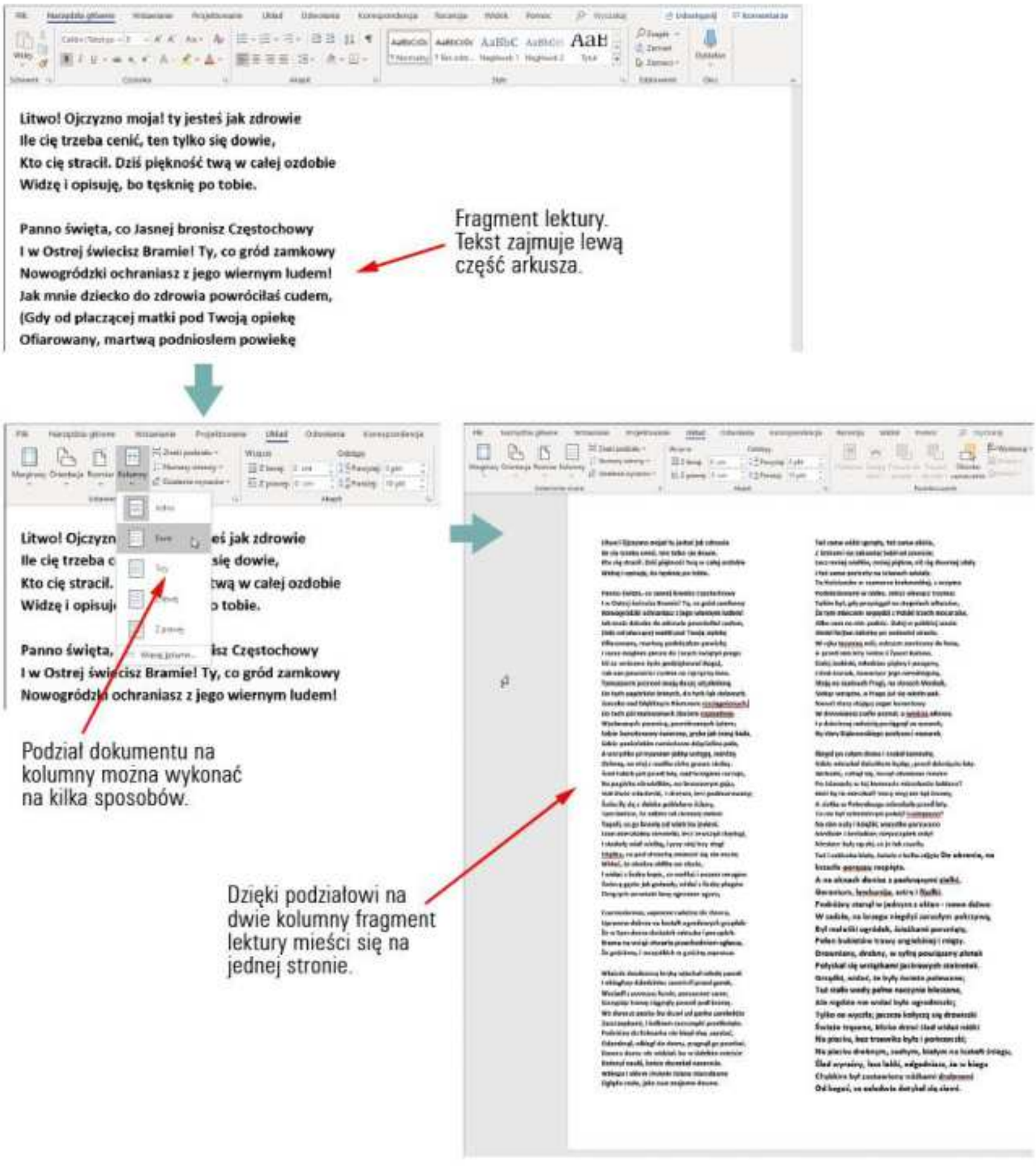
Uwaga!

Widok konspektu nie odzwierciedla wyglądu dokumentu na wydruku. Wcięcia, które widać, mają charakter informacyjny, pokazują strukturę dokumentu.

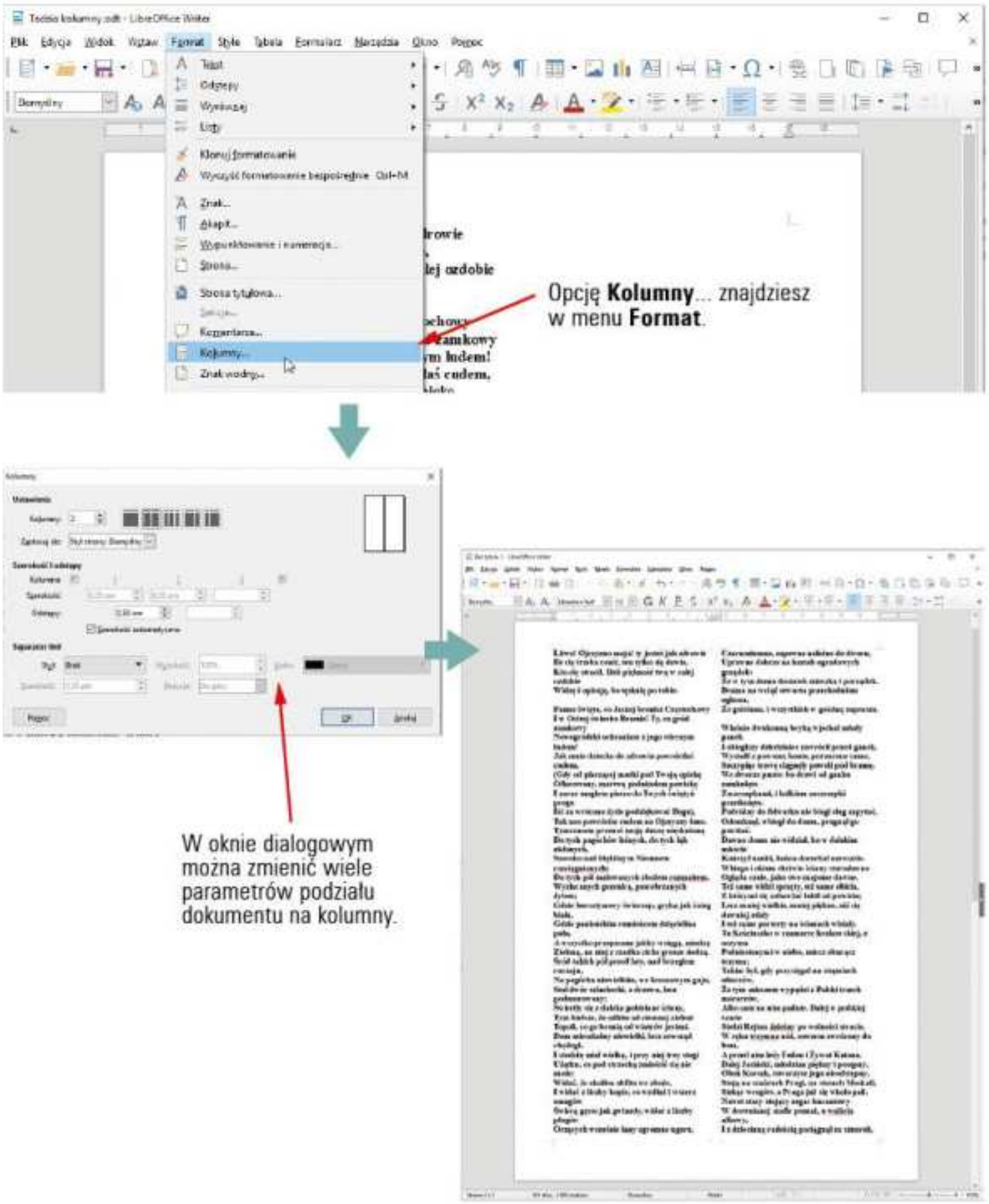
6.2. Kolumny i sekcje, czyli jak dzielić dokument tekstowy

podział tekstu na kolumny

Formatowanie tekstu nie sprawia ci trudności. Robisz to od dawna. Czy próbowałeś/próbowałaś umieszczać tekst w **kilku kolumnach**, podobnie jak to wygląda w gazetach? Większość edytorów pozwoli na taki podział dokumentu.



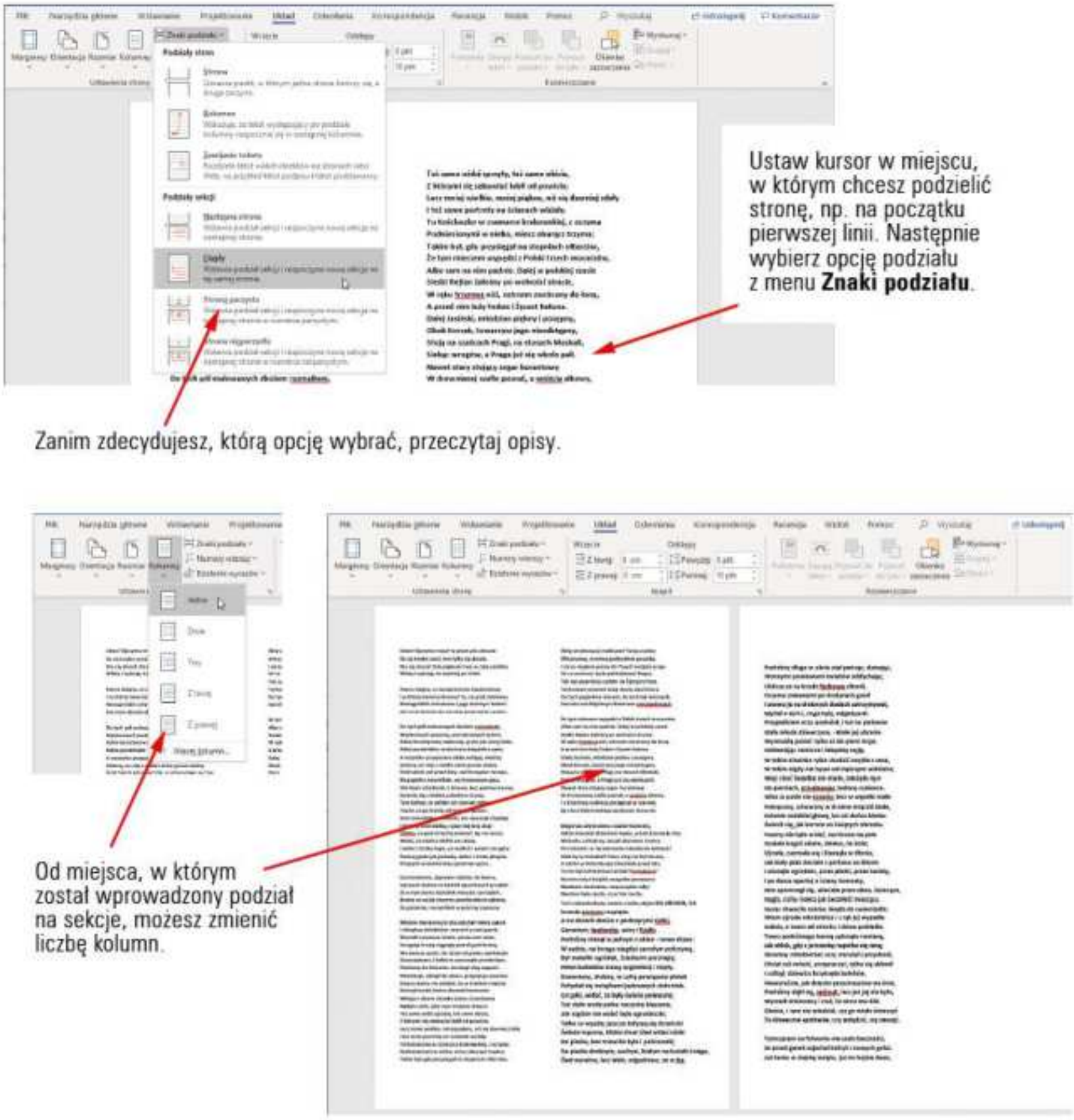
Rys. 6.3. Podział dokumentu na kolumny w edytorze Word



Rys. 6.4. Podział dokumentu na kolumny w edytorze LibreOffice Writer (czyt. rajter)

podział dokumentu na sekcje

Co zrobić w przypadku, gdy tylko część dokumentu chcemy podzielić na kolumny? Rozwiązaniem jest **podział dokumentu na sekcje** (rys. 6.5.).



Rys. 6.5. Podział dokumentu na sekcje w edytorze Word

ZADANIA DO WYKONANIA

1. Ułóż prezentację, np. w PowerPoint, w której opiszesz podział dokumentu na kolumny. Zawrzyj w niej także przykłady, w których taki podział byłby wskazany i pozytywnie wpłynąłby na cechy dokumentu.
2. Wykonaj tutorial w postaci pliku PDF opisujący podział dokumentu tekstowego na sekcje. Uwzględnij w nim różne znaki podziału. Uzasadnij użycie każdego ze sposobów podziału dokumentu.
3. Zaplanuj konspekt dokumentu zawierającego zasady postępowania w przypadku pożaru w zakładzie pracy. Wykorzystaj informacje ze szkoleń w zakładach pracy, w których odbywały się praktyki zawodowe, informacje z internetu i podręczników do BHP. Możesz wzorować się na istniejących instrukcjach. Gdy konspekt będzie już gotowy, wykonaj automatyczny spis treści.

PODSUMOWANIE LEKCJI

s. 37 • **Konspekt** dokumentu odgrywa kluczową rolę w tworzeniu spisu treści dokumentu.

s. 37 • **Akapit** jest często mylony z wcięciem w tekście. W rzeczywistości to fragment tekstu pomiędzy kolejnymi wciśnięciami klawisza Enter.

s. 39 • Dokument można edytować w wielu **kolumnach** na jednej stronie.

s. 40 • Podział dokumentu na **sekcje** pozwala oddzielić od siebie różne elementy, np. rozdziały, których tytuł zawsze będzie się znajdował na początku strony bez względu na to, w którym miejscu skończy się poprzedni rozdział.

7. Broszura, czyli jak projektować duże dokumenty

NA TEJ LEKCJI:

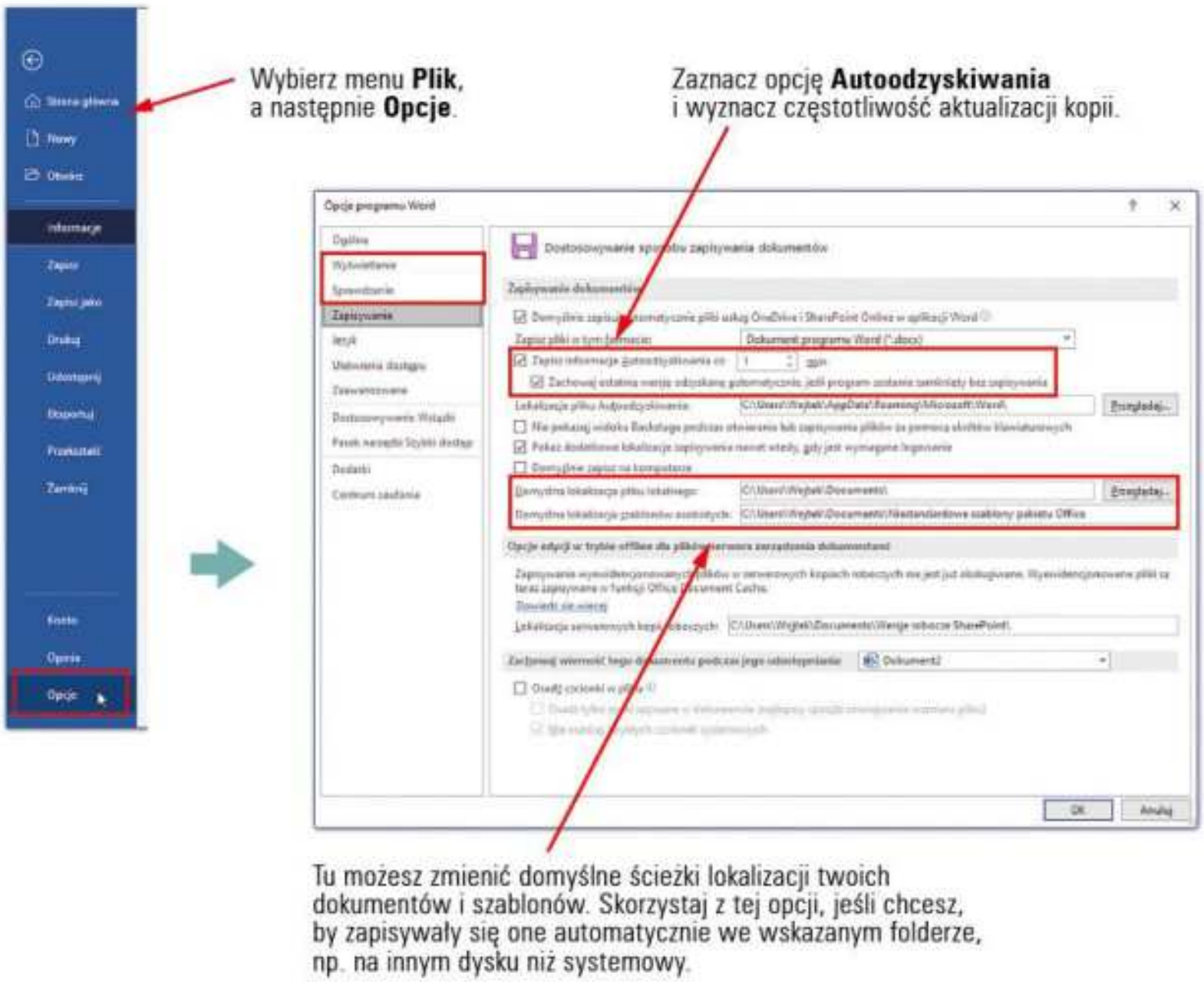
- dowiesz się, jak wykorzystać style, szablony, podział na sekcje lub kolumny w redagowaniu wielostronicowych dokumentów;
- wykonasz spisy treści, tabel i ilustracji;
- nauczysz się dobierać i wykorzystywać Kształty i SmartArty w dokumentach tekstowych;
- poznasz niektóre ważne opcje edytora, które pozwolą uniknąć utraty dokumentu i ułatwić odnalezienie pliku na dysku.

Redakcja i edycja obszernych dokumentów tekstowych staje się prostsza, gdy wykorzystasz w nich mechanizmy i zasoby edytora tekstu. Poznałeś/Poznałaś już niektóre z nich. Teraz czas na wykorzystanie ich w praktyce. Zanim jednak przystąpisz do dzieła, warto zmienić niektóre domyślne ustawienia i opcje edytora.

7.1. Wygodniej i bezpieczniej, czyli zmieniamy domyślne opcje edycji

opcje w edytorze

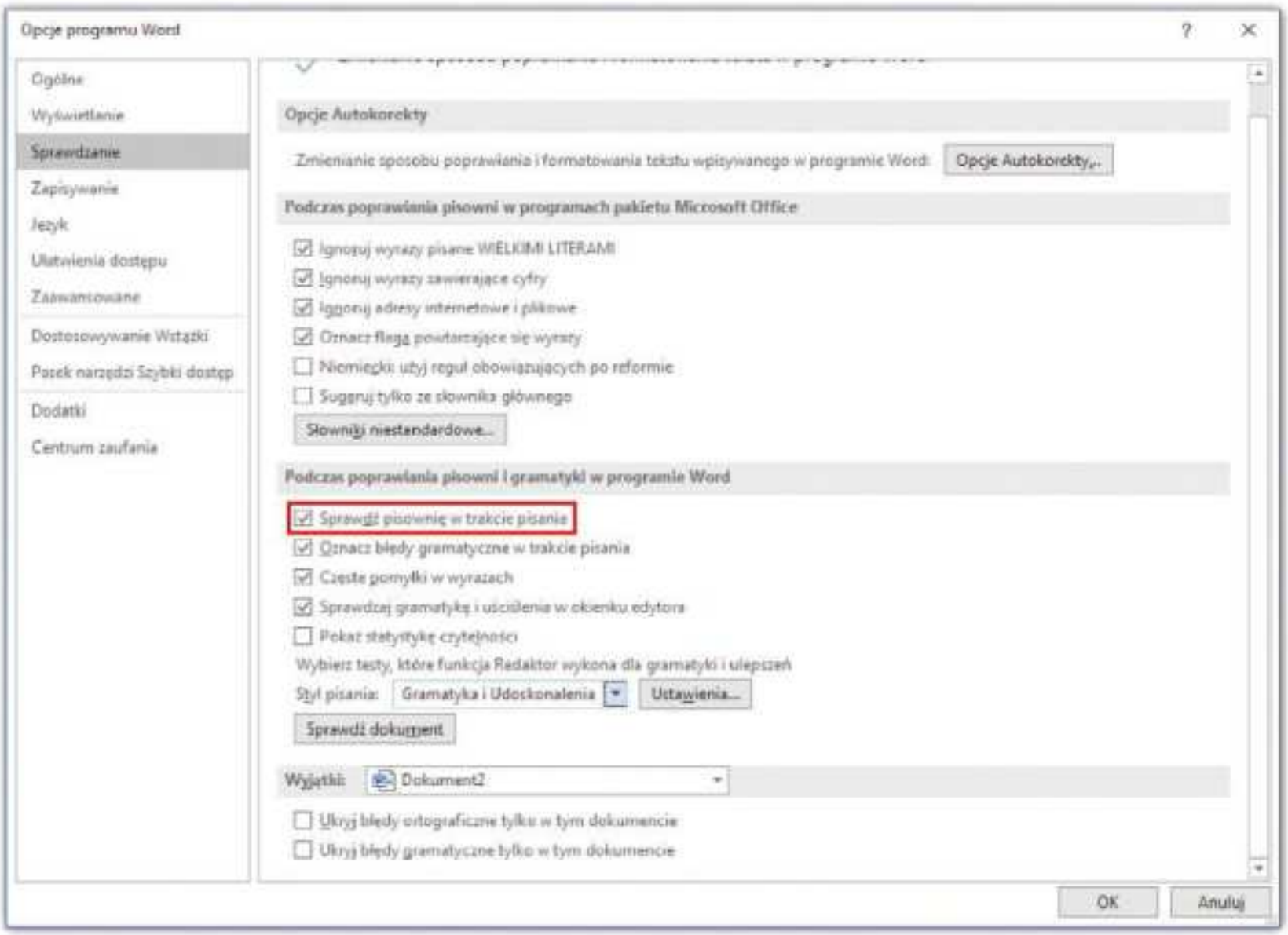
Przygotowując obszerny dokument, np. broszurę lub wydanie własnych dzieł literackich, należy zadbać o bezpieczeństwo dokumentu. Domyślne ustawienia edytora możesz sprawdzić lub zmienić w sekcji **Opcje** (rys. 7.1.). Bardzo duże znaczenie dla zmniejszenia niebezpieczeństwa utraty dokumentu ma opcja **Zapisz informacje autoodzyskiwania**. Dzięki niej kopia twojego dokumentu zostanie zaktualizowana co określony czas i w razie skasowania pliku z dokumentem będzie możliwe odzyskanie go właśnie z tej kopii. Czas zapisu i aktualizacji kopii możesz ustawić według uznania (rys. 7.1.). Zakładka **Zapisywanie** pozwala na zmianę wielu parametrów domyślnych dotyczących zachowywania plików edytora. Warto się z nimi zapoznać. Pomoc znajdziesz, wciskając klawisz **F1** (standardowe wywołanie opcji **Pomoc**).



Rys. 7.1. Zmiana domyślnych opcji zapisywania w edytorze Word

Nie mniej ważne są opcje zakładki **Sprawdzanie** (rys. 7.2.). Pozwalają one na ustalenie domyślnych parametrów korekty błędów w dokumencie, co znacznie przyspiesza edycję tekstu ze względu na brak konieczności zajmowania się poprawą ortografii lub interpunkcji. Ustawienie najlepszych dla ciebie opcji nie

będzie stanowić problemu. Wystarczy, że zrozumiesz ich nazwy. Oczywiście każdą z nich możesz wypróbować, a następnie dokonać korekty. Jeśli chcesz dodać do słownika nowy wyraz, np. specyficzny dla danej dziedziny nauki, napisz go w edytorze, zaznacz i wybierz opcję **Sprawdź dokument** z zakładki **Recenzja**. Potem kliknij wskazany przez edytor błąd w oknie **Redaktor** i użyj opcji **Dodaj do słownika**.



Rys. 7.2. Opcje sprawdzania dokumentu w edytorze Word

7.2. Kształty i SmartArt, czyli wykorzystujemy elementy graficzne dostępne w edytorach

Od dawna wiadomo, że graficzne przedstawienie procesu, danych i innych informacji jest skuteczniejsze od długich opisów słownych. W edytorach tekstu są dostępne gotowe obiekty graficzne, dzięki którym twój dokument może stać się atrakcyjniejszy i skuteczniej przekazywać treści.

używanie SmartArt i Kształtów

Nie każdy rysunek, graf lub wykres pasuje do danego dokumentu. Jego prawidłowy dobór jest istotny – decyduje o czytelności przekazu. Dla ułatwienia wyboru w edytorze Word elementy te pogrupowano według trafności zastosowania. Przejrzyj je, wybierając opcję **SmartArt** ze wstęgi **Wstawianie** (rys. 7.3.).

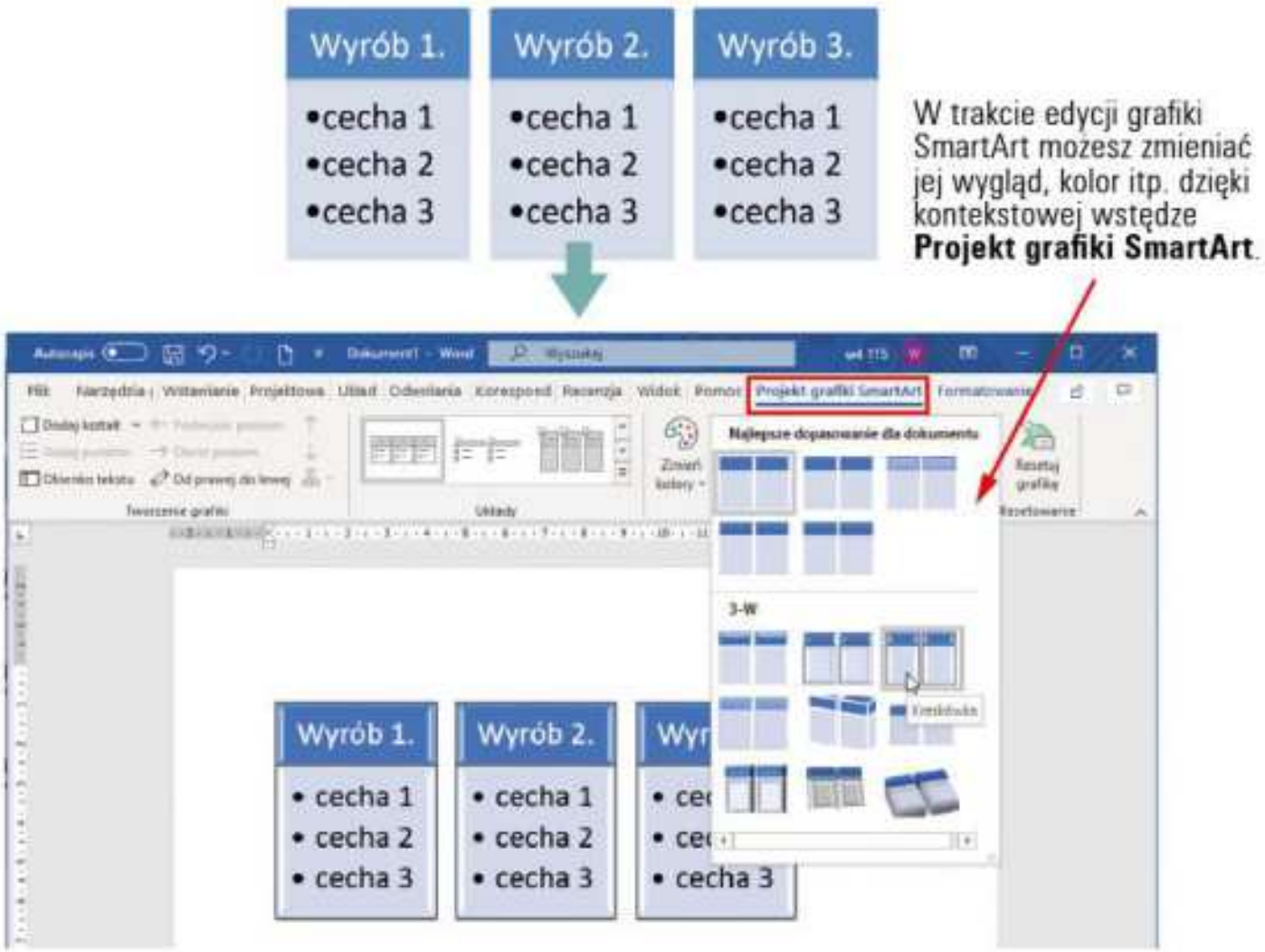


Rys. 7.3. Grupy grafik SmartArt edytora Word

Zastosowanie grafik z różnych grup

Przykład 1.

W tekście należy przedstawić trzy produkty i ich zalety w taki sposób, by łatwo było je porównać. Można to zrobić za pomocą tabeli, ale dzięki użyciu grafiki z grupy SmartArt **Lista** (rys. 7.4.) stanie się to atrakcyjniejsze i czytelniejsze.



Rys. 7.4. Użycie grafiki SmartArt z grupy **Lista**

Przykład 2.

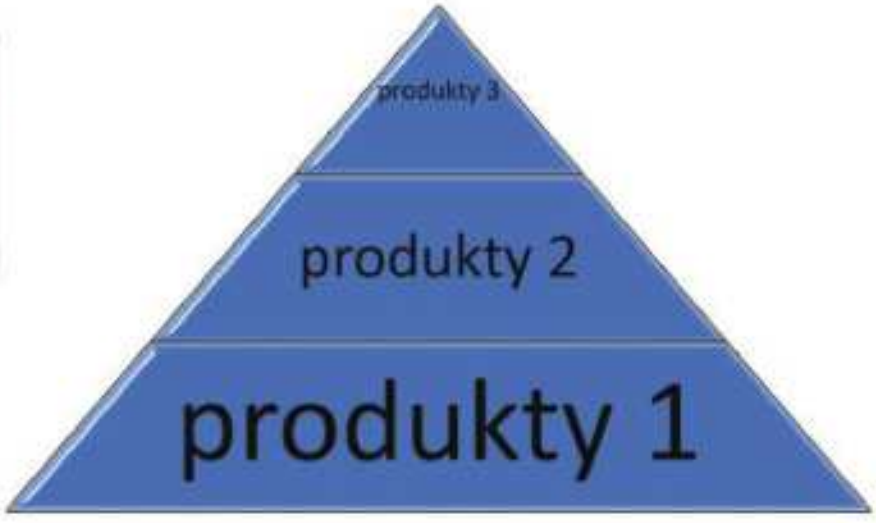
W broszurze chcesz przedstawić ewolucję danego wyrobu, np. programu komputerowego. Zatem grafika powinna prezentować pewien proces. Warto więc poszukać odpowiedniej oprawy dla tych danych w takiej grupie grafik (rys. 7.5.).



Rys. 7.5. Proces ewolucji oprogramowania przedstawiony za pomocą grafiki SmartArt z grupy Proces

Przykład 3.

Twoim zadaniem jest napisanie broszury promującej zdrowe odżywanie. W jednym z rozdziałów chcesz pokazać piramidę żywieniową. Możesz skorzystać z grafiki z grupy **Piramida** (rys. 7.6.).



Rys. 7.6. Przykład wykorzystania piramidy z grafiki SmartArt

Nie zawsze zachodzi potrzeba wstawiania grafiki SmartArt. Podczas budowania ilustracji, np. ze zrzutów ekranowych, można wykorzystać prostsze elementy graficzne z opcji **Wstawianie/Kształty** (rys. 7.7.). Wybrany kształt możesz dowolnie skalować, zmieniać jego proporcje, kolory itp. oraz ustawiać w dowolnym miejscu dokumentu. Kształty nie są traktowane jak litera.



Rys. 7.7. Obiekty graficzne z opcji **Kształty** gotowe do użycia w edytorze Word

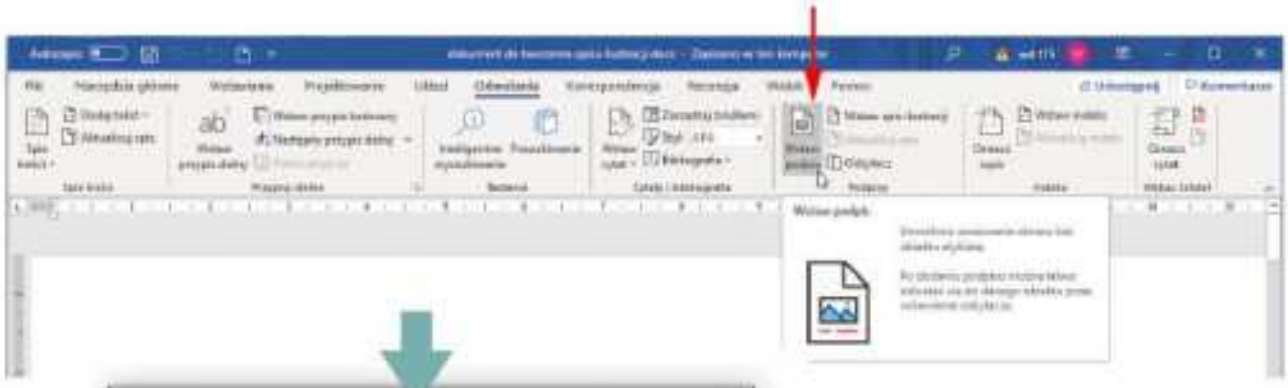
7.3. Ułatwiamy wyszukiwanie, czyli automatyczny spis ilustracji i tabel

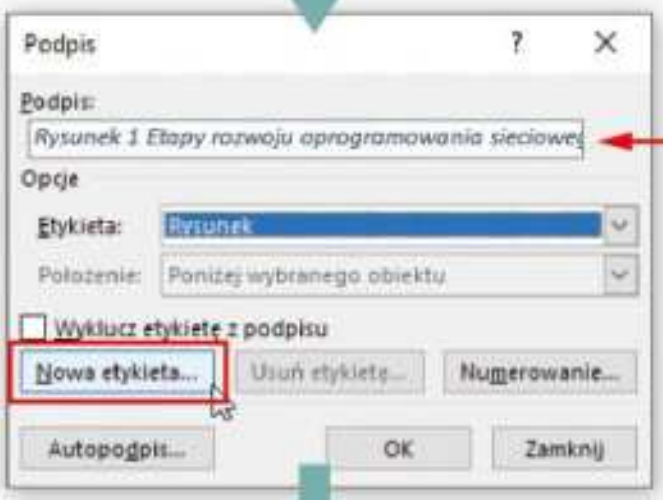
automatyzacja w edytorze

Spis ilustracji umieszczony na końcu obszernego dokumentu znacznie ułatwi znalezienie odpowiedniej informacji graficznej. Edytor Word jest wyposażony w mechanizm automatycznego wykonania spisu ilustracji. Podobnie jak w przypadku spisu treści (rozdz. 6) wymaga to pewnych zabiegów w trakcie edycji.

Wykonaj proste ćwiczenie. Korzystając ze SmartArt, utwórz kilka ilustracji. Każdą umieść na oddzielnej stronie tego samego dokumentu i podpisz (rys. 7.8.).

Ustaw kursor pod ilustracją i wybierz **Wstaw podpis** ze wstążki **Odwołania**.





W oknie dialogowym wpisz treść podpisu, który ma się znaleźć pod rysunkiem. Domyślnie ustawiono etykietę podpisu na **Rysunek**. Możesz ją zmienić na inną lub wpisać własną.

Pod rysunkiem pojawi się podpis. W taki sam sposób podpisz wszystkie rysunki w swoim dokumencie. Następnie ustaw kursor w miejscu, w którym ma się pojawić spis, i wybierz opcję **Wstaw spis ilustracji**.

Rysunek 1 Etapy rozwoju oprogramowania sieciowego

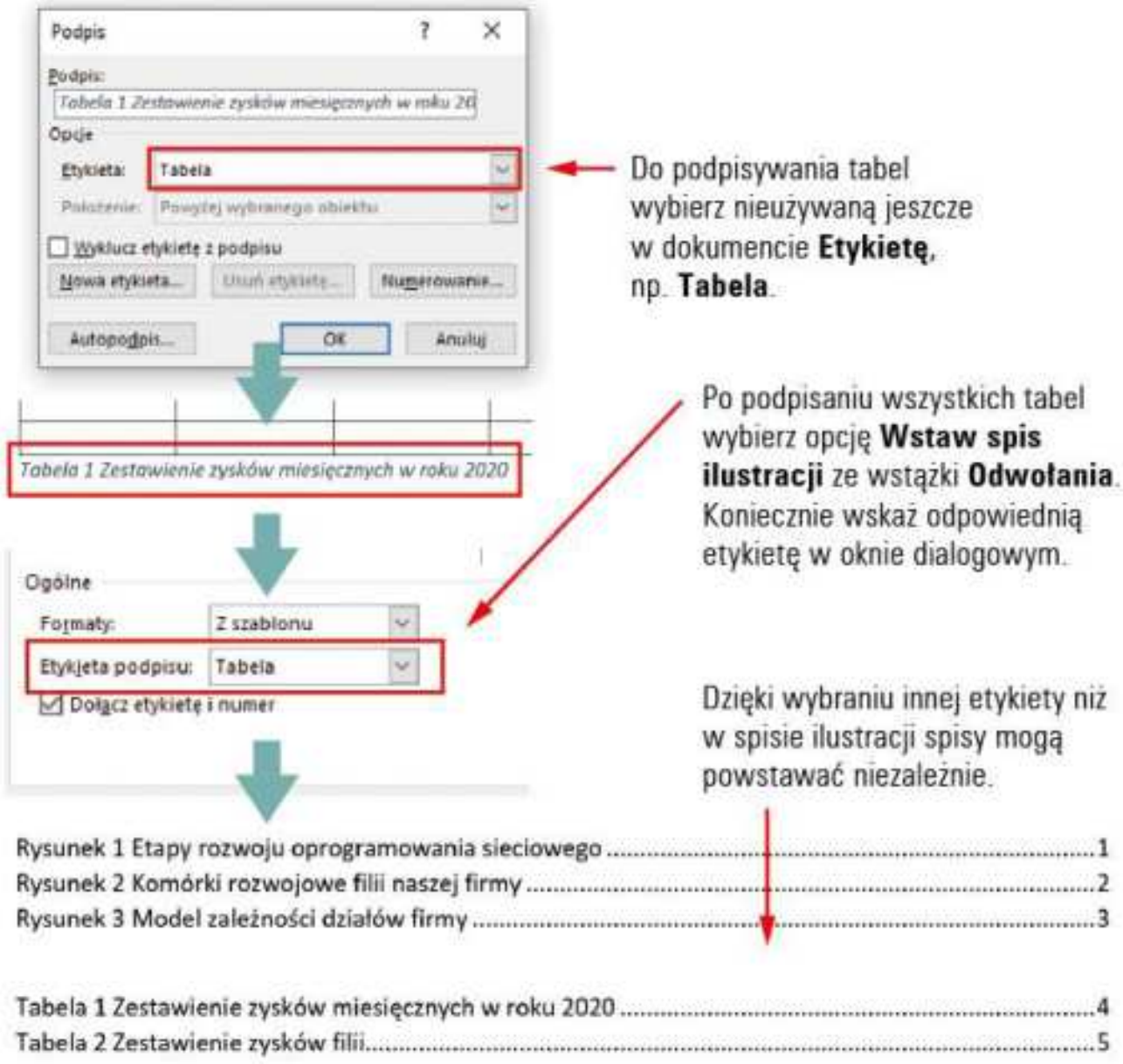


Spis ilustracji z wymienionymi numerami stron.

Rysunek 1 Etapy rozwoju oprogramowania sieciowego 1
Rysunek 2 Komórki rozwojowe filii naszej firmy 2
Rysunek 3 Model zależności działów firmy 3

Rys. 7.8. Przykład wstawiania spisu ilustracji w edytorze Word

W wielu dokumentach istotną rolę odgrywają zestawienia w postaci tabel. W takim przypadku także warto zrobić ich spis. Proces przebiega tak samo jak w przypadku tworzenia spisu ilustracji. Należy jednak wybrać inną etykietę, np. **Tabela** (rys. 7.9.).



Rys. 7.9. Proces wstawiania spisu tabel w edytorze Word

Gdy w dokumencie wprowadzisz zmiany skutkujące zmianą strony, na której znajduje się rysunek lub tabela, koniecznie użyj opcji **Aktualizuj spis** ze wstążki **Odwołania**.

Uwaga!

Dzięki opcji definiowania własnych etykiet jest możliwe wykonywanie spisów dowolnych podpisanych nimi obiektów.

7.4. Wielostronicowe dokumenty, czyli wykorzystujemy poznane mechanizmy

Wiesz już, jak utworzyć spis treści, podzielić dokument na kolumny i sekcje. Umiejętności te możesz wykorzystać do redagowania wielostronicowych dokumentów. Kieruj się przy tym pewnymi zasadami.

1. Spis treści tworzy się na podstawie konspektu. Sposób jego formatowania możesz wybrać spośród przygotowanych przez projektantów edytora.
2. Spis treści powstaje w miejscu, w którym ustawisz kursor.
3. Spis treści nie aktualizuje się automatycznie. Po istotnych zmianach w tekście musisz go odświeżyć.
4. Z mechanizmu podziału dokumentu na sekcje korzystaj wtedy, gdy każdy rozdział dokumentu chcesz zacząć od nowej strony lub jeżeli na jednej stronie chcesz zmienić strukturę dokumentu, np. 2 kolumny.
5. Podział na kolumny przyda się między innymi podczas tworzenia gazety lub czasopisma.
6. Jeśli tworzysz więcej dokumentów o wspólnych cechach, warto przygotować i zapisać własny szablon.
7. Jeśli w dokumencie często wykorzystujesz pewien sposób formatowania akapitów, utwórz własny **Styl**. Zaoszczędzisz wiele czasu podczas edycji.
8. Spisy tabel i ilustracji ułatwią czytelnikowi dotarcie do informacji.

Uwaga!

Drukując wielostronicowy dokument w postaci broszury, warto użyć opcji drukarki z mechanizmem duplex (druk na obu stronach kartki) polegającej na automatycznym drukowaniu broszur. Na każdej stronie kartki zostaną wydrukowane dwie strony. Po ułożeniu kartek w kolejności druku i złożeniu w połowie kolejność stron będzie zgodna z oryginałem.

ZADANIA DO WYKONANIA

1. Sprawdź ustawienia opcji twojego edytora tekstu. Jeśli domyślnie nie ustawiono automatycznego tworzenia kopii dokumentu, zmień to. Ustal czas, po którym będzie ona następowała.
2. Uruchom edytor tekstu, np. Word. Wpisz wyraz *print*. Dodaj go do słownika. W nowym dokumencie sprawdź, czy edytor nie wykazuje błędu.
3. Dobierz odpowiednią grafikę SmartArt do przedstawienia zawartości pewnych składników w produkcie, np. funkcjonalności w programie komputerowym. Uzasadnij swój wybór.
4. Utwórz dwukrotnie spis ilustracji i tabel z własnymi etykietami. W pierwszym przypadku ukryj etykiety w spisie, a w drugim pozwól na ich pojawienie się w podpisie.

PODSUMOWANIE LEKCJI

- **Ustawienia domyślne** edytora można zmieniać. Przykładem może być sprawdzanie błędów czy ustawienia ścieżek do folderów zapisu dokumentu itp. s. 43
- Dla wzmocnienia przekazu treści dokumentu można użyć grafiki **SmartArt** lub **Kształtów.** s. 44
- Edytując wielostronicowy dokument, warto używać mechanizmów **automatycznego** spisu treści, numerowania stron (umiejętność poznana w szkole podstawowej), podziału na sekcje itp. s. 47

8. Recenzja, czyli proponujemy poprawki w tekście

NA TEJ LEKCJI:

- dowiesz się, jak używać trybu recenzji;
- nauczysz się wprowadzać komentarze do dokumentów tekstowych.

zastosowanie trybu recenzji

Tryb recenzji daje autorom tekstu wiele możliwości jego edycji. Jest szczególnie przydatny, gdy nad jednym dokumentem pracuje zespół. Ta forma eliminuje tradycyjny sposób wprowadzania zmian czy dodawania uwag w dokumentach wydrukowanych na papierze. Zastępuje długopis, a także umożliwia przeglądanie dokumentu bez zmian, ze zmianami oraz ich zatwierdzanie. Stosowanie trybu recenzji to bardzo dobre i skuteczne narzędzie recenzentów, np. nauczycieli sprawdzających wypracowania, promotorów prac naukowych itp. Czas, abyś i ty zaczął/zaczęła z tego mechanizmu korzystać.

8.1. Recenzujemy tekst, czyli jak umieszczać komentarze w dokumencie

opcje w edytorze

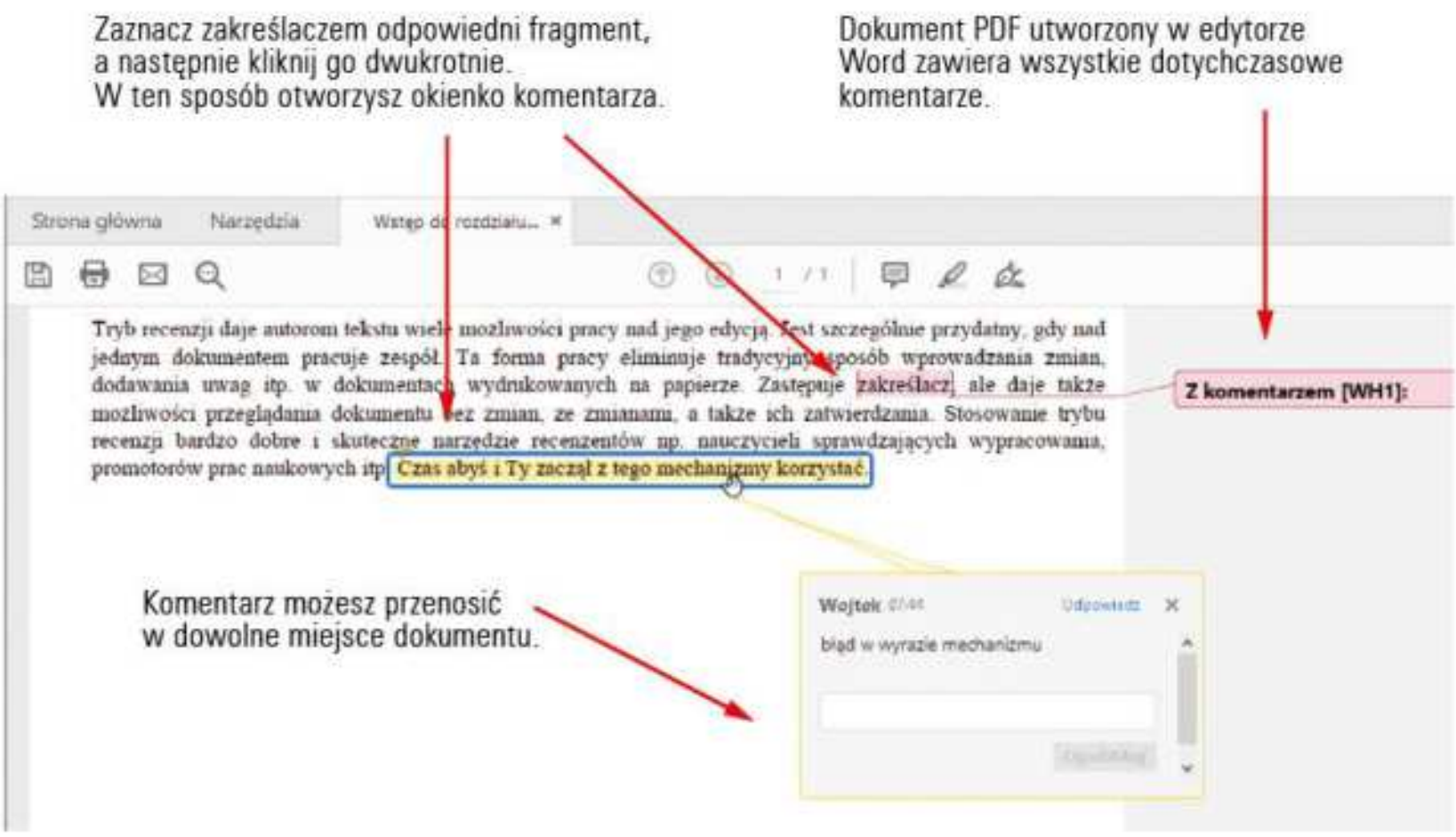
Załóżmy, że twój zespół pracuje nad pewnym projektem. Wszyscy członkowie grupy mają konta w chmurze OneDrive. Utworzyłeś/Utworzyłaś w niej dokument tekstowy. Wiesz już, że możecie redagować go w tym samym czasie, ale tym razem umówiliście się, że każdy napisze swój fragment pracy, a następnie pozostali go przeczytają i naniosą poprawki. Edytor Word zarówno w wersji chmurowej, jak i tej zainstalowanej w komputerze wyposażono w odpowiednie narzędzie, które znacznie ułatwi wam pracę. Znajdziesz je we wstążce **Recenzja**. Jest intuicyjne, więc używanie go nie sprawi kłopotów. Postępuj według opisu (rys. 8.1.).



Rys. 8.1. Wstawianie komentarzy w edytorze Word

W edytorze zainstalowanym na lokalnym komputerze opcje i narzędzia potrzebne do umieszczania komentarzy znajdziesz w tym samym miejscu. W edytorze **LibreOffice** opcja wstawiania komentarzy jest w menu **Wstaw**, a zarządzanie nimi – w menu **Edycja**.

Często się zdarza, że recenzent otrzymuje dokument w formacie PDF. Darmowy program **Adobe Acrobat Reader** także pozwala na umieszczanie komentarzy (rys. 8.2.).



Rys. 8.2. Wstawianie komentarzy do dokumentu w formacie PDF

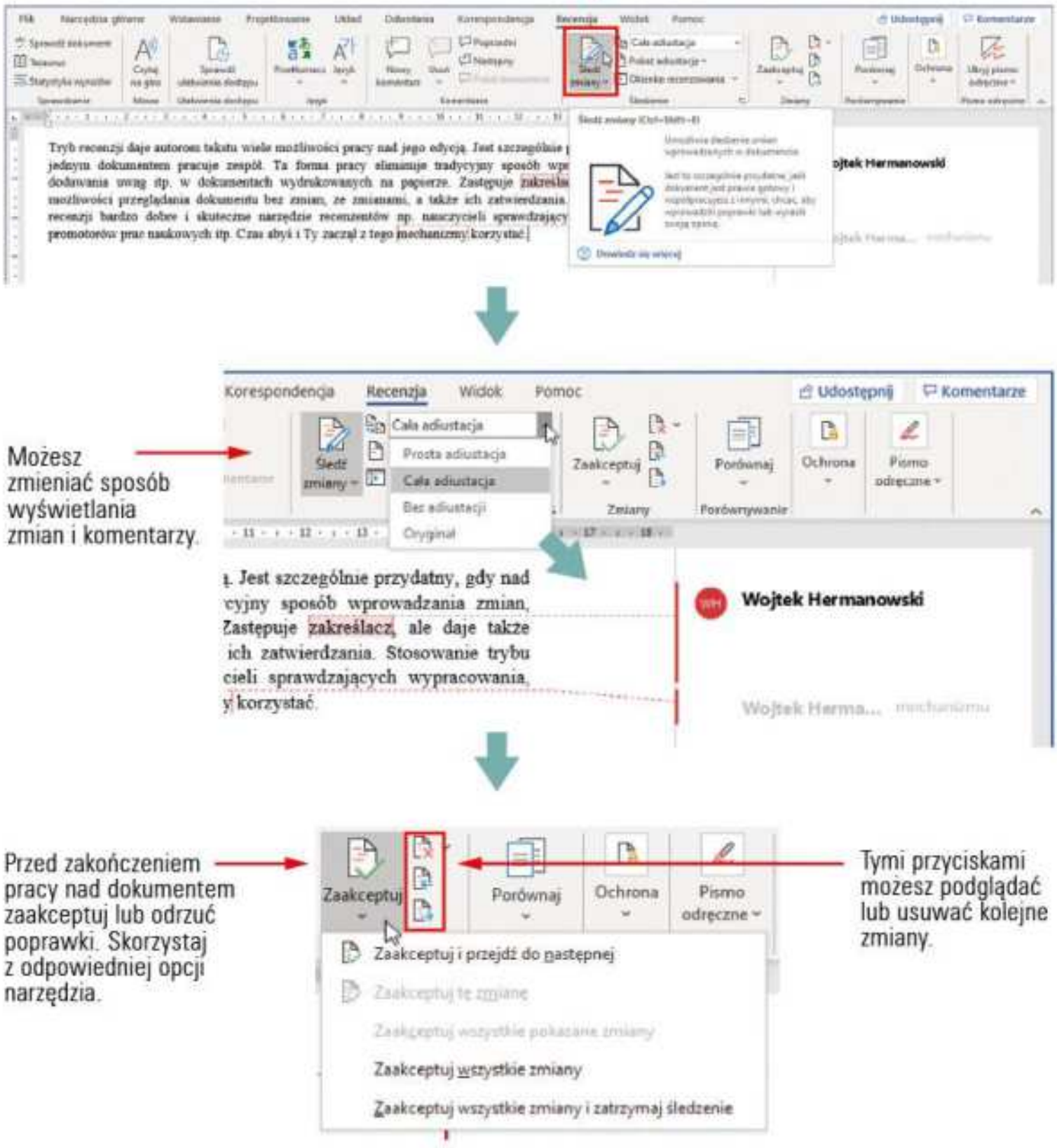
Podczas pracy w chmurze nie musisz zapisywać aktualnej wersji dokumentu, jednak jeśli pracujesz z lokalną aplikacją, pamiętaj o zachowaniu zmian.

8.2. Intensywnie edytujemy tekst, czyli śledzimy zmiany

Rzadko się zdarza, by obszerny dokument nie wymagał pewnych modyfikacji. Poprawiamy błędy, zmieniamy treść, pojedyncze wyrazy lub formatowanie. Nie zawsze okazują się one korzystne i zachodzi potrzeba powrotu do poprzedniej wersji. Jeśli nie zachowaliśmy dokumentu przed zmianami, jest to bardzo trudne. Edytory dają nam możliwość śledzenia zmian i ich zatwierdzania.

śledzenie zmian

Mechanizm **śledzenia zmian** jest intuicyjny. Należy jedynie pamiętać o jego uruchomieniu (rys. 8.3.).



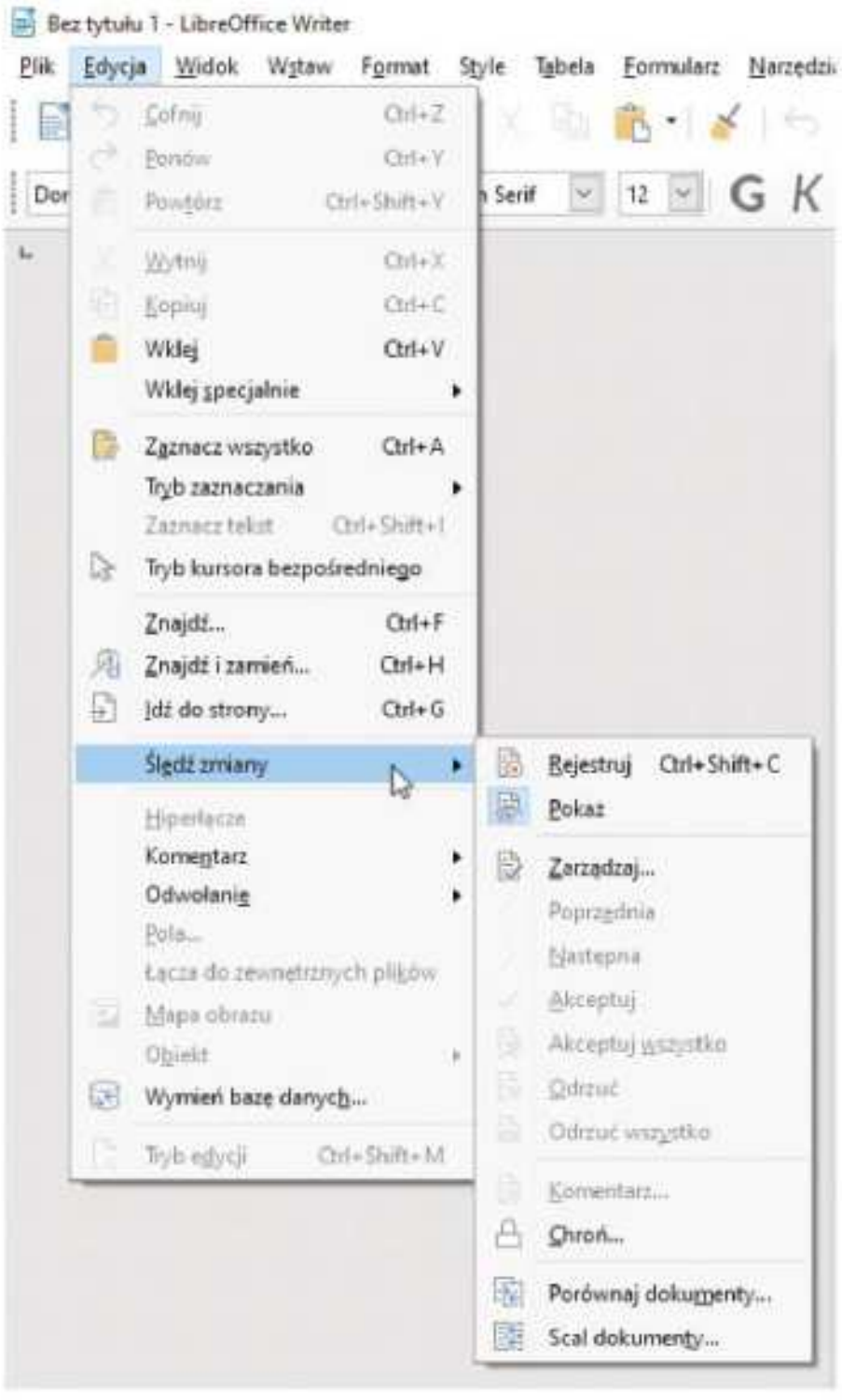
Rys. 8.3. Sposób używania mechanizmu śledzenia zmian w dokumencie w edytorze Word

Uwaga!

Adiustacja jest definiowana w różny sposób w zależności od kontekstu. Podczas edycji tekstu jest to nanoszenie zmian i korekt przez recenzenta lub autora.

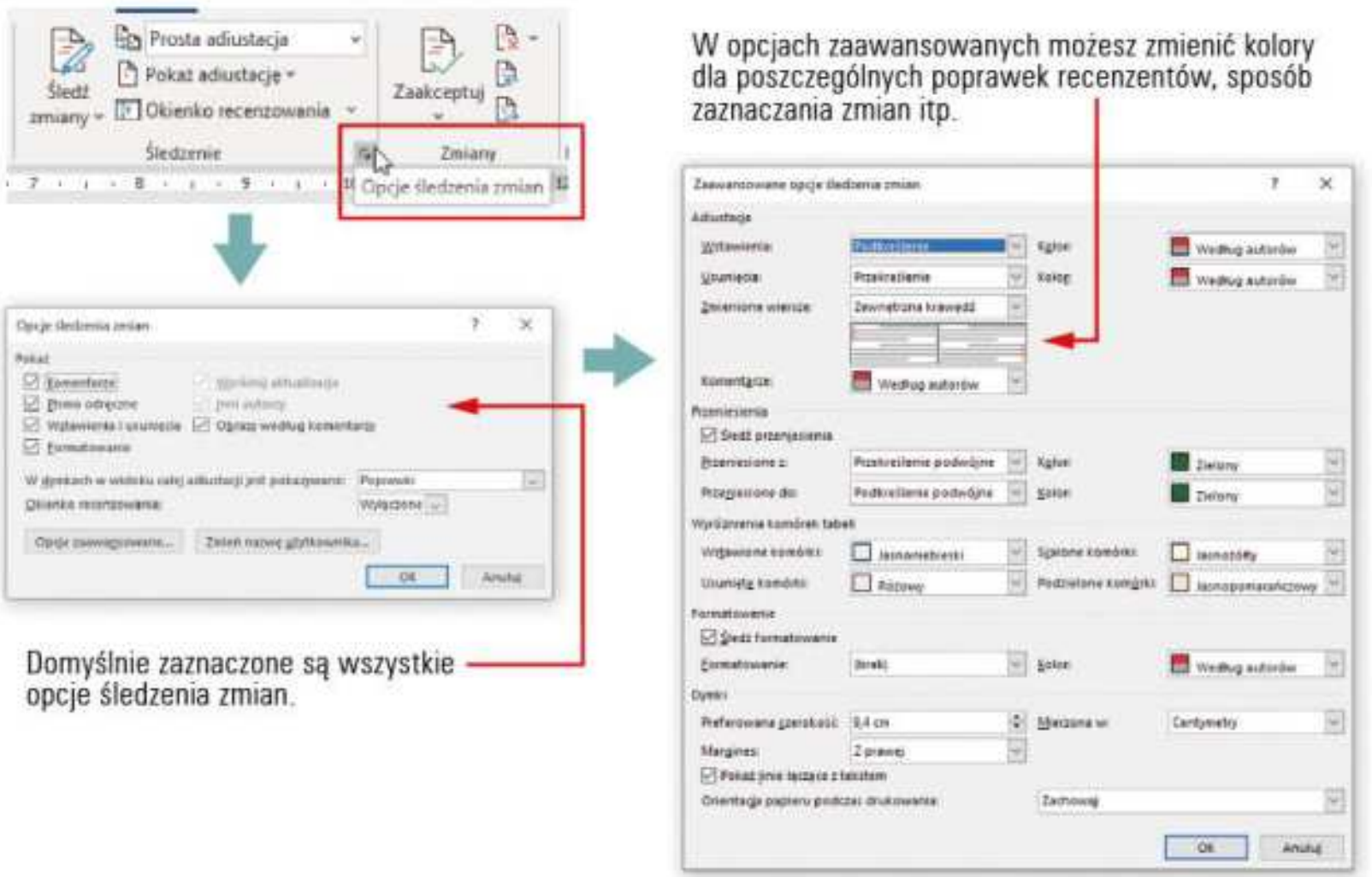
W wydawnictwach jest to redakcja techniczna, czyli proces opracowania treści pod względem **technicznym** i **graficznym**.

Inne edytory także mają wbudowany mechanizm śledzenia zmian (rys. 8.4.).



Rys. 8.4. Mechanizm śledzenia zmian w edytorze LibreOffice

Jeśli chcesz dokładniej określić zakres śledzenia zmian i sposób zaznaczania ich rodzajów, wciśnij strzałkę w narożniku sekcji **Śledzenie** wstążki **Recenzja** (rys. 8.5.).



Rys. 8.5. Opcje śledzenia zmian w dokumencie w edytorze Word

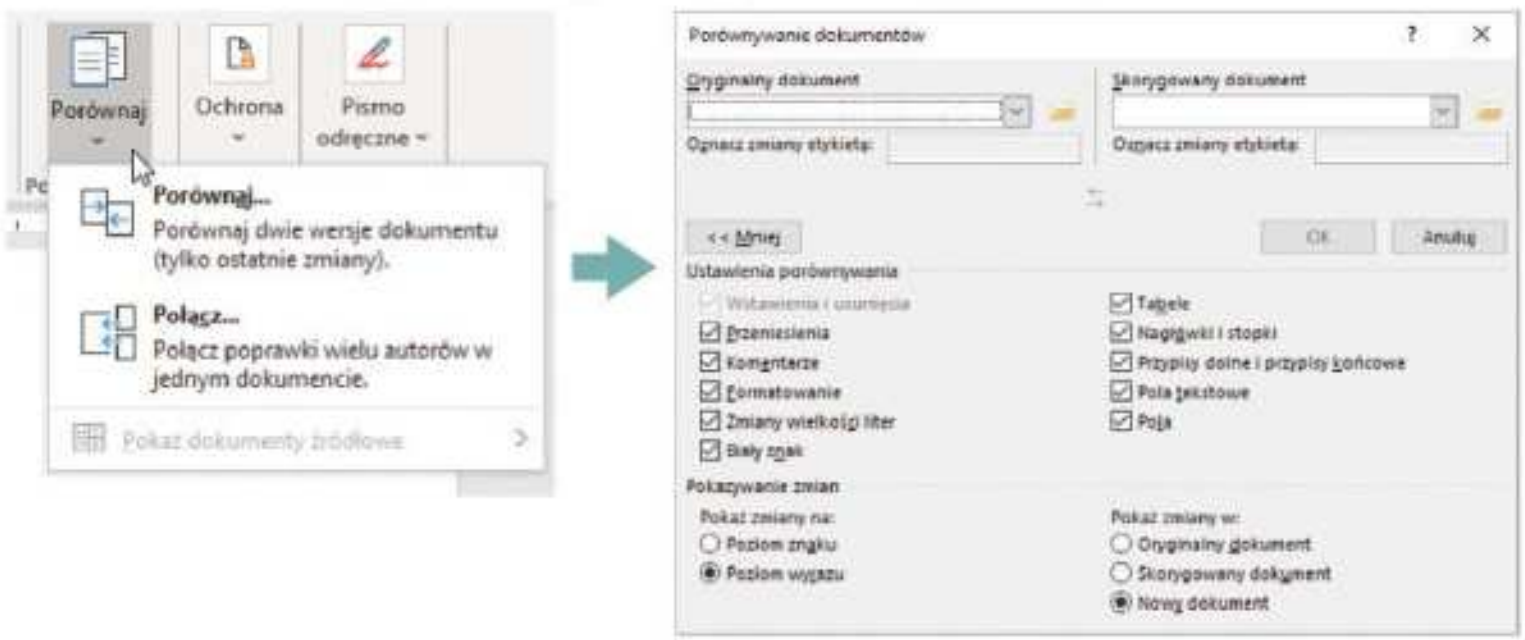
Uwaga!

Po zakończeniu pracy z tekstem w trybie śledzenia zmian należy sprawdzić, czy wszystkie właściwe poprawki zostały zaakceptowane.

8.3. Wykrywamy zmiany, czyli porównujemy dokumenty

porównywanie dokumentów

Jeśli zajdzie potrzeba porównania dwóch wersji dokumentu, możesz skorzystać z opcji **Porównaj** ze wstążki **Recenzja** (rys. 8.6.).



Rys. 8.6. Opcje porównywania dokumentów w edytorze Word

ZADANIA DO WYKONANIA

1. Utwórz dokument tekstowy w chmurze, np. OneDrive (czyt. landrajw) lub Office365. Udostępnij go koleżance lub koledze. Poproś o wprowadzenie poprawek i komentarzy. Przeczytaj je i zatwierdź lub odrzuć. Wydrukuj ostateczną wersję dokumentu i sprawdź, czy wszystkie poprawki zostały prawidłowo wprowadzone.
2. Zaloguj się do chmury (OneDrive lub Office365). Utwórz dokument tekstowy. Wstaw kilka komentarzy. Udostępnij dokument oraz poproś koleżanki i kolegów o wprowadzenie komentarzy. Pobierz go w formacie PDF. Użyj programu Acrobat Reader i sprawdź, czy komentarze są widoczne i czy zostały oznaczone nazwami autorów. Porównaj wygląd okien komentarzy i sposobów oznakowania dokumentów w chmurze i PDF.
3. Pobierz z internetu dowolny plik PDF, np. z opisem instrukcji języka C++. Wprowadź do niego komentarze, które ułatwią ci wyszukanie najważniejszych lub najtrudniejszych fragmentów tego dokumentu. Zapisz zmiany. Zaprezentuj dokument członkom swojego zespołu i poproś o ocenę przydatności i trafności twoich komentarzy.

PODSUMOWANIE LEKCJI

- **Zastosowanie trybu recenzji,** czyli użycie opcji **Komentarz** i **Śledzenie zmian,** ma szczególne znaczenie dla autorów obszernych tekstów, członków zespołów edytujących dokumenty oraz recenzentów. s. 50
- **Opcje śledzenia** możesz modyfikować. Dzięki temu dopasujesz sposób wyświetlania poprawek i komentarzy do swoich potrzeb i upodobań. Po zakończeniu pracy nad dokumentem w trybie śledzenia należy sprawdzić, czy zatwierdzono odpowiednie poprawki, a nieprzyjęte odrzucono. s. 52
- Jeśli chcesz **porównać** tylko niektóre elementy dwóch dokumentów, zaznacz je w opcjach porównania. s. 54

9. OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu

NA TEJ LEKCJI:

- dowiesz się, czym jest OLE;
- nauczysz się stosować w praktyce obiekty OLE;
- poznasz, czym jest OLE 2 i ActiveX.

Każdy, kto choć raz miał do czynienia z pakietem biurowym MS Office, wie, że jego atutem jest *object linking and embedding* (czyt. obdżekt linkin end embedin), czyli popularne OLE. Pisząc teksty w MS Word, robiąc kalkulacje w MS Excel, wysyłając i odbierając pocztę elektroniczną, przygotowując tabele i wykresy, przełączamy się pomiędzy okienkami, aby to wszystko dopracować i stworzyć jeden dokument. Korzystając z OLE, ułatwimy sobie pracę i zaoszczędzimy sporo czasu.

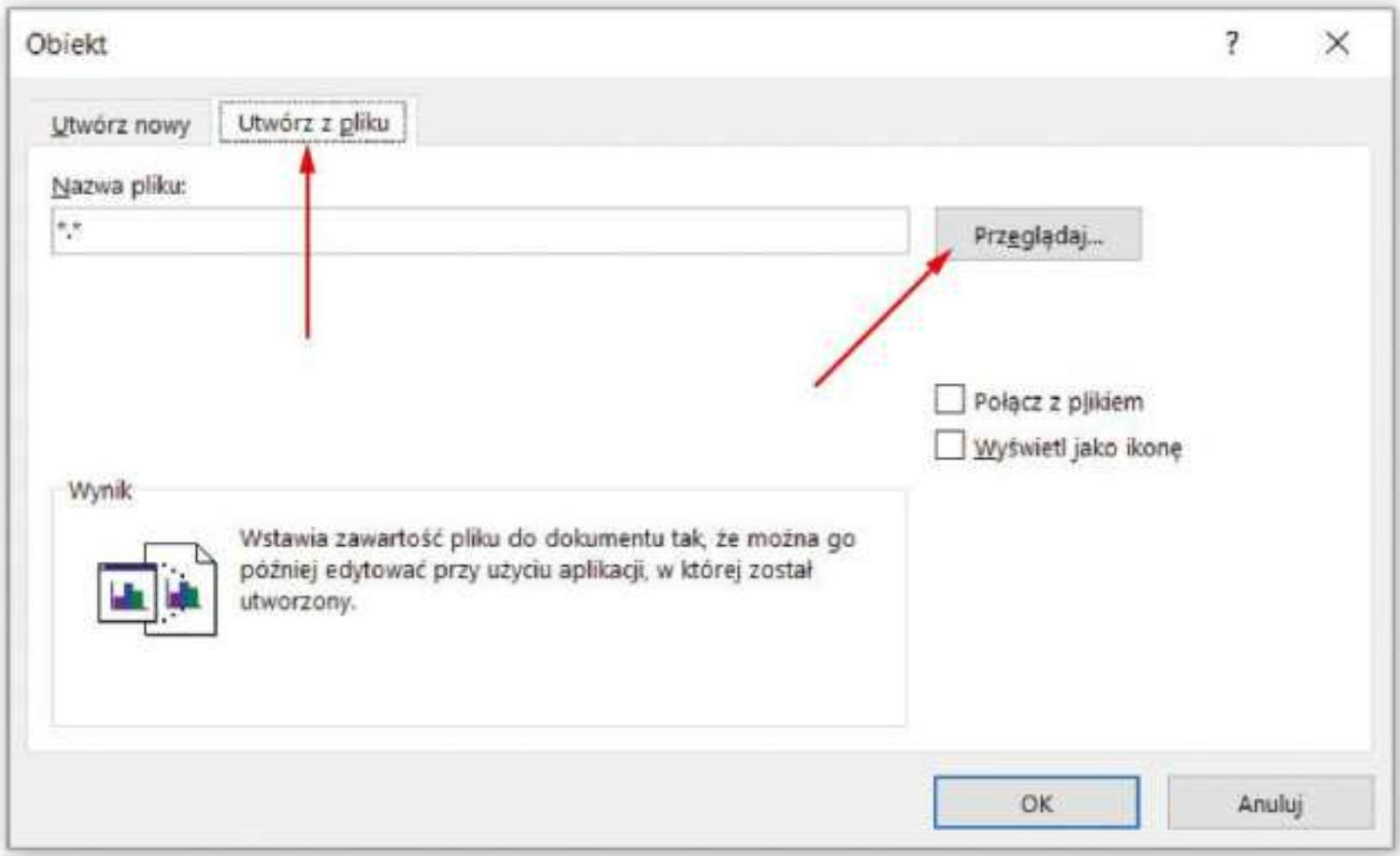
9.1. Czym jest object linking and embedding?

OLE ▸ **OLE** to technika pozwalająca na osadzanie obiektów pochodzących z różnych programów w jednym złożonym dokumencie. W takim dokumencie możemy używać funkcji, które oferują nam inne programy. Przyjmijmy, że dla swojego użytku chcemy stworzyć dokument – playlistę (czyt. plejlistę), ale taką, aby po kliknięciu danego tytułu uruchamiał się program odtwarzający wskazany utwór. Aby taką playlistę stworzyć, musimy wykonać następujące kroki (rys. 9.1.).



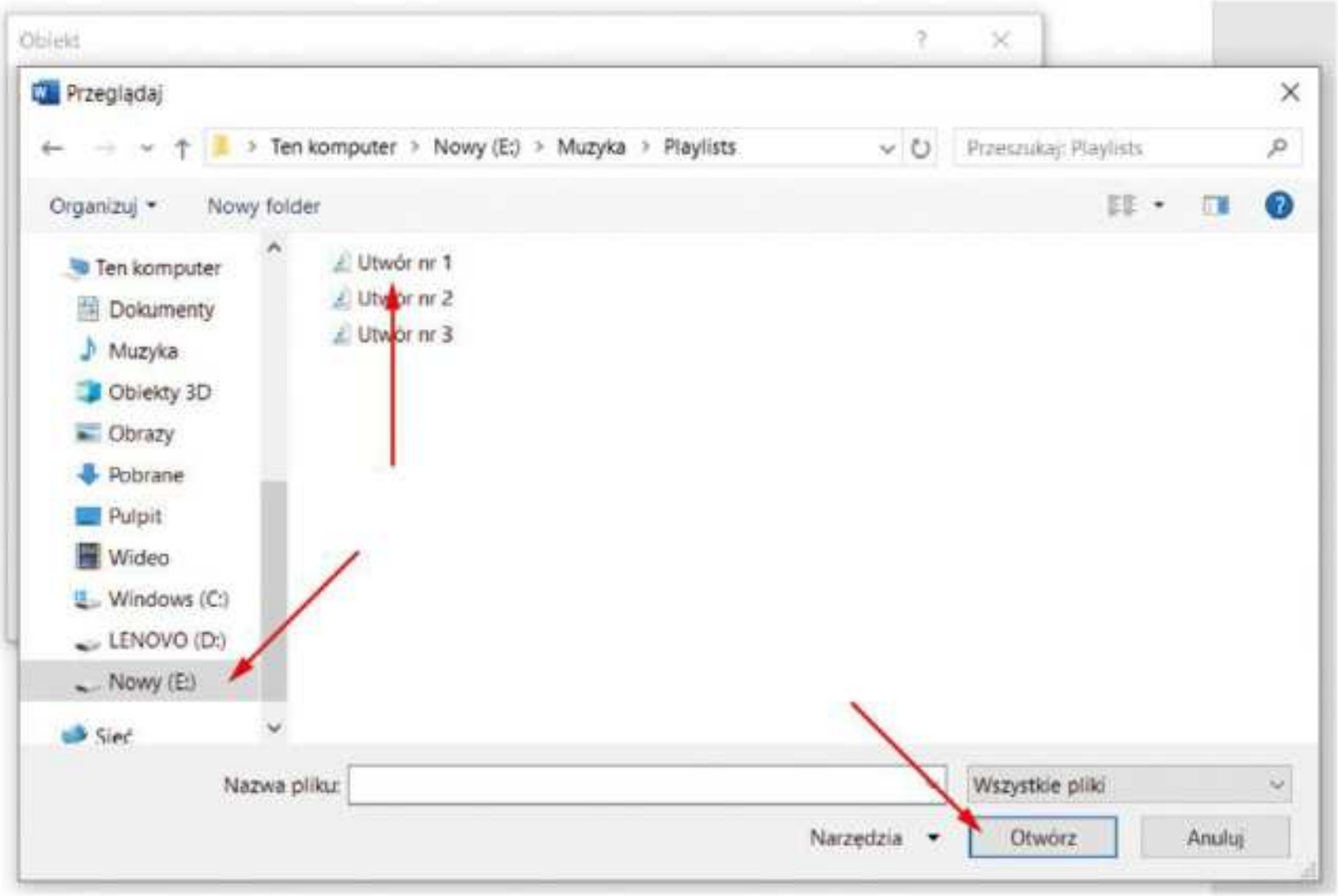
Rys. 9.1. Wybór miejsca, skąd można wstawić obiekt OLE w dokumencie MS Word

Krok 3. W oknie, które się otworzy, wybieramy zakładkę **Utwórz z pliku**, a następnie **Przeglądaj** (rys. 9.2.).



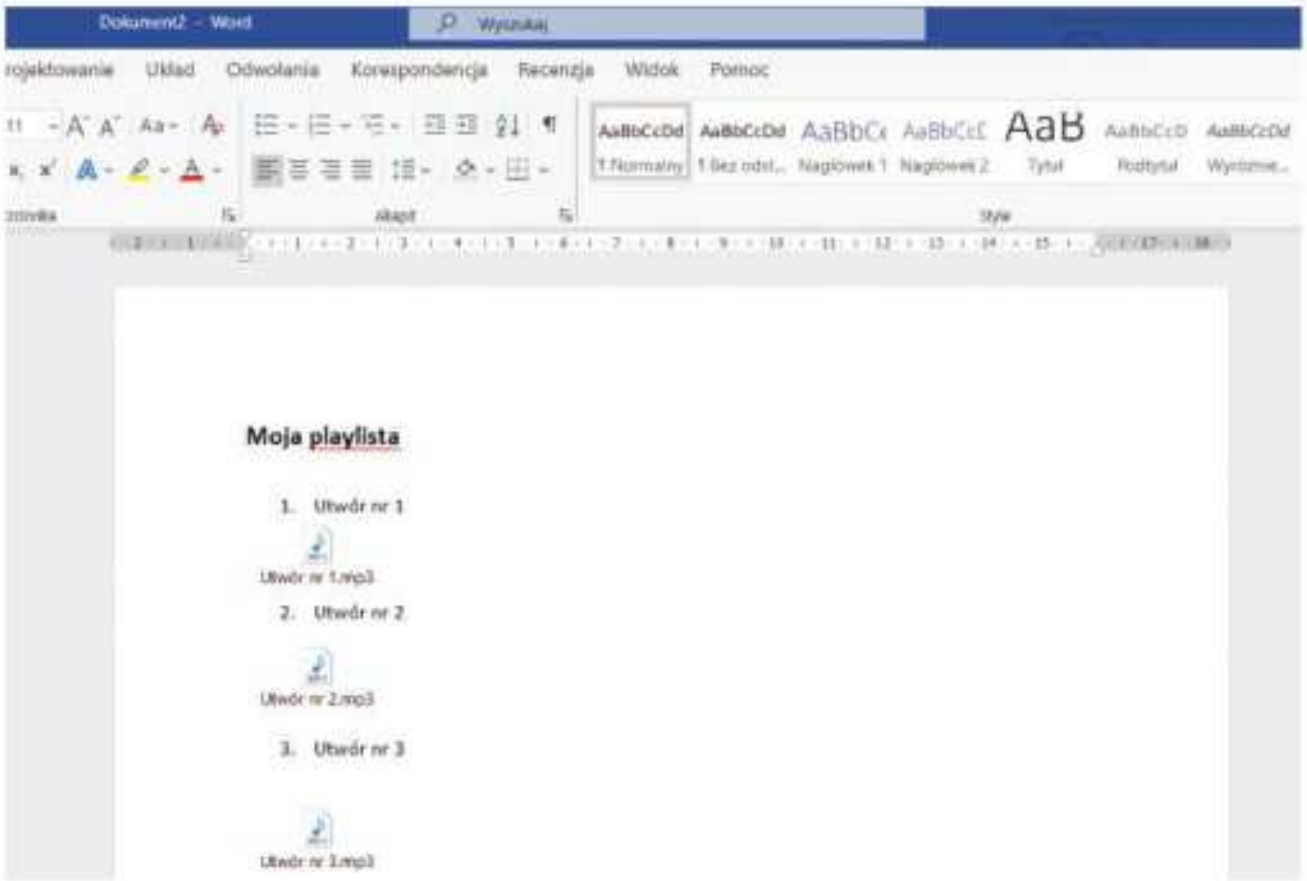
Rys. 9.2. Wybór obiektu z okna Obiekt

Krok 4. Wybieramy i wstawiamy plik muzyczny do dokumentu (rys. 9.3.).



Rys. 9.3. Wybór obiektu z dysku

W dokumencie pojawią się ikony reprezentujące pliki muzyczne. Po podpisaniu każdego pliku dokument Word może wyglądać tak jak na rysunku 9.4. Kliknięcie ikony pliku uruchomi program odtwarzający dźwięk.



Rys. 9.4. Wstawione pliki muzyczne tworzące playlistę

Każdy obiekt osadzony w dokumencie znajduje się w tzw. kontenerze OLE (*OLE container*, czyt. kontejner), który jest tworzony automatycznie podczas polecenia osadzenia obiektu. Osadzony obiekt zostanie wyświetlony tylko w obrębie kontenera, w którym się znajduje. Aplikacja, która powoduje jego uruchomienie oraz przetwarzanie (znająca jego format zapisu), jest nazywana serwerem OLE (OLE *server*, czyt. serwer).

Aplikacja Windows wspierająca technikę OLE umożliwia tworzenie kontenerów OLE w celu osadzania obiektów OLE innych aplikacji lub może być serwerem OLE, co pozwala na umieszczanie jej obiektów w innych dokumentach – może również tworzyć jedno i drugie.

Obiekty OLE mogą być osadzone w dokumencie lub tylko podłączone do dokumentu.

9.2. Przetwarzanie obiektów OLE

Przetwarzanie obiektów OLE jest możliwe w dwojaki sposób: poprzez uaktywnienie obiektu (*activation*, czyt. aktiwejszyn) lub poprzez tzw. OLE Automation (czyt. automejszyn). Aktywowanie obiektu pozwala na jego ręcznie przetwarzanie – użytkownik ma w tym czasie do dyspozycji funkcje oferowane przez serwer OLE, czyli macierzystą aplikację. W przypadku obiektu .xls (arkusz z Excela) osadzonego w formularzu użytkownik ma do dyspozycji menu Excela i funkcje dostępne w okienku formularza. OLE Automation polega na przetwarzaniu obiektu OLE w sposób programowy. Do tego celu używa się zbioru poleceń oferowanych przez serwer OLE. Klient OLE (aplikacja z kontenerem i obiektem OLE) za pośrednictwem serwera OLE i oferowanych przez niego metod może odczytywać własności obiektu oraz wprowadzać do niego nowe dane i je modyfikować, zmieniając często w ten sposób również wygląd obiektu w kontenerze OLE.

9.3. OLE 2 i ActiveX

OLE 2 ▸ W 1993 roku Microsoft wprowadził również technologię **OLE 2**, opartą na technologii COM, czyli standardzie definiowania i tworzenia interfejsów programistycznych na poziomie binarnym. OLE 2, poza osadzaniem dokumentu w innym dokumencie, umożliwia również budowanie programów i programowe przetwarzanie osadzonych obiektów – pozwala na to OLE Automation. Technologia ta ułatwia tworzenie uniwersalnych obiektów, które są pisane w dowolnych językach programowania. Od 1996 roku komponenty OLE, które są powiązane z interfejsem, otrzymały nazwę ActiveX (czyt. aktiweks).

ActiveX ▸ Technologia **ActiveX** to różnego rodzaju komponenty możliwe do użycia w różnych programach pisanych za pomocą takich języków, jak: Delphi (czyt. delfaj), Visual Basic (czyt. wizul bejzik), C++ czy obecnie bardzo popularnym Java (czyt. dżawa). ActiveX pozwala na przekazywanie danych za pomocą różnych aplikacji działających w systemie Microsoft Windows. Dzięki tej technologii programista w dużej mierze oszczędza czas, który musiałby poświęcić na pisanie własnych sposobów poruszania się i komunikowania między aplikacjami.

Niestety, ze względu na duże kłopoty z bezpieczeństwem podczas używania komponentów ActiveX przeważnie zaleca się je wyłącznie w przeglądarkach stworzonych przez Microsoft – MS Internet Explorer i Microsoft Edge.



Rys. 9.5. Logo ActiveX

ZADANIA DO WYKONANIA

1. Poszukaj w internecie informacji na temat stworzonego przez firmę Microsoft środowiska .NET. Napisz krótką notatkę w zeszycie o tym, czym charakteryzuje się to środowisko.
2. Stwórz w dowolnym programie graficznym plan dotarcia z przystanku komunikacji publicznej do twojej szkoły. Stworzony rysunek zapisz i osadź, wykorzystując obiekty OLE, w dokumencie reklamującym twoją szkołę.
3. Omów na przykładzie, jak w dokumencie MS Word umieścić wykres stworzony w dokumencie MS Excel za pomocą obiektu OLE.

PODSUMOWANIE LEKCJI

- **OLE** (ang. *object linking and embedding*) to technika pozwalająca na osadzanie obiektów pochodzących z różnych programów czy aplikacji w jednym dokumencie. ▸ s. 56
- Technologia **OLE 2** umożliwia nie tylko osadzenie obiektów z innych programów, lecz także budowanie programów i programowe przetwarzanie osadzonych obiektów. ▸ s. 58
- Technologia **ActiveX** pozwala na przekazywanie danych za pomocą różnych aplikacji napisanych np. w języku C++ lub Java, działających w systemie Microsoft Windows. ▸ s. 58

PODSUMOWANIE PRZED SPRAWDZIANEM

4. Zespół, czyli realizujemy projekty w chmurze

s. 26 ▸ • Do każdego zadania chmurę należy odpowiednio **skonfigurować** – założyć konta i sprawdzić, czy jej właściwości i zasoby są wystarczające do realizacji projektu.

s. 28 ▸ • Bardzo ważnym elementem dla **zespołu projektowego** jest utworzenie wspólnej **bazy kontaktów**.

s. 30 ▸ • Do **synchronizacji** terminów i przypominania o pracach do wykonania można wykorzystać kalendarz chmury. Można go także synchronizować z kalendarzem z telefonu komórkowego, co znacznie ułatwia powiadomienia.

5. Tutorial, czyli jak tworzyć pomoce i instrukcje obsługi

s. 32 ▸ • **Style** tworzy się na podstawie formatowania akapitów, w których został ustawiony kursor.

s. 34 ▸ • **Szablony** tworzy się poprzez zapisanie odpowiednio przygotowanego dokumentu w formacie *.dotx.

6. Rozbudowana struktura, czyli korzystamy z konspektu w edytorze tekstu

s. 37 ▸ • **Konspekt** dokumentu odgrywa kluczową rolę w tworzeniu spisu treści dokumentu.

s. 37 ▸ • **Akapit** jest często mylony z wcięciem w tekście. W rzeczywistości to fragment tekstu pomiędzy kolejnymi wciśnięciami klawisza Enter.

s. 39 ▸ • Dokument można edytować w wielu **kolumnach** na jednej stronie.

s. 40 ▸ • Podział dokumentu na **sekcje** pozwala oddzielić od siebie różne elementy, np. rozdziały, których tytuł zawsze będzie się znajdował na początku strony bez względu na to, w którym miejscu skończy się poprzedni rozdział.

7. Broszura, czyli jak projektować duże dokumenty

s. 43 ▸ • **Opcje domyślne** edytora można zmieniać. Przykładem może być sprawdzanie błędów czy ustawienia ścieżek do folderów zapisu dokumentu itp.

s. 44 ▸ • Dla wzmocnienia przekazu treści dokumentu można użyć grafiki **SmartArt** lub **Kształtów.**

s. 47 ▸ • Edytując wielostronicowy dokument, warto używać mechanizmów **automatycznego** spisu treści, numerowania stron (umiejętność poznana w szkole podstawowej), podziału na sekcje itp.

8. Recenzja, czyli proponujemy poprawki w tekście

s. 50 ▸ • **Zastosowanie trybu recenzji,** czyli użycie opcji **Komentarz** i **Śledzenie zmian,** ma szczególne znaczenie dla autorów obszernych tekstów, członków zespołów edytujących dokumenty oraz recenzentów.

s. 52 ▸ • **Opcje śledzenia** możesz modyfikować. Dzięki temu dopasujesz sposób wyświetlania poprawek i komentarzy do swoich potrzeb i upodobań. Po zakończeniu pracy nad dokumentem w trybie śledzenia należy sprawdzić, czy zatwierdzono odpowiednie poprawki, a nieprzyjęte odrzucono.

s. 54 ▸ • Jeśli chcesz **porównać** tylko niektóre elementy dwóch dokumentów, zaznacz je w opcjach porównania.

9. OLE, czyli łączymy dokumenty w dokumentacji pracy zespołu

s. 56 ▸ • **OLE** (ang. *object linking and embedding*) to technika pozwalająca na osadzanie obiektów pochodzących z różnych programów czy aplikacji w jednym dokumencie.

s. 58 ▸ • Technologia **OLE 2** umożliwia nie tylko osadzenie obiektów z innych programów, lecz także budowanie programów i programowe przetwarzanie osadzonych obiektów.

s. 58 ▸ • Technologia **ActiveX** pozwala na przekazywanie danych za pomocą różnych aplikacji napisanych np. w języku C++ lub Java, działających w systemie Microsoft Windows.



III. Rozwijanie kompetencji społecznych



- Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony
- Platformy uczą, czyli rola e-learningu w nauce i pracy
- Możesz być administratorem, czyli jak zarządzać platformą e-learningową
- Zasoby i testy, czyli wypełniamy kursy treścią
- Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach do e-learningu

10. Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony

NA TEJ LEKCJI:

- dowiesz się, jak wykorzystać smartfon do ratowania życia, monitorowania stanu zdrowia, tłumaczenia tekstów;
- sprawdzisz, jak działają wersje stron dla niedowidzących;
- poznasz niektóre funkcje urządzeń informatycznych dla niepełnosprawnych.

Informatyka to nie tylko edytory, gry i internet. Wykorzystując możliwości współczesnych komputerów, smartfonów i sieci, można także ulżyć niepełnosprawnym, pomóc w ratowaniu życia i zdrowia, monitorować stan organizmu, odnajdować zaginionych itp. Warto poznać przykłady rozwiązań informatycznych niosących codzienną pomoc.

10.1. Warto zainstalować, czyli aplikacje ratujące życie

aplikacje pierwszej pomocy

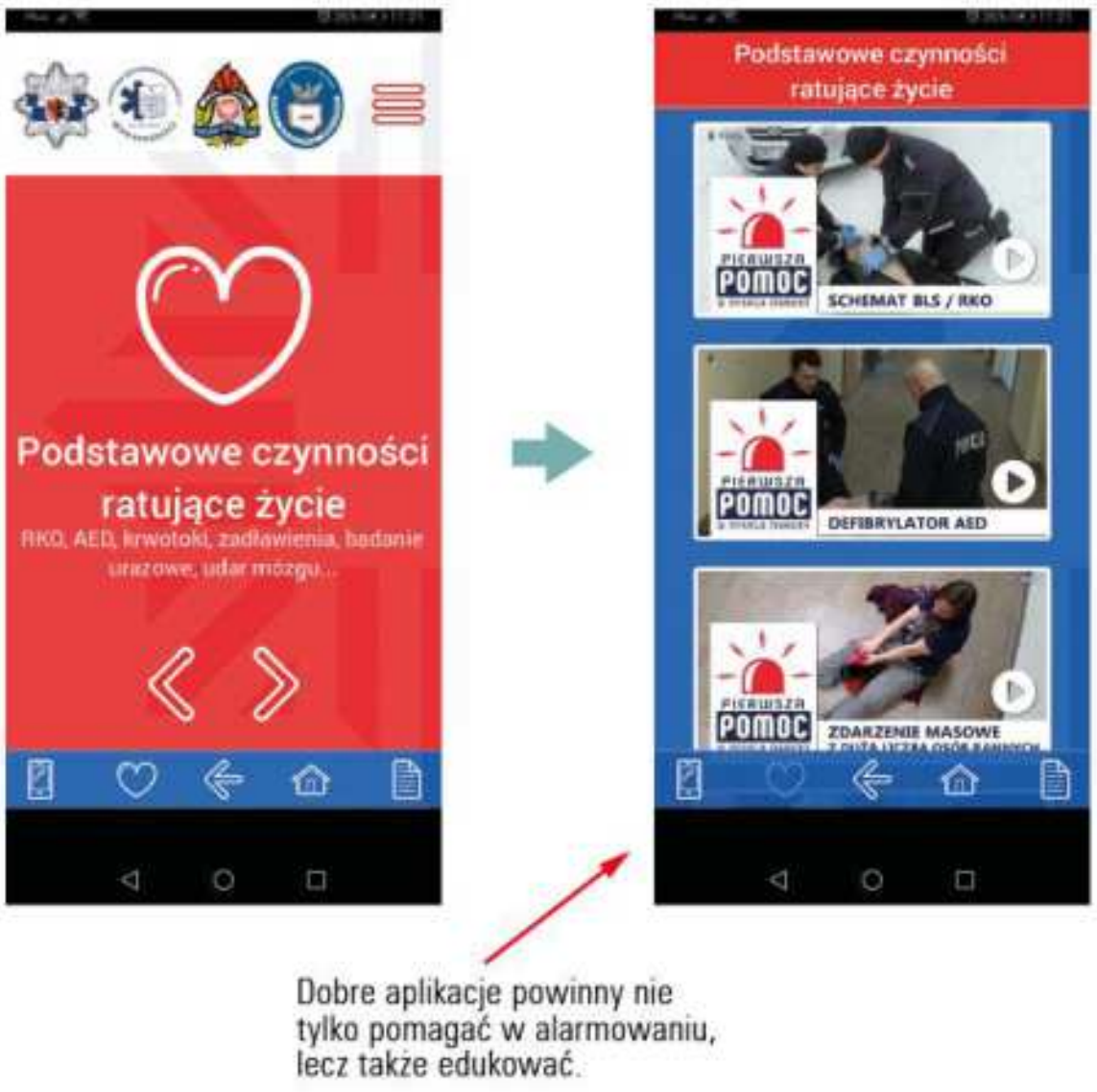
W sklepie z aplikacjami do smartfonów znajdziecie wiele ciekawych aplikacji dotyczących pierwszej pomocy. Jedną z nich opracowali policjanci z KWP w Bydgoszczy, ratownicy i strażacy, a więc osoby najbardziej kompetentne (rys. 10.1.).



Rys. 10.1. Aplikacja Pierwsza pomoc w sytuacjach zagrożeń opracowana przez służby ratownicze i policję

Są w niej trzy zasadnicze grupy zagrożeń i porady, jak się zachować, gdzie szukać pomocy, na jaki numer telefonować: wypadki drogowe, pożary i zamachy terrorystyczne. W sumie jednak użytkownik aplikacji może zapoznać się z 55 zagadnieniami wpływającymi na nasze bezpieczeństwo. Dowiemy się na przykład, jak udzielać pierwszej pomocy przedmedycznej, jak zachować się nad wodą i na lodzie. Twórcy aplikacji zadbali też o porady, co robić (rys. 10.2.), gdy dojdzie do katastrofy z dużą liczbą ofiar, jak wygląda segregacja rannych – mówi Szymon Tworowski, jeden z twórców aplikacji (źródło: pomorska.pl).

Podobne funkcje powinny pełnić także inne aplikacje, które mogą dodatkowo być wyposażone w obsługę GPS, automatycznego nawiązywania połączenia w przypadku wypadku drogowego itp.



Rys. 10.2. Filmy edukacyjne w aplikacji Pierwsza pomoc w sytuacjach zagrożeń

Zainstalowanie aplikacji pierwszej pomocy i korzystanie z niej w przypadkach nagłych wydarzeń – wypadków, zasłabnięć itp. – może bardzo ułatwić ratowanie życia innym. Gdy znajdziesz się w okolicy takiego zdarzenia, a nie wiesz, co robić, skorzystaj z aplikacji. Znajdź odpowiedni rozdział i postępuj według zaleceń.

10.2. Dodatki do telefonów, czyli monitorujemy stan organizmu

Wielu z was korzysta z urządzeń współpracujących z telefonem. Mogą one komunikować się z odpowiednimi aplikacjami z systemu Android wielu telefonów. Wspomniana aplikacja może także pracować autonomicznie i między innymi liczyć kroki podczas spaceru. Jednak najlepsze efekty przynosi powiązanie jej poprzez Bluetooth z innymi urządzeniami, np. opaską mierzącą tętno i inne parametry życiowe, wagą elektroniczną itp.

Opaska, smartwatch (czyt. smartłocz) i inne urządzenia mierzące parametry życiowe są używane głównie przez osoby aktywne fizycznie. Jednak w połączeniu z odpowiednią aplikacją sprzęty te pomagają ratować życie i zdrowie osób niepełnosprawnych lub starszych, np. żyjących samotnie. Alarm uruchomiony przez takie urządzenie może być odebrany przez bliskich lub służby ratunkowe. Możliwe jest także natychmiastowe połączenie z dyżurującym lekarzem. Prosta aplikacja przypomni także o porze wzięcia lekarstw, terminie wizyty u lekarza itp.

Wybór odpowiedniego urządzenia zależy od jego przyszłego zastosowania. Należy jednak pamiętać, że gdy w grę wchodzi życie i zdrowie, trzeba wziąć pod uwagę kilka istotnych parametrów i zasad. Należą do nich:

- niezawodność potwierdzona przez użytkowników, np. na forach,
- znana i polecana marka,
- niezawodna współpraca z danym modelem smartfona,
- długi czas pracy po każdym ładowaniu,
- szybkie i łatwe ładowanie akumulatora,
- wygoda noszenia lub instalacji,
- łatwość obsługi.

10.3. Projektant może pomóc, czyli ułatwienia informatyczne dla niepełnosprawnych

Rozwój technologii informatycznych i elektronicznych sprawił, że urządzenia są coraz bardziej przyjazne ludziom. Spotykacie się z takimi rozwiązaniami na każdym kroku. Czasami są niezauważalne, ale trudno byłoby sobie wyobrazić codzienne czynności bez nich.

Wiele miast i instytucji tak buduje swoje strony internetowe, by były przystosowane dla osób niepełnosprawnych, np. niedowidzących. Urzędy i instytucje użyteczności publicznej są do tego zobowiązane. Stan realizacji tego obowiązku bada Najwyższa Izba Kontroli (rys. 10.3.).

NIK o dostosowaniu stron internetowych do potrzeb osób z niepełnosprawnościami

10 LUTEGO 2016 08:50

Od czerwca 2015 r. publiczne strony internetowe powinny być w pełni dostosowane do potrzeb niepełnosprawnych. Nie wszystkim instytucjom udało się wywiązać z tego zadania. Część witryn wymaga jeszcze wielu poprawek i sporych nakładów pracy. NIK przypomina, że obowiązek dostosowania dotyczy nie tylko urzędów administracji publicznej, ale wszystkich podmiotów realizujących zadania publiczne.

Rys. 10.3. Fragment strony nik.gov.pl o raporcie w sprawie stanu dostosowania stron urzędów do potrzeb osób niepełnosprawnych

Ułatwienia dla niepełnosprawnych mogą polegać na przystosowaniu do wyświetlania liter i zdjęć w powiększeniu, a także na zwiększeniu kontrastu (rys. 10.4.).

ułatwienia dla niepełnosprawnych



Rys. 10.4. Przykłady stron e-urzędów miast przystosowanych do potrzeb osób niepełnosprawnych

Dzięki takim prostym zabiegom osoby niedowidzące mogą skorzystać z usług e-urzędu, bankowości elektronicznej itp.

Smartfony także są przystosowane do potrzeb osób niewidomych, starszych, mających problem z korzystaniem z ekranu dotykowego itp. dzięki usłudze, z której korzystamy na co dzień, czyli dzięki rozpoznawaniu głosu. Praktycznie każdy smartfon ma możliwość wybierania numeru głosem, rozpoznawania wypowiedzianych przez użytkownika wyrazów i wpisywania ich w postaci wiadomości, rozpoznawania adresu dla GPS itp.

Programy komputerowe potrafią także czytać tekst na głos. Większość z nich to programy komercyjne. Występują w wersji on-line lub instalowanej lokalnie.

translatory ▸ Jednym z najciekawszych rozwiązań informatycznych dla urządzeń mobilnych jest **translator** (tłumacz). Aplikacja potrafi tłumaczyć słowa i zdania wypowiadane do mikrofonu na dowolny język. Pakiety językowe można pobrać z sieci i nie trzeba wtedy wtedy utrzymywać połączenia internetowego. Tłumaczenie pokazuje się na wyświetlaczu (rys. 10.5.), ale możemy także uruchomić tłumaczenie dźwiękowe. Syntezator odczyta je przez głośniki telefonu.



Ustaw odpowiednie języki. Język, w którym mówisz, powinien zostać rozpoznany automatycznie.

Kliknij ikonę mikrofonu i wypowiedz tekst. Zostanie rozpoznany i przetłumaczony.

Rys. 10.5. Ekran tłumacza z telefonu komórkowego

Dodatkowo można także tłumaczyć napisy, np. z tablic informacyjnych (rys. 10.6.).



Rys. 10.6. Przykład tłumaczenia sfotografowanych tekstów

Łatwa w użyciu aplikacja tłumacza zapobiegnie problemom z komunikacją w obcym kraju. Dzięki temu wzrośnie poczucie bezpieczeństwa, a zmniejszy się stopień wykluczenia.

10.4. Sztuczna inteligencja, czyli elektroniczny asystent

W wielu filmach *science fiction* (czyt. sajens fikszyn) można spotkać roboty, które pomagają ludziom w codziennych obowiązkach i dotrzymują im towarzystwa. Współcześnie takie urządzenia pozostają w fazie eksperymentów i dużo czasu upłynie, zanim ziści się wizja filmowców. Obecnie są dostępne jedynie niewielkie urządzenia zwane asystentami. Mają one wbudowane funkcje sztucznej inteligencji (rys. 10.11.). Potrafią prowadzić dialog, wykonać proste polecenia, np. zestawić połączenie telefoniczne, uruchomić odtwarzanie muzyki, zmienić jej głośność itp. Te urządzenia mogą być bardzo przydatne dla osób chorych, leżących lub niepełnosprawnych.



Rys. 10.7. Przykład elektronicznego asystenta ze sztuczną inteligencją

W podobne funkcje są wyposażone także piloty do współczesnych telewizorów. Rozpoznają mowę i potrafią wykonać proste polecenia związane ze swoim funkcjonowaniem.

Uwaga!

Wygląd i rozmieszczenie opcji aplikacji i stron może się zmieniać. Zwróć jednak uwagę na ich funkcjonalność i cechy zgodne z opisem w rozdziale.

ZADANIA DO WYKONANIA

1. Znajdź stronę internetową urzędu swojego miasta lub gminy. Sprawdź, czy e-urząd jest przystosowany do potrzeb osób niedowidzących. Jeśli tak, określ, na czym polega to przystosowanie. Przedstaw swoje pomysły na unowocześnienie i dostosowanie strony dla niepełnosprawnych.
2. Zainstaluj w swoim telefonie bezpłatną aplikację **Pierwsza pomoc w sytuacjach zagrożeń** opisaną w rozdziale. Przejrzyj uważnie wszystkie zasoby i opisz w edytorze tekstu przykładowe sytuacje, w których będzie ona przydatna.
3. Znajdź w internecie aplikacje przyczyniające się do zwiększenia bezpieczeństwa. Oceń ich przydatność. Wybierz najciekawsze oraz opisz (korzystając z edytora tekstu) ich zalety i wady
4. Użyj tłumacza swojego telefonu do tłumaczenia zapisanego w egzotycznym języku (np. chińskim) menu twojego monitora lub telewizora. Użyj opcji rozpoznawania tekstu ze zdjęcia. Nie zapomnij przywrócić poprzedniego języka menu urządzenia.
5. Wypróbuj działanie wszystkich opcji tłumacza twojego telefonu, łącznie z tłumaczeniem głosowym. Oceń jakość tłumaczenia na podstawie słownika lub własnej znajomości języka obcego.

PODSUMOWANIE LEKCJI

s. 62 • **Aplikacje pierwszej pomocy** mogą uratować życie. Warto się z nimi zapoznać i w razie potrzeby z nich skorzystać.

s. 65 • Osoby do niedawna mające poczucie wykluczenia ze względu na swoją niepełnosprawność mogą korzystać z ułatwień, jakie dają programy i urządzenia komputerowe.

s. 66 • **Translatory** tłumaczą słowa pisane, mówione oraz sfotografowane, co znacznie ułatwia komunikację z ludźmi mówiącymi w obcym języku.

11. Platformy uczą, czyli rola e-learningu w nauce i pracy

NA TEJ LEKCJI:

- dowiesz się, czym jest e-learning i jakie są jego cechy;
- poznasz przykłady portali kształcących na odległość i wspomagających nauczanie w szkole;
- porównasz cechy e-learningu z kształceniem tradycyjnym.

Uczysz się już wiele lat. Wiesz, na czym polega ten proces. Przychodzisz do szkoły na określoną liczbę lekcji według z góry ustalonego planu. Siedzisz w ławce z kolegami i koleżankami i bierzesz udział w zajęciach. Taki sposób nauczania ma wiele zalet i jest skuteczny. Czy można uczyć się w inny sposób? Czy tradycyjna szkoła może być wspomagana przez nauczanie zdalne? Czy e-learning może ją zastąpić? Czym on jest? Jak z niego korzystać? Kiedy jest najskuteczniejszy? Zanim odpowiesz na te pytania, zapoznaj się z cechami nauczania zdalnego.

11.1. Tradycyjnie czy przez sieć, czyli czym różni się nauczanie tradycyjne od e-learningu

Zanim rozpowszechnił się internet, zdalne nauczanie odbywało się za pośrednictwem radia, telewizji i poczty. W Australii dzieci mające bardzo daleko do szkoły uczestniczyły w lekcjach za pośrednictwem radiostacji. Do niedawna działały także instytucje oferujące kursy korespondencyjne, np. języków obcych z wykorzystaniem nagrań audio i wideo. Wszystkie te formy miały wspólną cechę: uczeń uczył się samodzielnie z dostarczonych przez organizatora materiałów, a następnie zdawał odpowiednie egzaminy. Ta zasada jest zachowana także we współczesnej formie nauczania zdalnego – **e-learningu**. Jednak możliwości oraz szybkość przekazywania materiałów do nauki i rozwiązań zadań kontrolnych i testów są nieporównywalne.

▸ e-learning

Czy wiesz, jak tworzy się lekcje e-learningowe?

Cykl nauczania e-learningowego
Zespół specjalistów przygotowuje program nauczania oraz scenariusze lekcji e-learningowych.
Nauczyciele przygotowują, materiały edukacyjne, np. filmy, teksty, zdjęcia, rysunki, pliki audio, prezentacje, odnośniki do wiarygodnych informacji itp. Wskazują rozdziały w podręcznikach. Dla wszystkich grup uczestników są one identyczne i można je wykorzystać także w innych kursach.
Nauczyciele przygotowują testy i sprawdziany kształcąco-sprawdzające zamykające omawiane zagadnienie, dział lub całe szkolenie.
Nauczyciele lub administrator umieszczają wszystkie materiały zgodnie z programem i scenariuszami szkolenia na platformie zdalnego nauczania.

Cykl nauczania e-learningowego
Uczestnicy wykonują określone przez nauczyciela czynności i uczą się, wykorzystując umieszczone na platformie materiały
Po każdej lekcji rozwiązują do skutku testy i zadania sprawdzająco-uczące, by sprawdzić poziom opanowania materiału.
Uczniowie po zakończeniu cyklu uczestniczą w egzaminie, który także może odbyć się za pośrednictwem platformy e-learningowej w domu ucznia lub w ośrodku egzaminacyjnym.

Rys. 11.1. Przykładowy cykl przygotowania i przebiegu szkolenia e-learningowego

Różnice pomiędzy tradycyjną formą nauczania a e-learningiem są widoczne nie tylko w sposobie przygotowywania i przebiegu lekcji. Zestawmy je w tabeli (tab. 11.1.). Przeanalizuj ją uważnie. Które z cech uznasz za wady, a które za zalety e-learningu?

e-learning	Nauczanie tradycyjne
Czas jest optymalnie wykorzystany zgodnie z tempem pracy uczestnika.	Nauka odbywa się na 45-minutowych lekcjach.
Uczestnik może się uczyć z przerwami i dzielić lekcję na części.	
Uczestnik samodzielnie wybiera porę przystąpienia do kursu mieszczącą się w wyznaczonym przez nauczyciela terminie.	
Uczestnik ma możliwość zmiany kolejności realizacji zadań i tematów.	Temat i zadania są narzucone przez prowadzącego zajęcia.
Testy sprawdzające poziom opanowania tematu pozostają bez oceny wpisywanej do dziennika.	Sprawdziany najczęściej są oceniane.
Uczestnik korzysta z kompletu materiałów dostarczonych przez nauczyciela oraz dostępnych w sieci.	W czasie lekcji uczeń korzysta najczęściej z podręcznika i zeszytu ćwiczeń oraz wykonuje notatki.
W procesie uczenia uczestnik wykonuje szereg ćwiczeń i zadań będących integralną częścią lekcji.	Ćwiczenia wykonuje się z nauczycielem lub w ramach zadań domowych.
Uczestnik ma ograniczony kontakt z nauczycielem.	Jest bezpośredni kontakt z nauczycielem.
Nie ma możliwości dostosowania poziomu trudności do indywidualnych potrzeb ucznia.	Nauczyciel może różnicować poziom trudności omawianego tematu w zależności od zdolności uczniów, np. wykraczając poza program nauczania.
Nie ma bezpośredniego kontaktu z kolegami z klasy. Jest on możliwy jedynie za pośrednictwem czatu lub komunikatora.	Istnieje możliwość wspólnego uczenia się i zawiązywania przyjaźni.
Wymagana jest konieczność narzucenia sobie dużej dyscypliny i samokontroli.	Plan lekcji wymusza uczestnictwo w lekcjach.
Zaliczenie następuje przez internet lub w ośrodku egzaminacyjnym (dla kursów kwalifikacyjnych, np. nadających uprawnienia).	Poszczególne partie materiału zalicza się na pracach klasowych.
W trakcie kursu e-learningowego uczestnik nie jest zaskakiwany niezapowiedzianymi formami sprawdzania wiedzy i umiejętności.	W czasie lekcji można być odpytanym lub pisać niezapowiedziany sprawdzian.

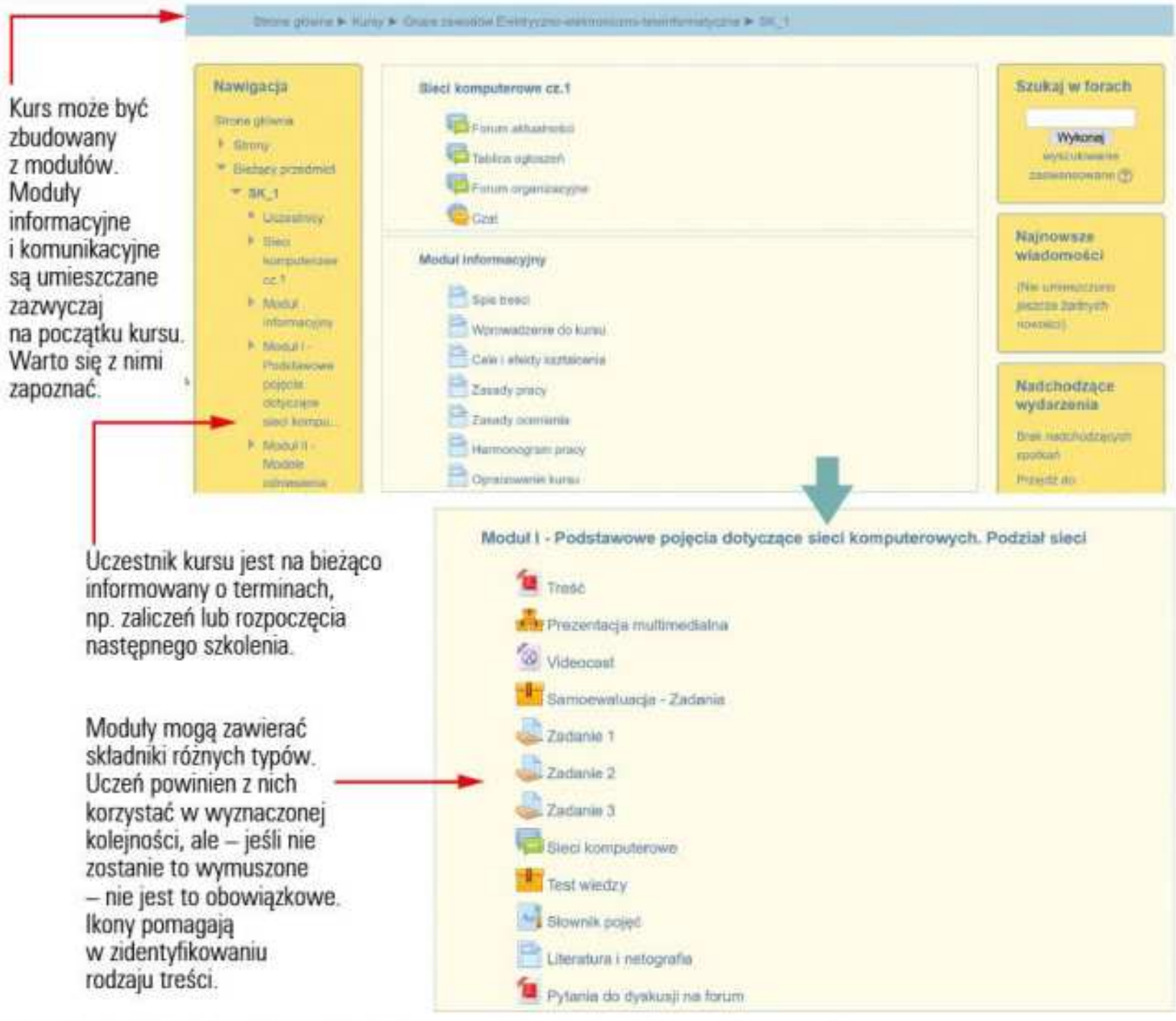
Tab. 11.1. Różnice pomiędzy tradycyjną lekcją a e-learningową

Trudno jest oceniać formy nauczania bez próby uczestniczenia w nich. Szkolne realia znasz. Sprawdź, jak można się uczyć w sieci. Korzystając z przeglądarki, znajdziesz wiele darmowych kursów, w których możesz wziąć udział. Na stronie Ośrodka Rozwoju Edukacji (ORE) pod adresem *https://kno.ore.edu.pl/repozytorium-kursow.html* znajdziesz komplet lekcji przygotowujących do egzaminów zawodowych.

Lekcje ORE zostały przygotowane za pomocą jednej z najpopularniejszych platform do zdalnego nauczania – **Moodle** (czyt. mudle). To darmowe środowisko zarządzania treściami edukacyjnymi jest dostępne na zasadzie licencji GNU i od wielu lat nieustannie doskonalone. Osoby zajmujące się intensywnym jego wykorzystywaniem i tworzeniem nowych wersji nazywa się modlerami. Do instalacji jest potrzebne konto na serwerze z bazą danych. Większość polskich uczelni korzysta z tej platformy.

▸ platforma Moodle

Zapisz się, używając konta **Gość**, na interesujące cię szkolenie, np. z grupy zawodów elektryczno-elektroniczno-teleinformatycznej – *Sieci komputerowe*, cz. 1. (rys. 11.2.).



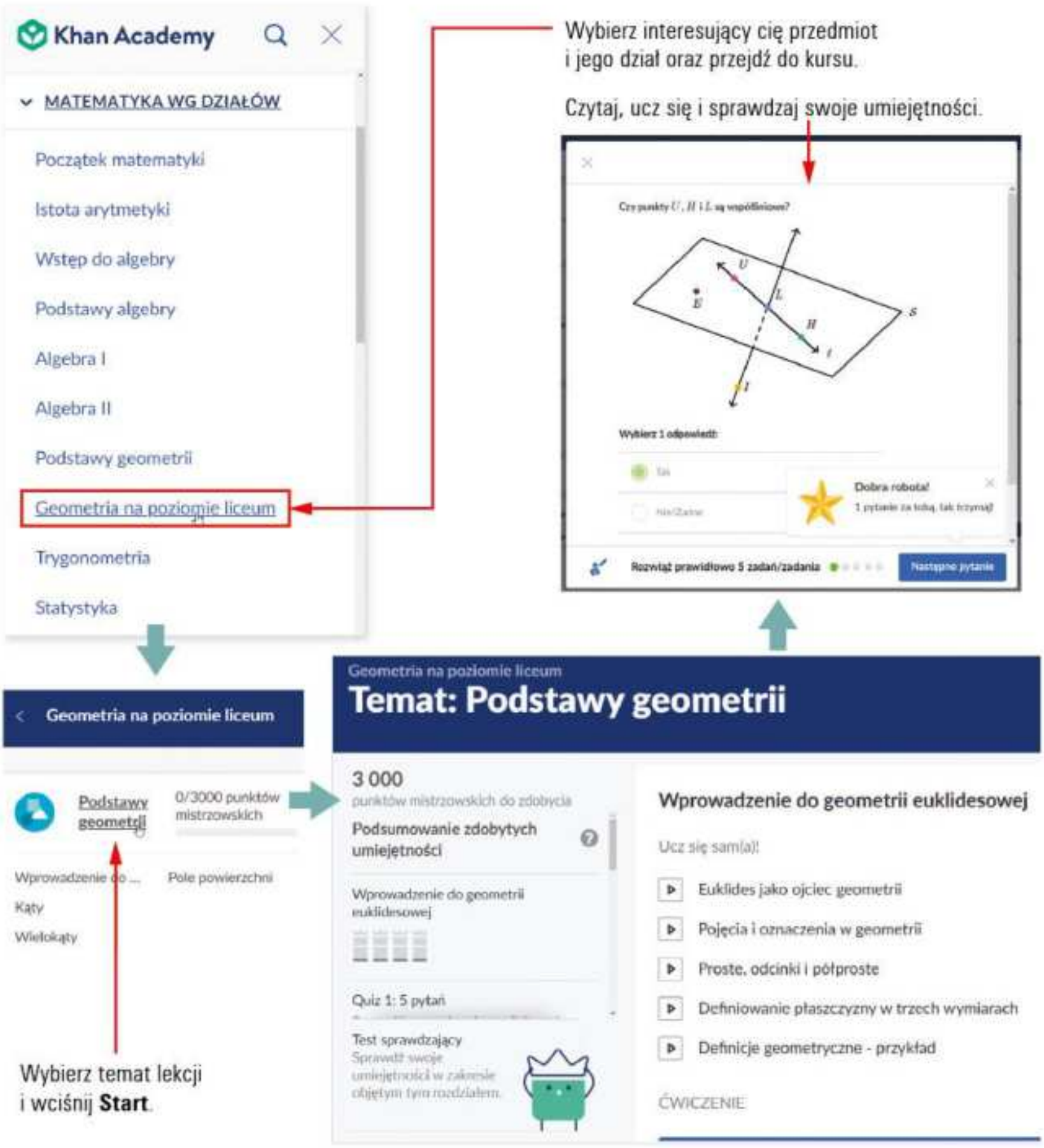
Rys. 11.2. Przykładowy kurs e-learningowy ze strony Ośrodka Rozwoju Edukacji

Niektóre elementy szkolenia ORE zostały przygotowane w uniwersalnym formacie SCORM. Wiele platform e-learningowych ma możliwość importowania

▸ format SCORM

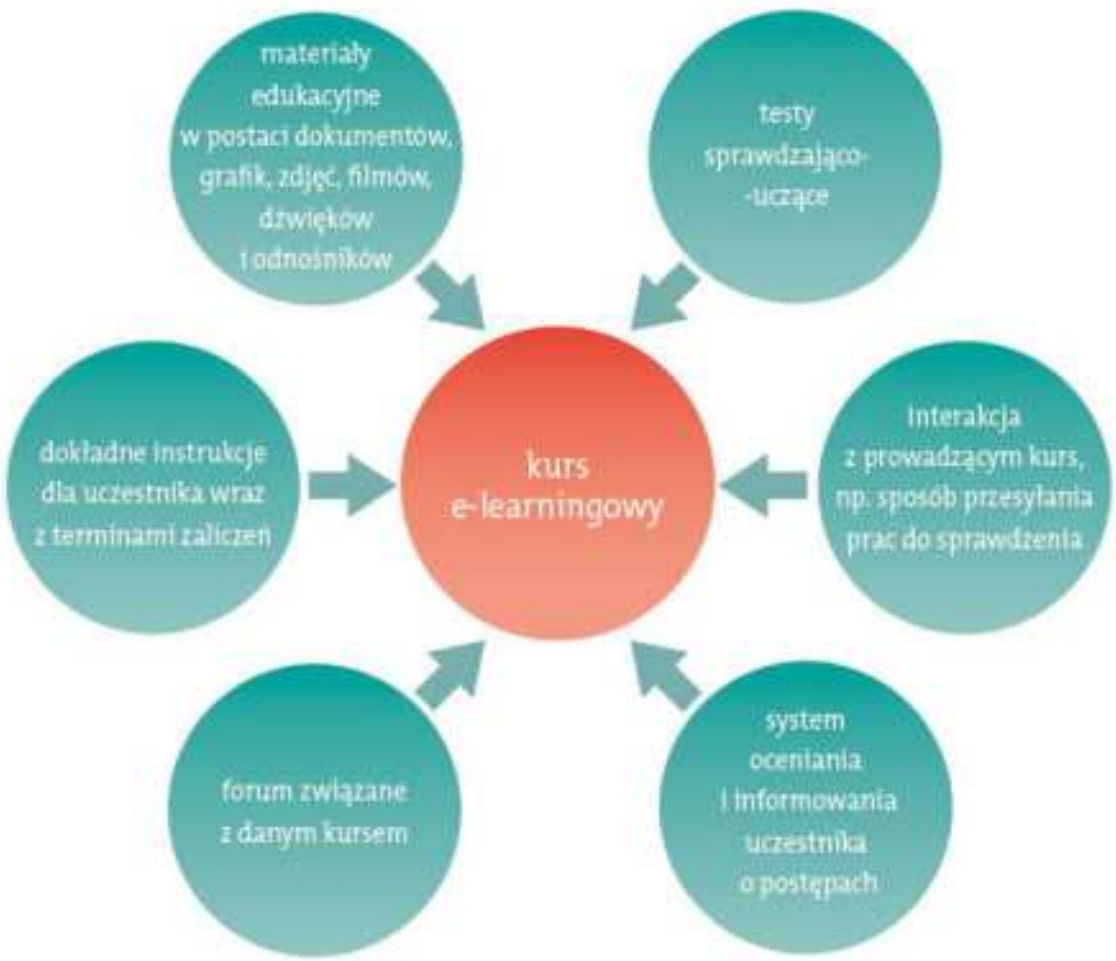
takich treści. Autorzy mogą więc przygotowywać zasoby edukacyjne akceptowalne przez wielu organizatorów szkoleń elektronicznych.

W sieci znajdziesz także wiele elektronicznych lekcji, bardzo przydatnych w szkole. Na przykład na portalu *pl.khanacademy.org* umieszczono kursy dla wszystkich poziomów kształcenia, w tym również dla uczniów liceum i technikum (rys. 11.3.). Jeśli masz zaległości z któregoś z przedmiotów, pomoc znajdziesz w Khan Academy (czyt. kan akademi). Możesz tam założyć własne konto lub skorzystać ze swojego konta Google lub Facebook.



Rys. 11.3. Przykład lekcji e-learningowej z portalu edukacyjnego Khan Academy

Po zapoznaniu się z powyższymi kursami e-learningowymi i korzystając z własnych doświadczeń z innymi portalami edukacyjnymi, możesz już ocenić trafność informacji z tabeli 1. Wiesz już także, jak mogą być zbudowane kursy e-learningowe (rys. 11.4.).



Rys. 11.4. Przykładowa struktura lekcji e-learningowej

zastosowanie platformy

Dobrym uzupełnieniem platformy jest uruchomienie konsultacji z nauczycielem za pośrednictwem komunikatora wideo. W takim przypadku prowadzący ustala w kalendarzu kursu dzień i godzinę dyżuru oraz sposób połączenia i odnalezienia go w systemie komunikatora. Jest to szczególnie przydatne w przypadku nauczania języków obcych.

Dzięki ciągłemu rozwojowi systemów zdalnego nauczania istnieje **nieskończenie wiele możliwości ich zastosowań**. Wszystko zależy od inwencji i pomysłowości twórców kursów. Platforma jest tylko narzędziem w rękach nauczyciela, tak jak kreda i tablica. Gdy podejmiesz pracę zawodową, dzięki możliwości kształcenia na odległość możesz ustawicznie podnosić swoje kwalifikacje. Wiele firm finansuje swoim pracownikom takie szkolenia.

Prowadzenie komercyjnych kursów e-learningowych to także dobry plan twojego przyszłego rozwoju zawodowego.

Uwaga!

Opisane portale i lekcje mogą w przyszłości przybrać inny kształt. Nie powinny jednak zmienić swojej funkcjonalności.

ZADANIA DO WYKONANIA

1. Przejrzyj zawartość kursu udostępnionego przez ORE i w prezentacji PPTX przedstaw najważniejsze założenia i możliwości platformy Moodle. Uwzględnij umieszczanie materiałów rozmaitych formatów, quizy (testy), ocenianie itp. Prezentację zaprojektuj w taki sposób, by mogła stać się wstępem do kursu o Moodle.
2. Znajdź w sieci oficjalną stronę Moodle i zapoznaj się z aktualnościami na jej temat. Niektóre informacje będą podane w języku angielskim. Możesz się posłużyć tłumaczem Google. Opisz krótko w zeszycie strukturę tej strony i wskaż elementy potwierdzające, że została ona wykonana za pomocą Moodle.
3. Przejrzyj ofertę Khan Academy i wynotuj w zeszycie adresy kursów dotyczących najbliższych lekcji z dowolnego przedmiotu. Wybierz jeden, zapoznaj się z treściami oraz wykonaj testy i ćwiczenia. Napisz w zeszycie krótką notatkę na temat tego, jak lekcja e-learningowa pomogła ci w zrozumieniu nowego tematu na lekcji w klasie.
4. Ułóż scenariusz własnej lekcji e-learningowej na temat platform zdalnego nauczania (e-learningowych) i ich roli w doskonaleniu zawodowym.
5. Wyszukaj w sieci strony szkół ponadpodstawowych korzystających z platform zdalnego nauczania. Sprawdź, jakie przedmioty ujęto na tych platformach. Czy w twojej szkole korzysta się z tej formy wspomagania nauczania?

PODSUMOWANIE LEKCJI

s. 69 • Zdalne nauczanie za pośrednictwem internetu nazywa się **e-learningiem**.

s. 71 • Popularna platforma **Moodle** jest darmowa i można ją instalować na serwerach z aktywnym systemem baz danych.

s. 71 • **SCORM** jest uniwersalnym formatem zapisu treści edukacyjnych akceptowalnym przez wiele platform e-learningowych.

s. 73 • Istnieje wiele programów służących kształceniu na odległość, w tym komunikatory i programy multimedialne instalowane lokalnie.

s. 73 • Zastosowanie platformy zależy tylko od inwencji i pomysłowości autora kursu.

12. Możesz być administratorem, czyli jak zarządzać platformą e-learningową

NA TEJ LEKCJI:

- dowiesz się, jak wykorzystać narzędzia chmury informatycznej do prowadzenia szkoleń elektronicznych;
- zbudujesz strukturę własnego kursu e-learningowego.

Wiesz już, jakie korzyści możesz wynieść z kursów e-learningowych. Poznałeś/Poznałaś ich cechy. Zdajesz sobie także sprawę, że nie wszystkie są dobrze dopasowane do twoich potrzeb. Spróbuj samodzielnie utworzyć strukturę takiego kursu.

12.1. Wybór jest ważny, czyli jaką platformę wybrać

wybór platformy

Znasz już jedną z darmowych platform e-learningowych – Moodle. To bardzo rozbudowane narzędzie pozwalające nie tylko na budowanie rozmaitych szkoleń, lecz także na zarządzanie szkoleniami, w tym – ich sprzedaż. Wymaga samodzielnego zainstalowania na serwerze z bazą danych, a to wiąże się z wykorzystaniem usługi hostingowej (wynajęcia miejsca na serwerze i podpięcia go pod określoną domenę). Można także skorzystać z komercyjnych usług udostępniających administrowanie gotową platformą. Początkujący edukator internetowy może mieć z tym problem. Czy są inne rozwiązania?

Posiadacze konta w Google korzystają najczęściej z najpopularniejszych usług – poczty, YouTube, wyszukiwarki, a rzadziej z dysku i programów z chmury (np. edytora tekstu, prezentacji itp.). Mało kto jednak wie, że w tej chmurze można znaleźć aplikację **Classroom** (czyt. klasrum) – narzędzie do tworzenia lekcji e-learningowych i wspomagania pracy nauczyciela lub prowadzącego kursy. Warto się z nim zapoznać i wykorzystać, np. do przygotowania szkolnego konkursu. Oprócz zadań będą w nim zamieszczone treści do przygotowania i odnośniki do sprawdzonych źródeł. Testy pozwolą z kolei na przeprowadzenie części konkursowej i wyłonienie zwycięzców.

12.2. Od czego zacząć, czyli planujemy przygotowanie kursu e-learningowego

Pracę z dowolną platformą zawsze zaczyna się od założenia konta. Uczestnicy twojego konkursu będą więc musieli takie konto utworzyć. Następnym krokiem jest zaplanowanie struktury kursu oraz zakresu wiedzy i umiejętności, jakie chcesz przekazać swoim „uczniom". To jeden z najtrudniejszych etapów tworzenia lekcji e-learningowej. Od niej zależy, czy twoje szkolenie będzie skuteczne i atrakcyjne.

Aby dobrze zaplanować kurs, musisz odpowiedzieć na kilka pytań. Swoje przemyślenia najlepiej zanotować i później korzystać z nich w trakcie tworzenia e-szkolenia.

Załóżmy, że twój kurs będzie eliminacją do szkolnego konkursu matematycznego. Ta informacja daje już odpowiedzi na wiele istotnych pytań (tab. 12.1.).

Lp.	Pytanie	Przykładowa odpowiedź
1.	Jaki zakres wiedzy i umiejętności chcesz przekazać w kursie?	Zakres przewidziany w podstawie programowej matematyki w zakresie rozszerzonym.
2.	Dla kogo przeznaczony jest kurs?	Dla uczniów mojej szkoły.
3.	W jakim wieku będzie większość uczestników?	W wieku szkolnym.
4.	Jaki jest poziom aktualnej wiedzy i umiejętności potencjalnych użytkowników?	Poziom rozszerzony z matematyki dla szkoły ponadpodstawowej.
5.	Jaki jest przewidywany czas opanowania wiedzy i umiejętności przez potencjalnych użytkowników?	Odpowiedź będzie znana po zaplanowaniu treści, jakie zostaną umieszczone w kursie.
6.	Której platformy będziesz używać?	W ćwiczeniu zostanie użyta platforma Classroom z Google.
7.	Jak zapewnisz dostęp do kursu potencjalnym uczestnikom?	Dostęp zyskają użytkownicy kont Google zaproszeni przez organizatora na podstawie adresu e-mail.

Tab. 12.1. Przykładowe odpowiedzi na pytania postawione przed projektowaniem kursu e-learningowego

12.3. Co mogę zaplanować, czyli poznajemy narzędzia platformy

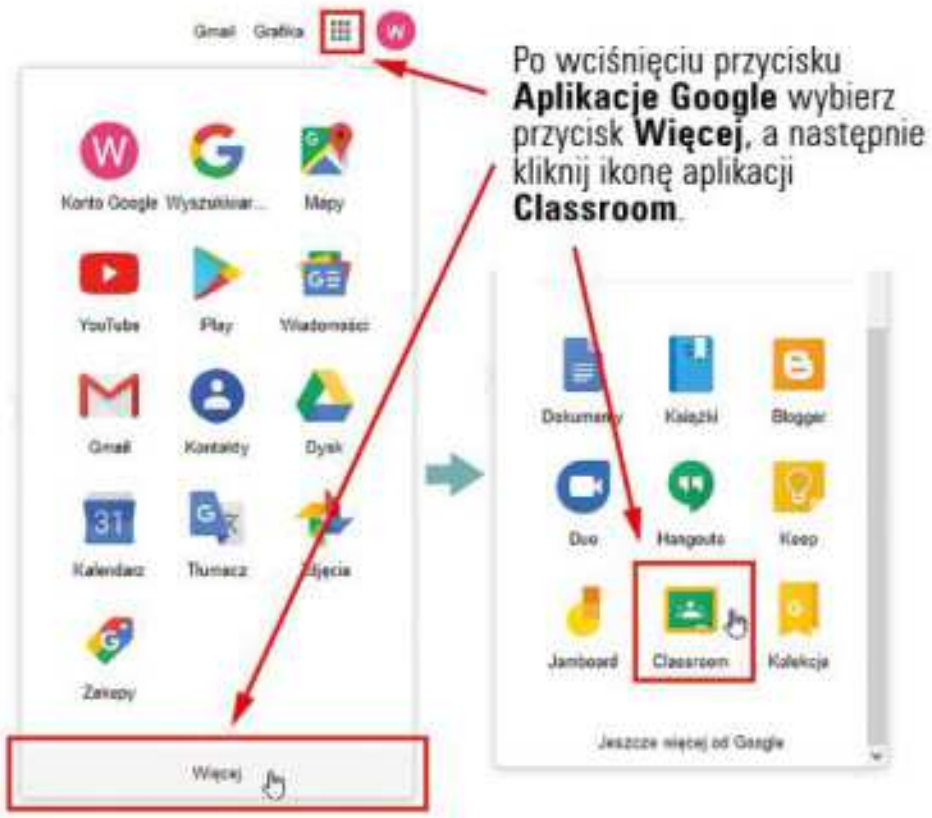
Skoro przypadła ci w udziale odpowiedzialność za przygotowanie platformy do umieszczenia w niej treści konkursu i zorganizowania części konkursowej, zaplanuj strukturę kursu (rys. 12.1.).



Rys. 12.1. Przykładowy projekt struktury kursu dla szkolnego konkursu matematycznego

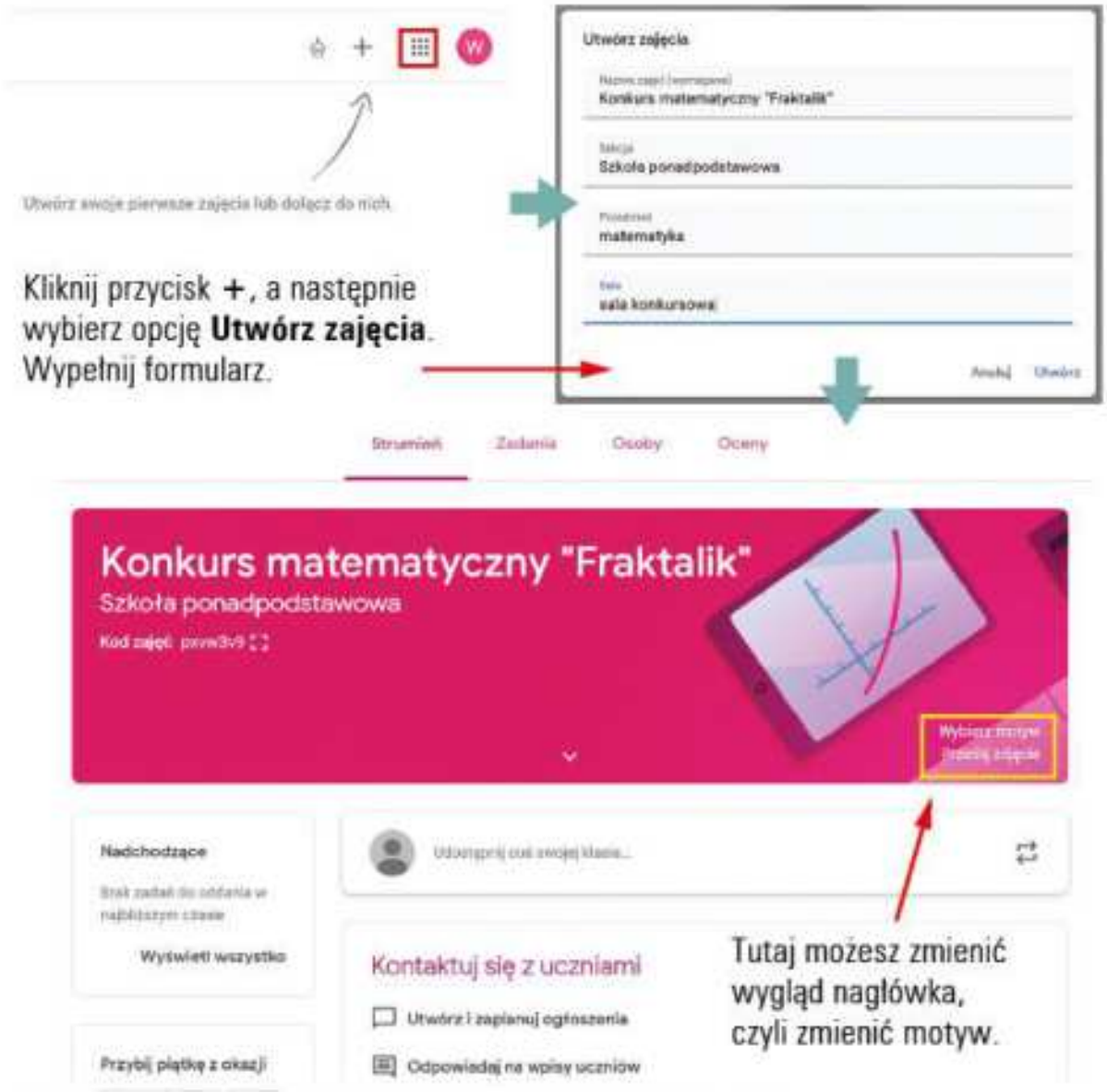
Skoro projekt struktury kursu jest już gotowy, czas go odpowiednio skonfigurować w platformie.

Google Classroom ▸ Załóżmy, że zalogowałeś/zalogowałaś się do konta Google. Następnym krokiem będzie znalezienie i uruchomienie aplikacji Classroom (rys. 12.2.).



Rys. 12.2. Uruchomienie aplikacji **Classroom** w chmurze Google

Aplikację obsługuje się intuicyjnie. Zorganizujmy więc zaplanowaną wcześniej strukturę kursu. Ponieważ w aplikacji Classroom największą strukturą kursu jest konstrukcja **Zajęcia**, całość przygotowanej struktury zamieścimy właśnie tam (rys. 12.3.).



Rys. 12.3. Pierwszy krok w tworzeniu lekcji (**Zajęć**) w **Classroom**

W konstrukcji **Zajęcia** należy teraz umieścić poszczególne etapy naszego konkursu, czyli zasoby lekcji (rys. 12.4.). Czynności z rysunku wykonaj jeszcze dla części: Etap 1., Etap 2. i Finał.

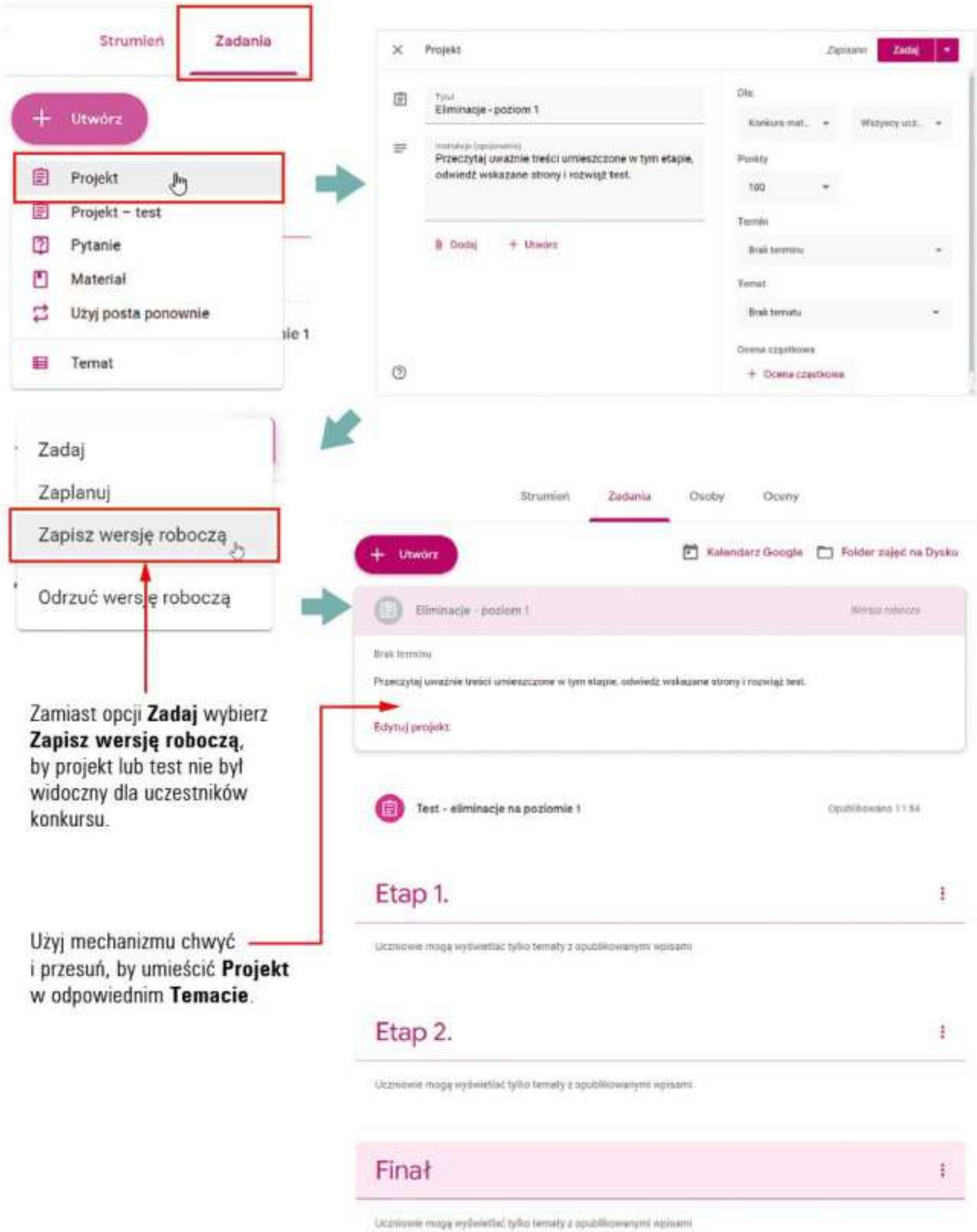


Rys. 12.4. Dodawanie tematu do **Zajęć** w aplikacji **Classroom**

Ponieważ pierwszy temat będzie zawierał materiały do nauki i zasady konkursu, umieść w nim projekt (rys. 12.5.). Następne tematy pozostaw puste. Zostaną one wypełnione treścią po opracowaniu całości materiału i zadań konkursu.

Uwaga!

Widoczne są wyłącznie tematy zawierające treści.



Zamiast opcji **Zadaj** wybierz **Zapisz wersję roboczą**, by projekt lub test nie był widoczny dla uczestników konkursu.

Użyj mechanizmu chwyć i przesuń, by umieścić **Projekt** w odpowiednim **Temacie**.

Rys. 12.5. Tworzenie **Projektu** w **Temacie** w aplikacji Classroom

Jeśli uważnie wykonałeś/wykonałaś wszystkie czynności, struktura twojego kursu – konkursu powinna być gotowa do wypełniania jej treścią. Pamiętaj, aby nie udostępniać zasobów kursu, jeśli nie są one jeszcze gotowe. Uczestnik powinien je zobaczyć dopiero w wyznaczonym terminie.

ZADANIA DO WYKONANIA

1. Uruchom aplikację Classroom. Utwórz w niej **Zajęcia** pod nazwą *Informatyka w mojej klasie*. Wewnątrz zaprojektuj test sprawdzający wiedzę z ostatnich trzech lekcji informatyki.
2. Zaprojektuj strukturę kursu (Zajęć) dla twojego ulubionego przedmiotu szkolnego. Uwzględnij w nim tematy, które już zostały omówione na zajęciach. Przygotuj strukturę do wypełniania treścią.
3. Opracuj prezentację opisującą poszczególne etapy tworzenia kursu (Zajęć) w aplikacji Classroom. Udostępnij ją kolegom i koleżankom. Poproś ich o ocenę merytoryczną twojej pracy.

PODSUMOWANIE LEKCJI

s. 75
- Dobrze zaplanowany kurs pozwoli na zbudowanie odpowiedniej struktury w platformie. **Wybór platformy** e-learningowej do tworzenia własnych kursów zależy od wymagań, jakie stawia kurs.

s. 76
- Kursy lub lekcje o nieskomplikowanej strukturze możesz wykonać i udostępnić w **aplikacji Classroom** w chmurze Google.

13. Zasoby i testy, czyli wypełniamy kursy treścią

NA TEJ LEKCJI:

- dowiesz się, jak projektować zasoby kursów e-learningowych;
- dodasz treści i testy do platformy;
- zaprosisz użytkowników do korzystania z twojego kursu.

W poprzednim rozdziale przygotowaliśmy strukturę konkursu matematycznego jako Temat w aplikacji Classroom w chmurze Google. Czas na wypełnienie jej treścią do nauki i pytaniami konkursowymi. Zanim jednak do tego przystąpisz, zapoznaj się z niektórymi zasadami tworzenia zasobów e-learningowych.

13.1. Co mogę umieścić w kursie, czyli rodzaje zasobów e-learningowych

Zasoby platform e-learningowych można podzielić na grupy, w których ważne są parametry techniczne (np. czy jest to tekst, czy dźwięk) oraz funkcjonalne (np. czy są informacjami, testami itp.).

Podział zasobów z punktu widzenia parametrów technicznych:

▸ rodzaje zasobów

1. **Dokumenty tekstowe** są najczęściej wybierane ze względu na swoje zalety. Po pierwsze, przy zastosowaniu odpowiedniego formatu są łatwe w odczycie bez względu na system operacyjny, którego używamy. Po drugie, można je wydrukować i korzystać z nich poza komputerem lub smartfonem. Nie wytwarzają dźwięków. Mogą za to zawierać aktywne odnośniki do treści zamieszczonych w sieci. Najczęściej stosowanym formatem jest PDF.

2. **Filmy i animacje** są atrakcyjnym zasobem. Wymagają jednak dużego zaangażowania autora i odpowiedniego sprzętu. W dobie smartfonów nie ma problemu z nagraniem obrazu, jednak pewne trudności może sprawić uzyskanie zadowalającej jakości dźwięku. Filmy można umieścić w pliku, a użytkownik musi je pobrać i odtworzyć za pomocą odpowiedniego programu. Często stosowanym formatem jest MPEG4. Alternatywnym sposobem jest transmisja strumieniowa na życzenie, czyli umieszczenie filmu na przykład na platformie YouTube. Wówczas użytkownik ogląda film bezpośrednio w oknie kursu. Zasób nie jest pobierany w postaci pliku.

3. **Zasoby graficzne** to dostępne w kursie zdjęcia lub rysunki. Mogą występować jako pliki do pobrania, na których uczestnik kursu może ćwiczyć lub wykonywać zadanie, albo jako ilustracje dokumentów tekstowych. Często ilustracje znajdują się także w pytaniach testowych.

4. **Zasoby multimedialne** powinny być tak przygotowane, by uczestnik szkolenia nie musiał instalować dodatkowego oprogramowania. Wiele mediów tworzy zasób multimedialny. W prezentacjach możliwe jest stosowanie kombinacji wielu różnych zasobów: grafiki, dźwięku, animacji, filmu, tekstu.

5. **Zasoby dźwiękowe** występują często w kursach, w których są po prostu niezbędne, np. w zasobach do nauki języka, muzyki itp. Najczęściej są zapisywane i udostępniane w formacie mp3. Można je także, podobnie jak zasoby filmowe i animacje, umieszczać na serwerach typu YouTube. Oczywiście ich przygotowanie wymaga zadbania o dobrej jakości dźwięk, a więc i dobry sprzęt do rejestracji.

Podział zasobów z punktu widzenia funkcjonowania w kursie:

1. **Zasoby informacyjne**, które są nośnikiem wiedzy podanej bezpośrednio, takiej jak: definicje, charakterystyki, opisy.

2. **Zasoby aktywujące**, które powinny zmuszać uczestnika kursu do aktywnego przyswajania umiejętności i wiedzy. Są to najczęściej rozmaite ćwiczenia, zadania, zagadki, łamigłówki itp.

3. **Zasoby sprawdzania wiedzy**, które mogą służyć ocenie i zaliczeniu kursu, a także samoocenie aktualnej wiedzy przez samego uczestnika. Najczęściej występują dwa rodzaje zadań testowych: krótkiej odpowiedzi i jednokrotnego wyboru.

4. Aktywowane przez autora **mechanizmy systematyzujące** pracę kursantów, np. kalendarze, sygnalizacja zbliżającego się terminu itp.

tworzenie zasobów ▸ Wiesz już, jak tworzyć niektóre z wymienionych rodzajów zasobów. Dobrze redagujesz teksty i eksportujesz je do różnych formatów. Umiesz wykonywać rysunki i przygotowywać zdjęcia do publikacji. Z nagraniem dźwięku też nie będzie problemu. Tworzysz również prezentacje, w których łączysz wszystkie media. Zajmijmy się więc przygotowaniem **zasobów** służących sprawdzaniu wiedzy. W większości platform występują wszystkie podstawowe rodzaje pytań i zadań.

Najpopularniejszy jest oczywiście **test wyboru** (rys. 13.1.) jednokrotnego lub wielokrotnego. W drugim przypadku można ustalić ważność danej odpowiedzi, czyli niektóre z nich mogą być wyżej punktowane od innych.

Prawda czy fałsz to kolejny typ pytania w teście. Jak widać, istnieją tylko dwie możliwości odpowiedzi.



Połącz elementy to rodzaj zadania sprawdzającego między innymi skojarzenia uczestnika. Polega na łączeniu pasujących do siebie danych, np. nazwy jednostki miary i jej symbolu.

Uzupełnij to typ polecenia, w którym uczestnik musi wpisać określony element, np. brakujący wyraz w zdaniu. Szczególnie przydatne jest to w testach językowych.

Niektóre platformy oferują o wiele więcej rodzajów pytań do testów (rys. 13.1.).

Uwaga!

Test w platformach e-learningowych jest często nazywany quizem.

Uwaga!

W Classroom występują jedynie dwa rodzaje zadań testowych: krótkiej odpowiedzi i jednokrotnego wyboru.

Rys. 13.1. Rodzaje zadań quizu w platformie Moodle

zasady tworzenia pytań ▸ Bez względu na to, którego rodzaju pytań użyjesz, **przestrzegaj pewnych zasad**:

1. Formułuj pytania i zdania oraz odpowiedzi w sposób jednoznaczny i precyzyjny.

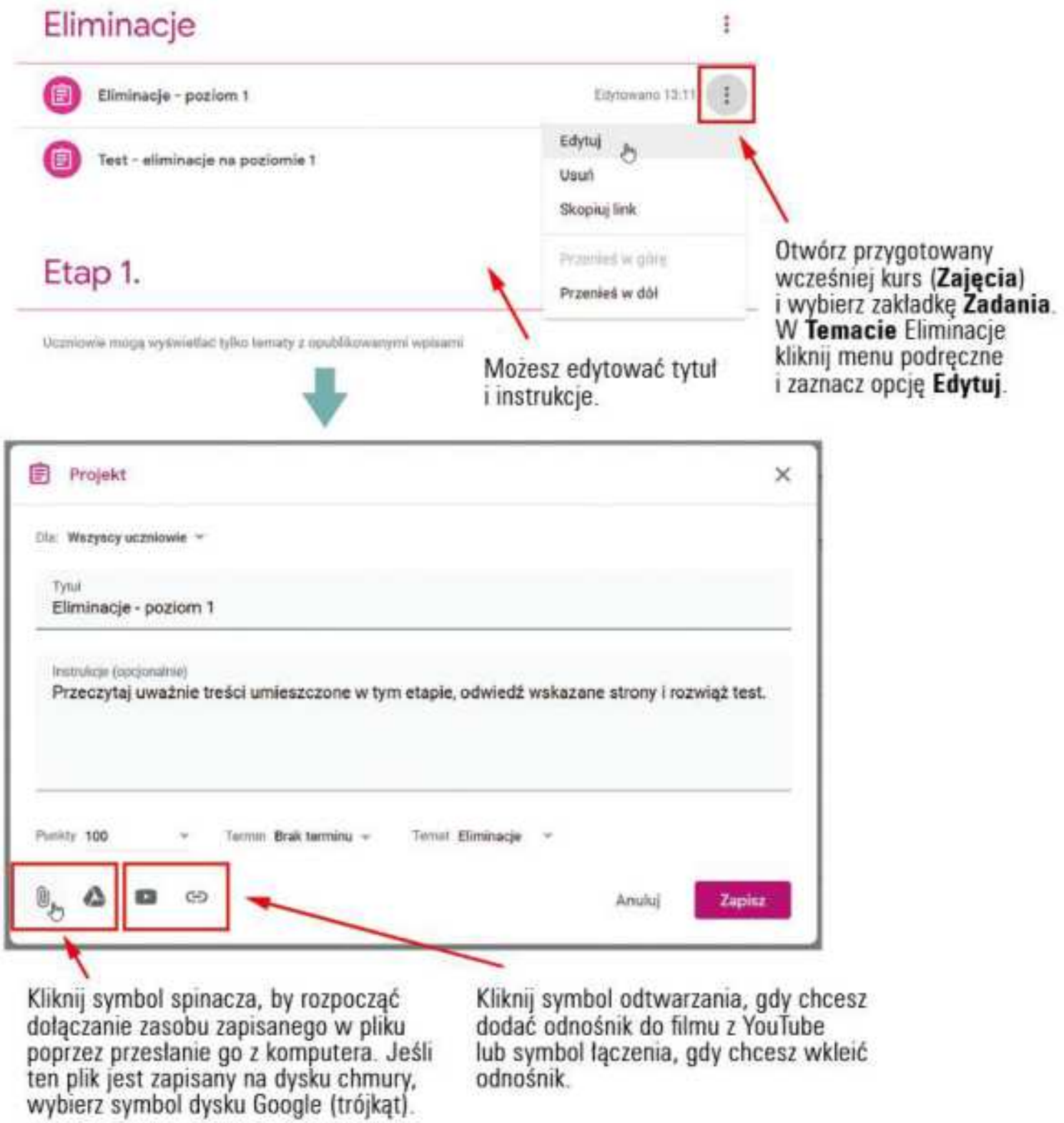
2. Nie wykraczaj poza zakres kursu.

3. Jeśli to możliwe i korzystne, dołączaj ilustracje do pytań.

4. Staraj się nie używać rozbudowanych zdań.

13.2. Konkurs matematyczny, czyli wypełniamy treściami przygotowany Temat platformy

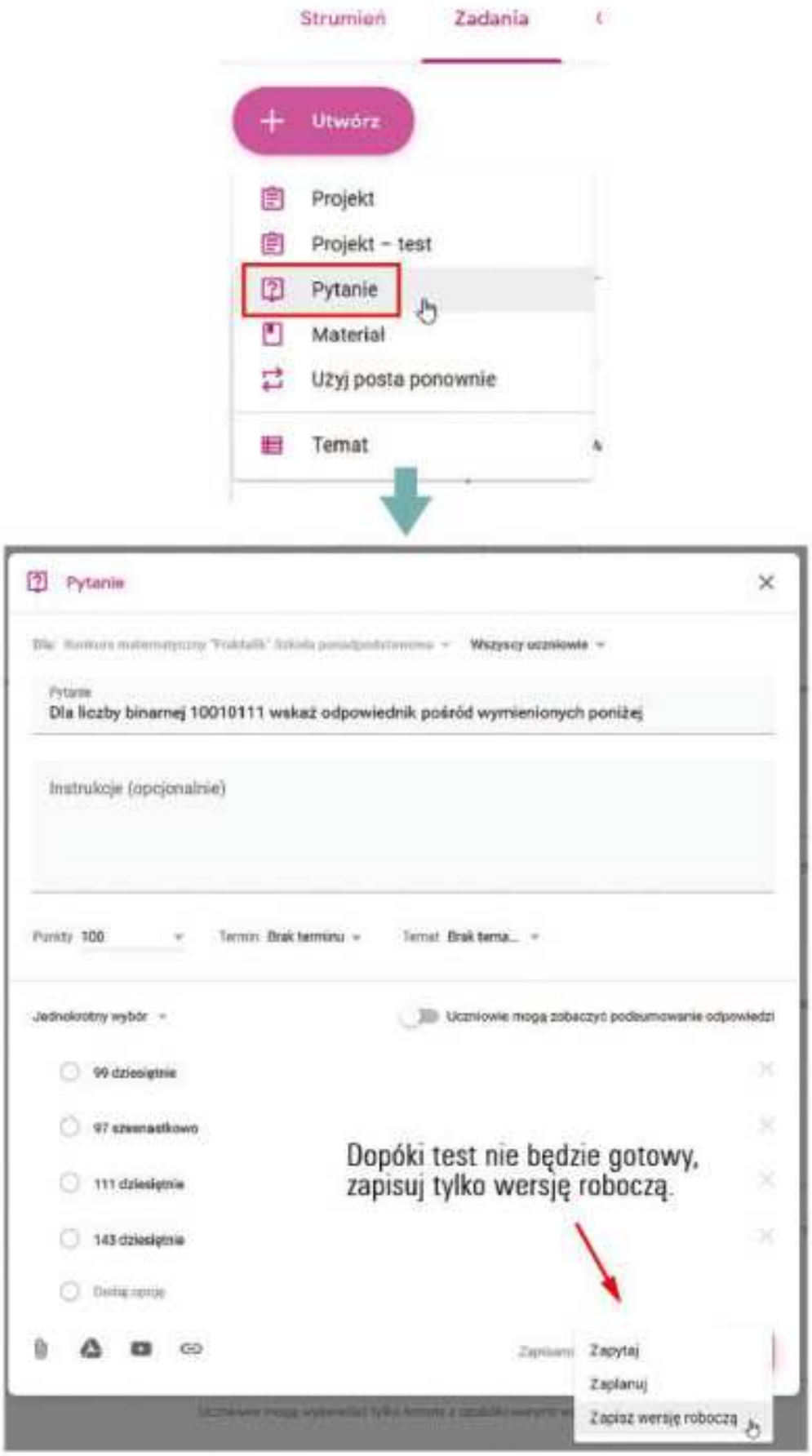
Załóżmy, że dokumenty tekstowe, rysunki, prezentacje, treści zadań itp. są przygotowane zgodnie z zasadami. Należy teraz użyć aplikacji Classroom i umieścić je w odpowiednich miejscach (rys. 13.2.).



Rys. 13.2. Umieszczanie zasobu edukacyjnego w aplikacji Classroom

Twoje zasoby będą widoczne w wybranym temacie. Podobnie należy uzupełnić temat Test – eliminacje, w którym powinny się znaleźć regulamin i przykładowe pytania. Nie udostępniaj ich uczestnikom konkursu, dopóki cały kurs nie będzie gotowy.

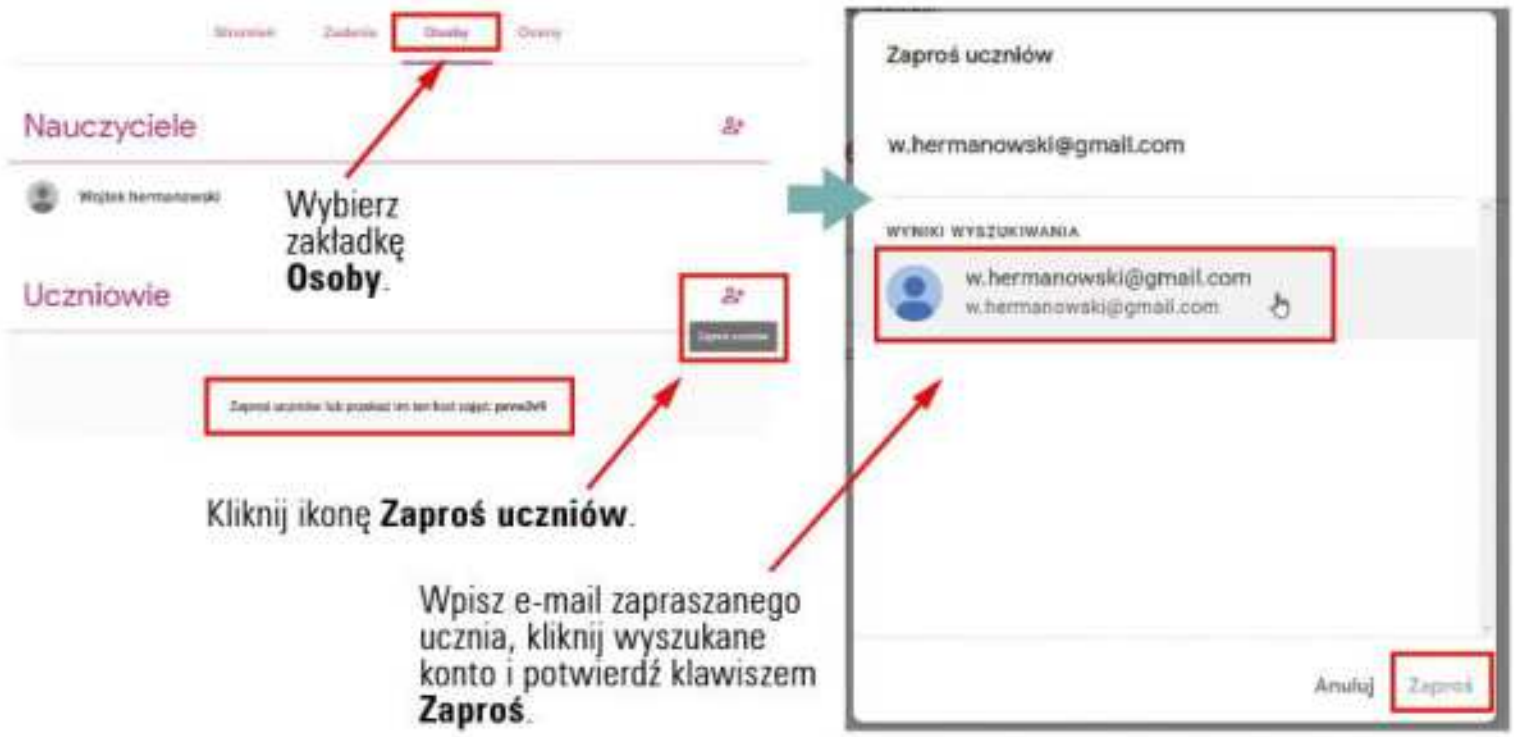
Przyszedł czas na zadania. W Classroom są dostępne tylko dwa rodzaje pytań. Wykorzystamy pytanie jednokrotnego wyboru (rys. 13.3.).



Rys. 13.3. Tworzenie pytania w aplikacji Classroom

13.3. Uczestnicy konkursu, czyli jak zaprosić uczniów do kursu

Gdy wszystkie zasoby kursu tworzącego konkurs matematyczny są już gotowe, należy zaprosić do niego uczestników. Trzeba skorzystać z odpowiedniego formularza (rys. 13.4.).



Rys. 13.4. Proces zapraszania uczniów do udziału w kursie w aplikacji Classroom

Proces zapraszania możesz zastąpić przesłaniem kodu kursu uczniom chętnym do udziału w kursie (konkursie) (rys. 13.4.).

ZADANIA DO WYKONANIA

1. Przeczytaj ponownie podrozdział 13.1. i sporządź w punktach plan dwóch wybranych rodzajów zasobów dla kursu do przedmiotu informatyka.
2. Utwórz kilka pytań z matematyki i umieść je w kursie utworzonym na podstawie rozdziałów 8. i 13. Wskaż poprawne odpowiedzi.
3. Zaproś do udziału w swoim kursie znajomych i poproś o rozwiązanie zadań quizu. Oceń ich pracę.
4. Umieść w kursie po jednym zasobie z każdego rodzaju źródła. Możesz posłużyć się darmowymi zasobami z internetu, np. filmami z YouTube oraz grafiką i dźwiękiem na licencji CC. Pamiętaj o przestrzeganiu praw autorskich.

PODSUMOWANIE LEKCJI

- **Zasoby do kursu** mogą pochodzić z dysku (będzie przesłany do chmury) lub z dysku chmury Google. Są to zazwyczaj filmy, zdjęcia albo odnośniki (link do strony z zasobem). ▸ s. 82
- W platformach e-learningowych można tworzyć różne rodzaje zadań. Podczas tworzenia pytań należy **przestrzegać zasad**. ▸ s. 82

14. Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach do e-learningu

NA TEJ LEKCJI:

- dowiesz się, czym jest współdzielenie plików w chmurze;
- nauczysz się tworzyć wspólny projekt ze współdzieleniem plików;
- przećwiczysz udostępnianie projektu na platformie e-learningowej.

Dynamiczny rozwój i coraz szerszy dostęp do zdobyczy technologii komputerowej i informacyjnej spowodował, że na szeroką skalę zaczęły się upowszechniać nowe formy kształcenia na odległość. Jedną z nich jest e-nauczanie, czyli e-learning. Nauczyciele czy osoby szkolące bardzo chętnie korzystają z tej formy popularyzacji wiedzy.

Do wykonywania zadań grupowych, czyli projektów, warto skorzystać z aplikacji, które oferują nam chmury – te najpopularniejsze to Google czy OneDrive. Dzięki tym aplikacjom można tworzyć wspólne prace, udostępniając sobie nawzajem pliki, na których pracujemy.

14.1. E-nauczanie, e-edukacja, e-learning

e-learning ▸ **E-nauczanie**, zwane również z języka angielskiego e-learningiem, to metoda kształcenia, podczas której wykorzystuje się technologie i różnego rodzaju media. Zazwyczaj jest to komputer z dostępem do internetu.

E-learning to najczęściej zmieniająca się forma nauczania na świecie. Coraz powszechniejsze są rozwiązania, w których nauczyciel wykorzystuje do nauki telefony czy inne przenośne urządzenia. Również w Polsce e-edukacja zdobywa coraz większe rzesze zwolenników, dlatego przybliżymy wam, jak możecie sami bądź podczas wykonywania wspólnego projektu zastosować narzędzia do współtworzenia zasobów e-learningowych.

14.2. Tworzenie projektu na współdzielonych plikach

Jeżeli chcesz właściwie stworzyć swój projekt i materiały, którymi będziesz się dzielić na platformach e-learningowych, zacznij pracę od konspektu. Na początku – jeżeli będziecie robić projekt w kilka osób – musicie odpowiedzieć na następujące pytania:

1. Czego projekt będzie dotyczył?
2. Kto będzie liderem?
3. Kto będzie odpowiedzialny za moderowanie poszczególnych części projektu?

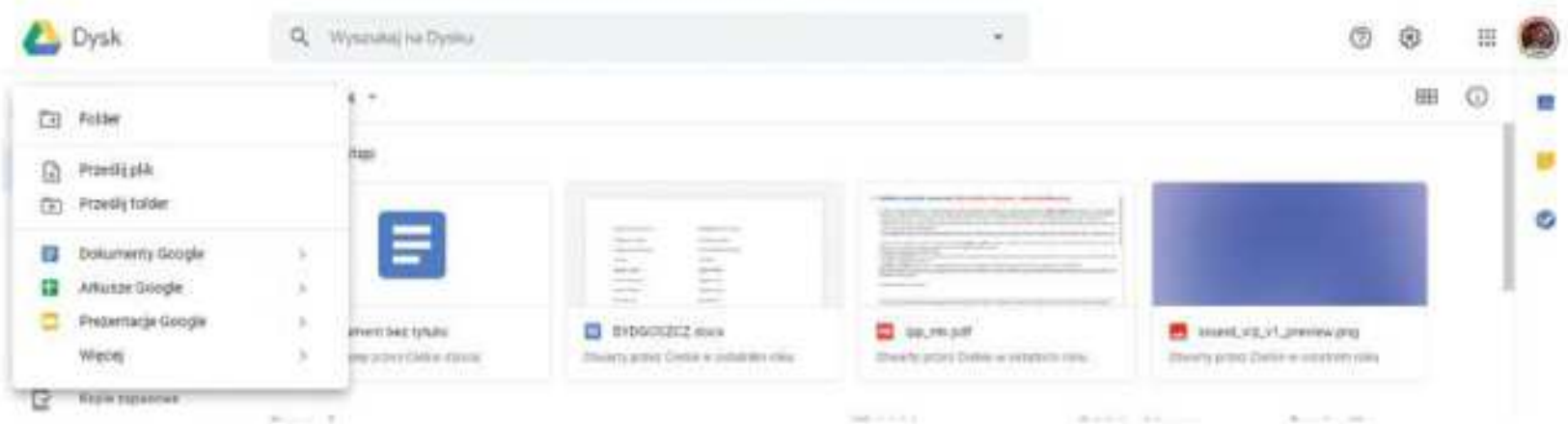
Lider projektu może go realizować w darmowej aplikacji **Dysk** w chmurze Google, gdzie jest możliwe współdzielenie plików z innymi. Pliki stworzone w tej aplikacji z łatwością umieścimy w aplikacji Classroom, omówionej w poprzednich rozdziałach.

Przykład. Projekt dotyczący tworzenia raportów w programie MS Access

Na **Dysku** Google tworzymy folder, w którym wykonamy nasz projekt. Aby to zrobić, musimy założyć konto na stronie *www.google.pl*. Następnie z chmury Google, znajdującej się również tej stronie, wybieramy aplikację **Dysk**.

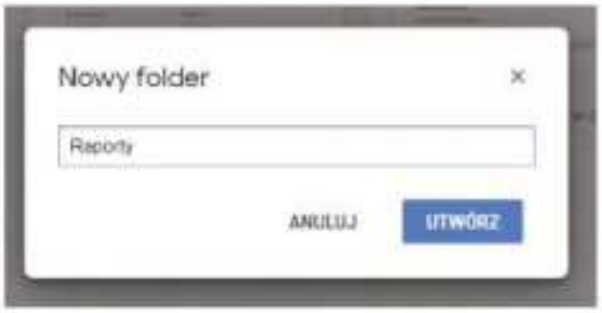


Rys. 14.1. Uruchomienie aplikacji **Dysk** na chmurze Google



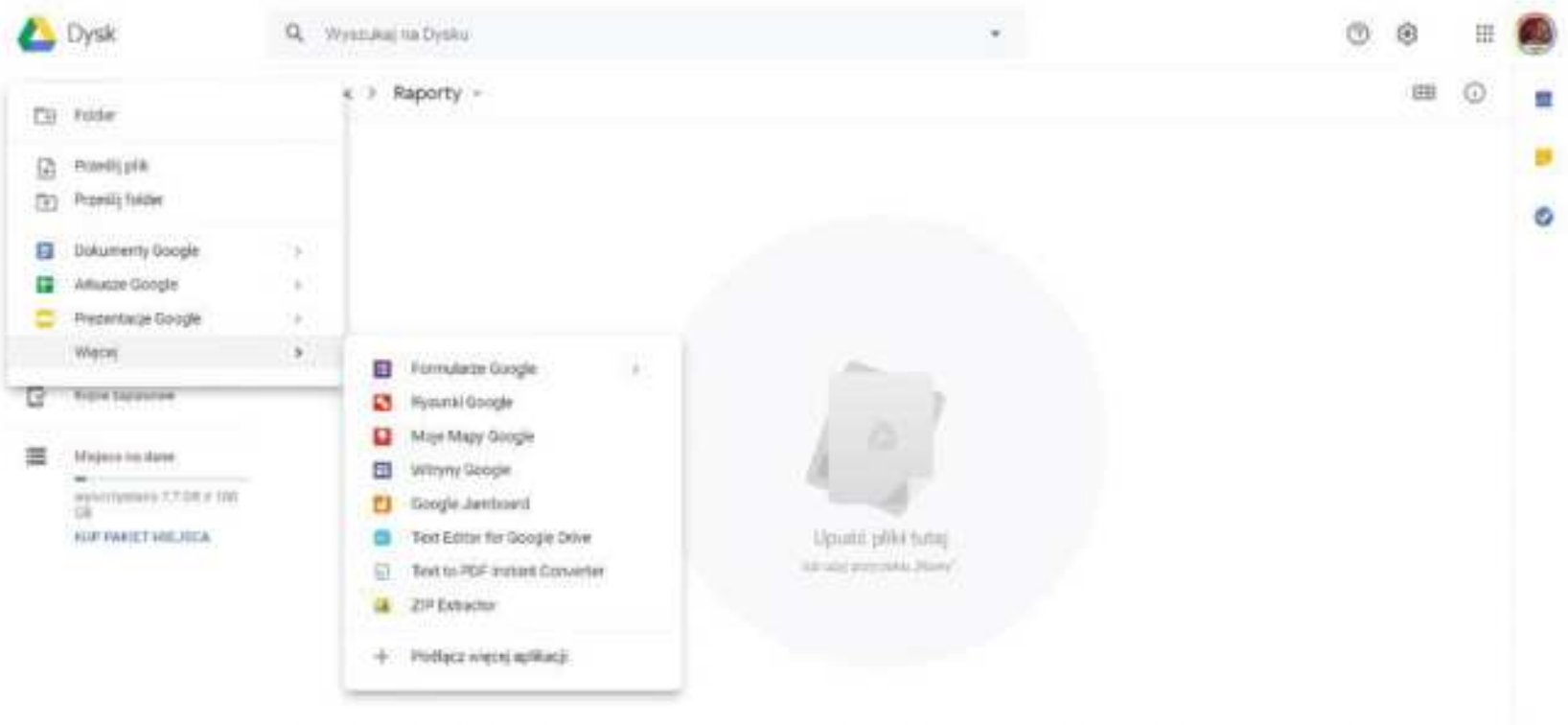
Rys. 14.2. Wybierz **Utwórz**, a następnie **Folder**, aby stworzyć folder projektu

W oknie, które pojawi się na środku okienka aplikacji, wpisujemy nazwę naszego projektu: **Raporty**.



Rys. 14.3. Tworzymy folder projektu

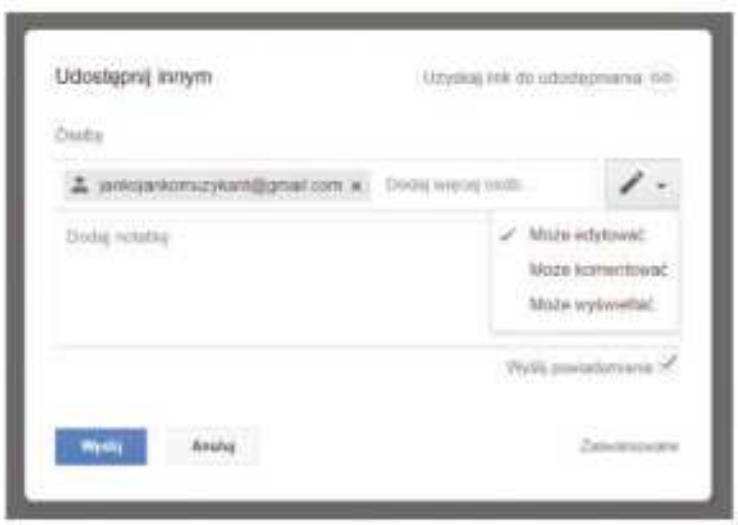
Otwieramy folder **Raporty**, w którym będziemy przechowywać współdzielone pliki. Mamy możliwość utworzenia **Dokumentu Google**, **Arkusza Google**, **Prezentacji Google** i wielu innych rodzajów plików, które będziemy współdzielić.



Rys. 14.4. Pliki, które możemy współdzielić w aplikacji **Dysk** na chmurze Google

współdzielenie dokumentu

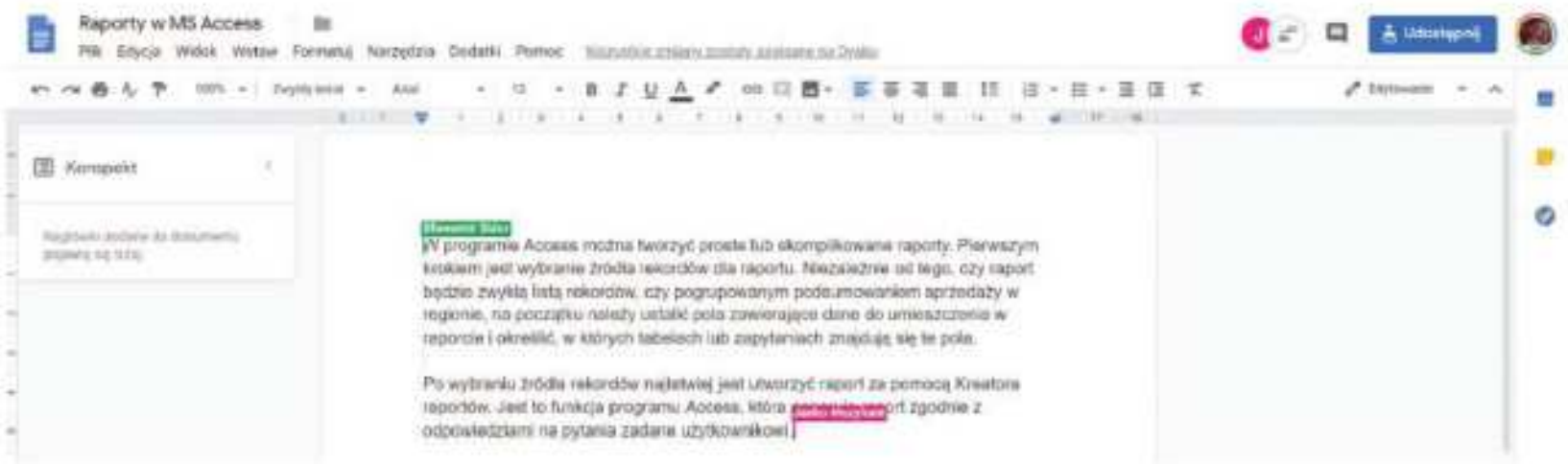
Wybieramy pusty dokument tekstowy, aby z innymi autorami redagować tekst dotyczący raportów w programie MS Access. Aby pozostałe osoby mogły uzupełniać nasz dokument, musimy im go udostępnić, klikając **Udostępnij** w górnym prawym roku otwartego dokumentu tekstowego. Pojawi się okno, w którym możemy dodać adresy mailowe osób, które uzyskają uprawnienia do edytowania pliku. W ten sposób utworzymy **współdzielony dokument**.



Rys. 14.5. Okno udostępniania pliku

Po kliknięciu **Uzyskaj link do udostępnienia** wygeneruje się link, który zostanie wysłany na wpisane adresy e-mail. Za pomocą tego linku współautorzy dokumentu otworzą go na swoich komputerach i będą mogli ten dokument edytować.

Podczas edycji dokumentu w czasie rzeczywistym będzie się pojawiał tekst na chmurce z informacją, który z użytkowników go wprowadził.



Rys. 14.6. Współdzielony dokument tekstowy

W górnym prawym rogu okna dokumentu mamy informację, którzy użytkownicy mają do niego dostęp.

Do naszego folderu możemy również dodawać pliki zewnętrzne – w naszym przypadku będzie to plik bazodanowy **baza.mdb**.

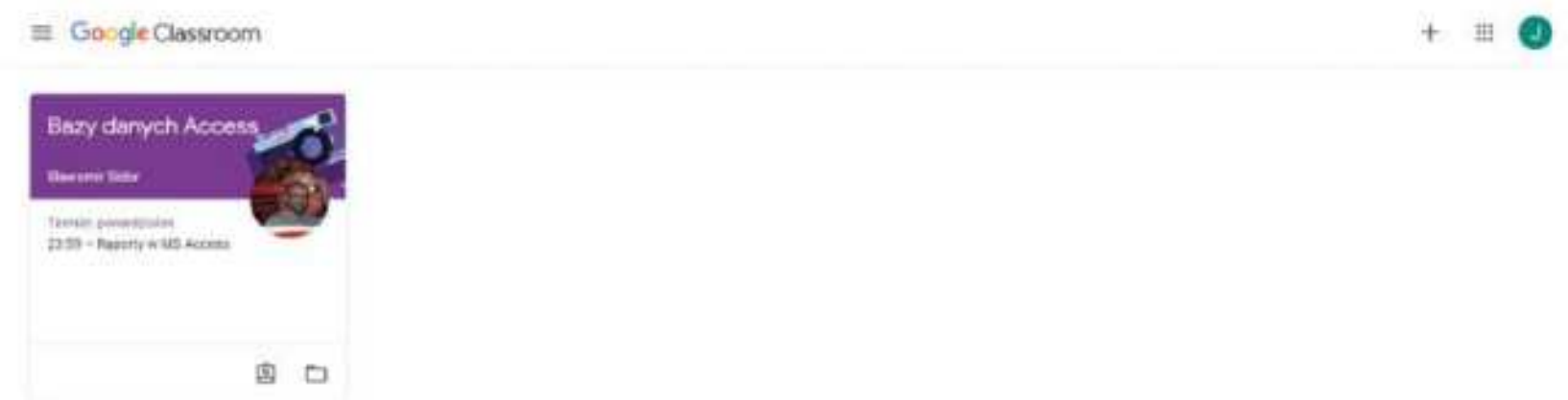
Po zakończeniu prac należy udostępnić projekt na platformie e-learningowej.

14.3. Udostępnianie projektu na platformie e-learningowej

Stworzyliśmy projekt, który nazwaliśmy „Raporty w MS Access". Czas najwyższy udostępnić go na platformie e-learningowej.

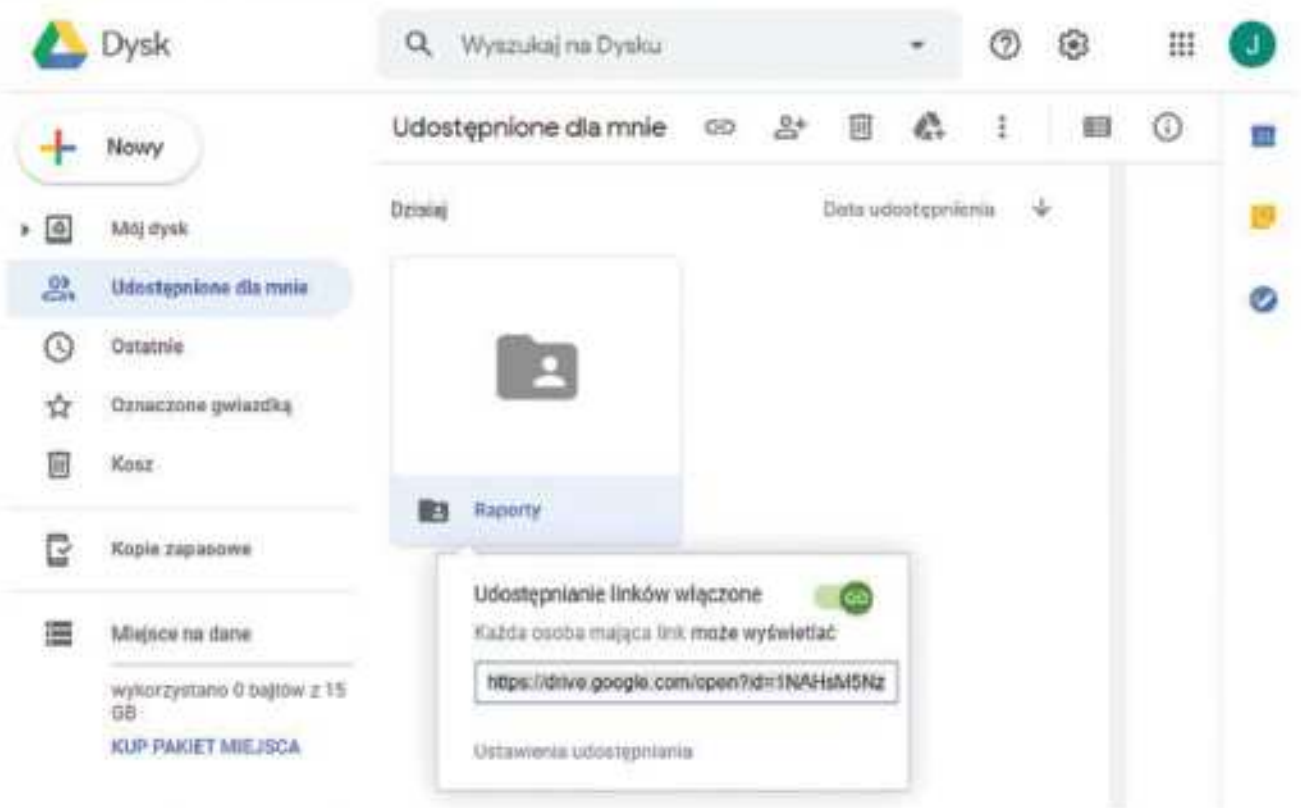
Uruchamiamy platformę edukacyjną – w naszym przypadku będzie to aplikacja Classroom w chmurze Google. To na niej umieścimy projekt dotyczący raportów w programie Access.

Gdy wejdziemy z naszego konta Google w aplikację Classroom, od razu nam się wyświetli lista zadań, które mamy do wykonania.



Rys. 14.7. Lista zadań w aplikacji Classroom

Do odpowiedniego zadania dołączymy nasz projekt. Zrobimy to poprzez udostępnienie linku, który można wygenerować. W tym celu wchodzimy na aplikację **Dysk** i w **Udostępnionych dla mnie** wyszukujemy nasz folder **Raporty**. Klikamy prawym przyciskiem myszy, aby otworzyło się menu kontekstowe, z którego wybieramy **Pobierz link do udostępnienia**. Po skopiowaniu linku wracamy do aplikacji Classroom.



Rys. 14.8. Tworzenie linku udostępniającego folder

Otwieramy zadanie dotyczące baz danych i udostępniamy w nim link do naszego projektu. Aby to zrobić, w górnym menu klikamy **Zadania** i wybieramy to dotyczące raportów.



Rys. 14.9. Wybór zadania, w którym umieszczamy nasz projekt

Po otwarciu zadania klikamy **Wyświetl projekt**, z menu po prawej stronie wybieramy **Dodaj lub utwórz**, a następnie **Link**. W ostatnim kroku wklejamy link do naszego projektu i klikamy **Dodaj link**. Aby ta operacja się powiodła, musi ją wykonać właściciel folderu.



Rys. 14.10. Dodawanie projektu do zadania

Po dodaniu linku w menu z prawej strony musimy potwierdzić, że chcemy oddać zadanie do oceny nauczycielowi, klikając przycisk **Oddaj**. Po zatwierdzeniu, do czasu, kiedy nie minie termin oddania zadania, będzie można je jeszcze wycofać i wprowadzić poprawki, a potem ponownie oddać do oceny.



Rys. 14.11. Zadanie umieszczone na platformie e-learningowej

ZADANIA DO WYKONANIA

1. Korzystając z chmury Google, stwórz z trzema osobami z klasy współdzielony dokument tekstowy, w którym będą zawarte informacje dotyczące historii systemu operacyjnego Windows.
2. Korzystając z chmury Google, stwórz z koleżanką lub kolegą współdzielony plik arkusza kalkulacyjnego, w którym obliczycie, ile może was kosztować wycieczka z waszego miasta do Krakowa. Jedna osoba niech obliczy koszty podróży, a druga koszty noclegów. Wasza wycieczka ma trwać 5 dni.
3. Korzystając z chmury Google, stwórz z koleżanką lub kolegą plik współdzielony prezentacji multimedialnej o najważniejszych zabytkach w waszym mieście lub regionie.

PODSUMOWANIE LEKCJI

- **E-nauczanie**, zwane również z języka angielskiego e-learningiem, to metoda kształcenia, podczas której wykorzystuje się technologie i różnego rodzaju media. Najczęściej jest to komputer z dostępem do internetu. s. 86
- Wykorzystując aplikacje, które są udostępniane w chmurach, takich jak Google czy OneDrive, można tworzyć **wspólne dokumenty** dzięki ich współdzieleniu. Dokumenty takie można później umieszczać na platformach edukacyjnych w ramach wspólnych projektów. s. 88

PODSUMOWANIE PRZED SPRAWDZIANEM

10. Informatyka pokonuje schody, czyli nikt nie powinien być wykluczony

s. 62
- **Aplikacje pierwszej pomocy** mogą uratować życie. Warto się z nimi zapoznać i w razie potrzeby z nich skorzystać.

s. 65
- Osoby do niedawna mające poczucie wykluczenia ze względu na swoją niepełnosprawność mogą korzystać z ułatwień, jakie dają programy i urządzenia komputerowe.

s. 66
- **Translatory** tłumaczą słowa pisane, mówione oraz sfotografowane, co znacznie ułatwia komunikację z ludźmi mówiącymi w obcym języku.

11. Platformy uczą, czyli rola e-learningu w nauce i pracy

s. 69
- Zdalne nauczanie za pośrednictwem internetu nazywa się **e-learningiem**.

s. 71
- Popularna platforma **Moodle** jest darmowa i można ją instalować na serwerach z aktywnym systemem baz danych.

s. 71
- **SCORM** jest uniwersalnym formatem zapisu treści edukacyjnych akceptowalnym przez wiele platform e-learningowych.

s. 73
- Istnieje wiele programów służących kształceniu na odległość, w tym komunikatory i programy multimedialne instalowane lokalnie.

s. 73
- Zastosowanie platformy zależy tylko od inwencji i pomysłowości autora kursu.

12. Możesz być administratorem, czyli jak zarządzać platformą e-learningową

s. 75
- Dobrze zaplanowany kurs pozwoli na zbudowanie odpowiedniej struktury w platformie. **Wybór platformy** e-learningowej do tworzenia własnych kursów zależy od wymagań, jakie stawia kurs.

s. 76
- Kursy lub lekcje o nieskomplikowanej strukturze możesz wykonać i udostępnić w **aplikacji Classroom** w chmurze Google.

13. Zasoby i testy, czyli wypełniamy kursy treścią

s. 82
- **Zasoby do kursu** mogą pochodzić z dysku (będzie przesłany do chmury) lub z dysku chmury Google. Są to zazwyczaj filmy, zdjęcia albo odnośniki (link do strony z zasobem).

s. 82
- W platformach e-learningowych można tworzyć różne rodzaje zadań. Podczas tworzenia pytań należy **przestrzegać zasad**.

14. Dzielimy się wiedzą, czyli współtworzymy zasoby udostępniane na platformach do e-learningu

s. 86
- **E-nauczanie**, zwane również z języka angielskiego e-learningiem, to metoda kształcenia, podczas której wykorzystuje się technologie i różnego rodzaju media. Najczęściej jest to komputer z dostępem do internetu.

s. 88
- Wykorzystując aplikacje, które są udostępniane w chmurach, takich jak Google czy OneDrive, można tworzyć **wspólne dokumenty** dzięki ich współdzieleniu. Dokumenty takie można później umieszczać na platformach edukacyjnych w ramach wspólnych projektów.

IV. Bazy danych

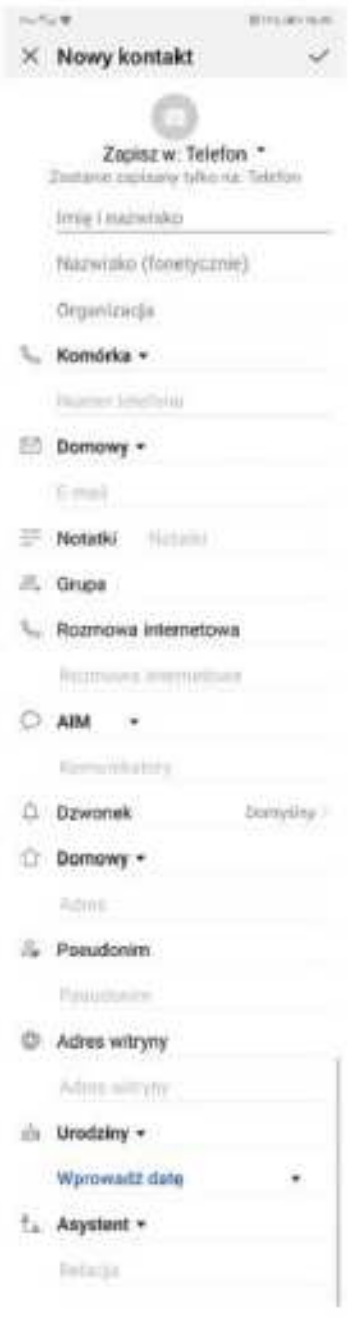


- Jak gromadzić informacje, czyli komputerowe bazy danych
- Tabele i formularze, czyli jak utworzyć bazę danych
- Relacje i pytania, czyli jak uczynić bazę użyteczną
- Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access
- Formularze, czyli ułatwiamy sobie wprowadzanie informacji do bazy danych
- Czy to możliwe, czyli baza danych w arkuszu Excel
- Wiedza w sieci, czyli internet mądrych ludzi
- Wirtualne serwery, czyli instalujemy Apache
- Język zapytań, czyli poznajemy język SQL
- Zadajemy pytania, czyli kwerendy w języku SQL
- Więcej pytań, czyli jeszcze o kwerendach w języku SQL
- Podzapytania, czyli instrukcje modyfikujące dane w języku SQL
- Modyfikacje, czyli zarządzanie bazą danych
- Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe

15. Jak gromadzić informacje, czyli komputerowe bazy danych

NA TEJ LEKCJI:

- dowiesz się, co to jest komputerowa baza danych;
- poznasz przykłady baz danych;
- poznasz podstawowe pojęcia dotyczące baz danych.



Czy wiesz, że dawniej do tworzenia baz danych używano papieru? Zapisywano na kartkach wszystkie dane i sortowano je według określonego klucza (identyfikatora). Przykładami takich baz są do dziś istniejące katalogi w bibliotekach i archiwach. Występujące w nich identyfikatory znalazły także zastosowanie w bazach komputerowych. Współczesne aplikacje przystosowane do gromadzenia informacji znacznie ułatwiają tworzenie struktur i szybkie dotarcie do właściwych treści. Ich poznawanie zaczniemy od **MS Access**.

15.1. Informacje wokół nas, czyli przykłady baz danych

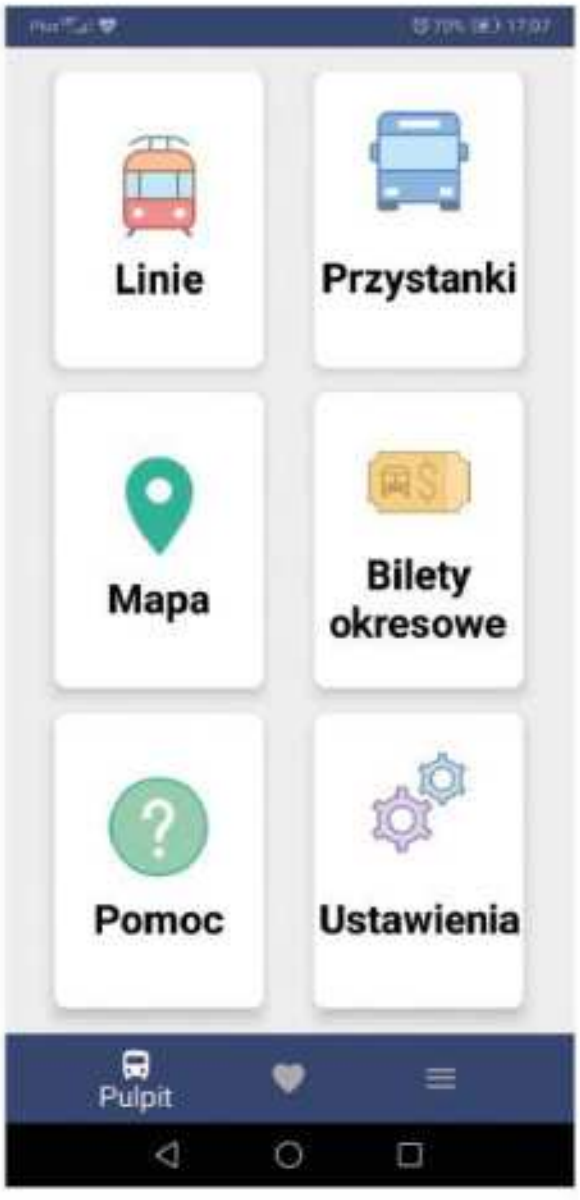
Czy umiesz wskazać miejsca w sieci lub aplikacje, w których występują bazy danych? Często nie zdajemy sobie sprawy z tego, że są one zaszyte w usługach sieciowych, programach itp. W twoim smartfonie masz co najmniej jedną taką bazę – spis kontaktów. Kontakty możesz dodawać, kasować i grupować. Nie jest to jednak zwykły spis numerów telefonów. Do każdego kontaktu możesz dopisać wiele informacji (rys. 15.1.).

Rys. 15.1. Przykład danych, które można zapisać w bazie danych – spisie kontaktów smartfona

Elektroniczny dziennik lekcyjny jest skomplikowaną i rozbudowaną bazą danych. Dane te wprowadza administrator, ale korzysta z nich każdy uczeń, nauczyciel i rodzic. Obsługa jest łatwa, bo projektant zadbał o przyjazny interfejs. Użytkownik wybiera po prostu odpowiednie opcje na ekranie, a oprogramowanie samodzielnie pobiera z różnych miejsc dane i podaje je w łatwej do odczytania formie.

Część stron internetowych nie mogłaby istnieć bez systemu bazy danych zainstalowanego na serwerze. Użytkownik nie zdaje sobie nawet z tego sprawy. Wie o tym za to każdy, kto tworzył stronę z zastosowaniem CMS, czyli systemu zarządzania treścią.

Aplikacje pomagające sprawnie poruszać się po mieście to także bazy danych. Przykładem może być oprogramowanie wyświetlające dane o systemie komunikacji miejskiej, korzystające ze zintegrowanego systemu zarządzania ruchem ITS. Dzięki aktualizowanej na bieżąco bazie pasażer może poznać realny czas odjazdu z dowolnego przystanku (rys. 15.2.). Każda ikona jest w zasadzie zapytaniem złożonym bazie.



Rys. 15.2. KomByd – aplikacja dla pasażerów komunikacji miejskiej w Bydgoszczy

Gdy używasz smartwatcha lub opaski treningowej, także korzystasz z bazy danych. Aplikacja współpracująca z takim urządzeniem zapisuje dane w uporządkowany sposób. W każdej chwili są one gotowe do odczytu.

Strona z rozkładem jazdy pociągów to też baza danych. Możemy zadać jej pytanie o połączenie z dowolną stacją. Oprogramowanie zbierze dla nas wszystkie informacje i wyświetli je na ekranie.

Jeśli uważnie rozejrzysz się wokół, przejrzysz kilka zasobów sieciowych itp., z łatwością zidentyfikujesz miejsca, w których zastosowano bazę danych.

15.2. Garść informacji, czyli z czego jest zbudowana baza danych

Zanim zaczniesz budowanie własnych baz danych, poznaj kilka pojęć, których zrozumienie ułatwi ci zgłębienie mechanizmów działania takich zbiorów.

elektroniczna baza danych ▸ Elektroniczną **bazą danych** nazywa się uporządkowany zbiór informacji z określonym przez program komputerowy sposobem zapisu, wyszukiwania i dostępu do poszczególnych danych. Program do tworzenia baz i system, w którym one działają, to **system zarządzania bazą danych**.

Z elektronicznych baz danych korzystasz na co dzień. Każda wyszukiwarka internetowa jest wielkim i dynamicznym zbiorem informacji. Posługujesz się także małymi bazami, np. spisem kontaktów w komunikatorze, telefonie, listą adresów w kliencie poczty itp.

Rozbudowane systemy zarządzania bazami danych są także dostępne w postaci aplikacji pozwalających na samodzielne tworzenie tabel z danymi i relacji pomiędzy nimi. Nie trzeba być programistą, by z nich korzystać. Niektóre pakiety biurowe zawierają taki program. Przykładem może być MS Access zawarty w Microsoft Office (lub Office 365) oraz darmowe LibreOffice Base i OpenOffice.org Base.

tabela ▸ **Tabela** jest wydzielonym w bazie logicznym zbiorem danych. Chociaż w systemach baz często dane pojawiają się na ekranie w postaci tabelki, to w rzeczywistości jest to tylko ich reprezentacja przyjazna dla użytkownika. Ten element bazy nazwano tak z powodu sposobu uporządkowania danych w wierszach i kolumnach.

rekord ▸ **Rekord** w relacyjnej bazie danych to jeden wiersz w tabeli.

Relacje w bazie danych to powiązania, dzięki którym dane z różnych tabel, powiązane ze sobą tematycznie, mogą być wspólnie udostępniane. Łatwo zapamiętać tę definicję, ponieważ podobne znaczenie tego wyrazu funkcjonuje także w języku potocznym, np. jeden rekord klasy (z parametrami takimi jak profil, rok rozpoczęcia, itp.) jest powiązany z wieloma rekordami uczniów tej klasy. Nie musimy przy każdym uczniu powtarzać informacji o klasie. Wystarczy relacja, aby przy uczniu wyświetlić dane o jego klasie.

Formularz to narzędzie pozwalające tworzyć interfejsy do wprowadzania i edycji danych. Stąd właśnie jego nazwa.

Kwerenda jest zapytaniem do bazy danych. Na jej podstawie, wykorzystując relacje, oprogramowanie bazy odnajduje dane i tworzy z nich odpowiedź wyświetlaną użytkownikowi.

Raport służy do wyświetlania danych aktualnie znajdujących sie w bazie

Budowa bazy danych zależy od projektanta i wytycznych, które dostał od zamawiającego. Podstawowe czynności, jakie powinna umożliwiać baza danych, to:

1. dodawanie i usuwanie danych,
2. edytowanie i modyfikacja danych,
3. przeglądanie danych na różne sposoby,
4. udostępnianie danych może obejmować eksport, np. do innej bazy lub aplikacji.

ZADANIA DO WYKONANIA

1. Przejrzyj aplikacje twojego komputera i wytypuj te, które według ciebie są oparte na bazie danych. Wybierz jedną z nich i – korzystając z edytora tekstu – opisz, na czym polega wykorzystanie bazy w tym programie.
2. Przejrzyj aplikacje twojego smartfona i podobnie jak w zadaniu 1. wytypuj te, które twoim zdaniem są oparte na bazie danych. Wybierz jedną z nich i – korzystając z edytora tekstu – opisz, na czym polega wykorzystanie bazy w tym programie.
3. Uruchom sklep twojego telefonu, np. Sklep Play z systemu Android. Wpisz w wyszukiwarce *baza danych*. Sprawdź, jakie aplikacje są polecane. Poznaj ich zastosowanie. Opisz w zeszycie cechy jednej z nich, np. z grupy baz danych o lasach.

PODSUMOWANIE LEKCJI

- Elektroniczną **bazą danych** nazywa się uporządkowany zbiór informacji z określonym przez program komputerowy sposobem zapisu, wyszukiwania i dostępu do poszczególnych danych. Dane zapisywane są na nośnikach elektronicznych (np. dyskach), a dostęp do nich następuje za pomocą aplikacji komputerowej. ◂ s. 96
- Podstawowymi elementami bazy danych są **tabele**, a ich wiersze to **rekordy**. ◂ s. 96
- Powiązania między rekordami to **relacje**. ◂ s. 96

16. Tabele i formularze, czyli jak utworzyć bazę danych

NA TEJ LEKCJI:

- utworzysz tabele i formularze bazy danych w programie Access;
- dowiesz się, czym są pola kombi.

Programy do tworzenia baz danych można znaleźć w najpopularniejszych pakietach biurowych – komercyjnym MS Office i darmowym LibreOffice. Kolejność czynności podczas tworzenia baz w obu programach jest podobna. Tworzenie baz danych za pomocą aplikacji przedstawiliśmy na przykładzie bazy Access z pakietu MS Office.

16.1. Miejsce na dane, czyli tworzymy tabele

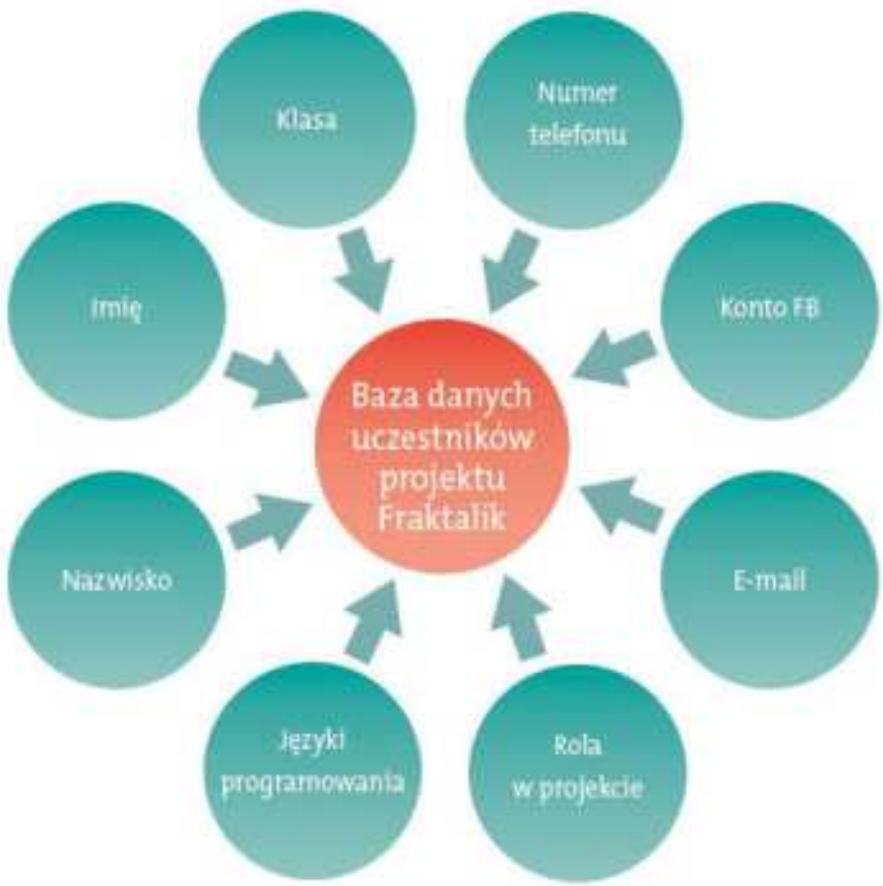
Szybkość dostępu do właściwych informacji zależy od poprawnego zaprojektowania bazy. Te duże, zawierające ważne dane (np. w urzędach i bankach), są projektowane przez wyspecjalizowane zespoły w firmach informatycznych. Zapewniają one nie tylko niezawodność, lecz także bezpieczeństwo danych. Wiele z baz jest dostępnych za pośrednictwem sieci,

a więc ich serwery muszą być odpowiednio zabezpieczone. Nie wszystkim bazom stawia się jednak tak wysokie wymagania. Niektóre działają lokalnie, np. są zainstalowane na komputerze ich użytkownika. Taką właśnie bazę utworzysz na podstawie informacji zawartych w tym rozdziale.

Ważnym etapem tworzenia bazy jest właściwy wybór narzędzia, np. aplikacji. Dla potrzeb budowy lokalnej bazy odpowiednie będą programy zawarte w niektórych pakietach biurowych, np. Access z MS Office lub Base z LibreOffice. Ich narzędzia i intuicyjne menu ułatwiają tworzenie poszczególnych elementów bazy. Programy te pomagają także w zarządzaniu, np. we wprowadzaniu, wyszukiwaniu i wyprowadzaniu danych. Dzięki takim programom użytkownik nieznający języków programowania może samodzielnie zbudować w pełni funkcjonalną bazę. Działanie programu najlepiej poznać na przykładzie.

projekt bazy danych

Załóżmy, że pracujesz nad projektem, w którym jest zaangażowanych wiele osób. Komunikacja pomiędzy członkami zespołu jest jednym z fundamentów dobrej współpracy. Podstawowe dane warto zebrać w bazie danych (rys. 16.1.).



Rys. 16.1. Przykładowe dane dla tworzonej bazy uczestników projektu

Skoro ustaliliśmy, jakie dane będzie zawierała twoja baza, można przystąpić do jej budowania.

tabele

Wiesz już, że podstawowymi obiektami, w których gromadzi się dane w bazie, są tabele. Ich wiersze nazwano **rekordami**, a kolumny – **polami**. Od projektowania tabel zaczniemy więc tworzenie naszej bazy (rys. 16.2 na stronie obok.). Ponieważ ma ona gromadzić dane związane z projektem, nazwiemy ją **Fraktalik**.

pomoc w programie

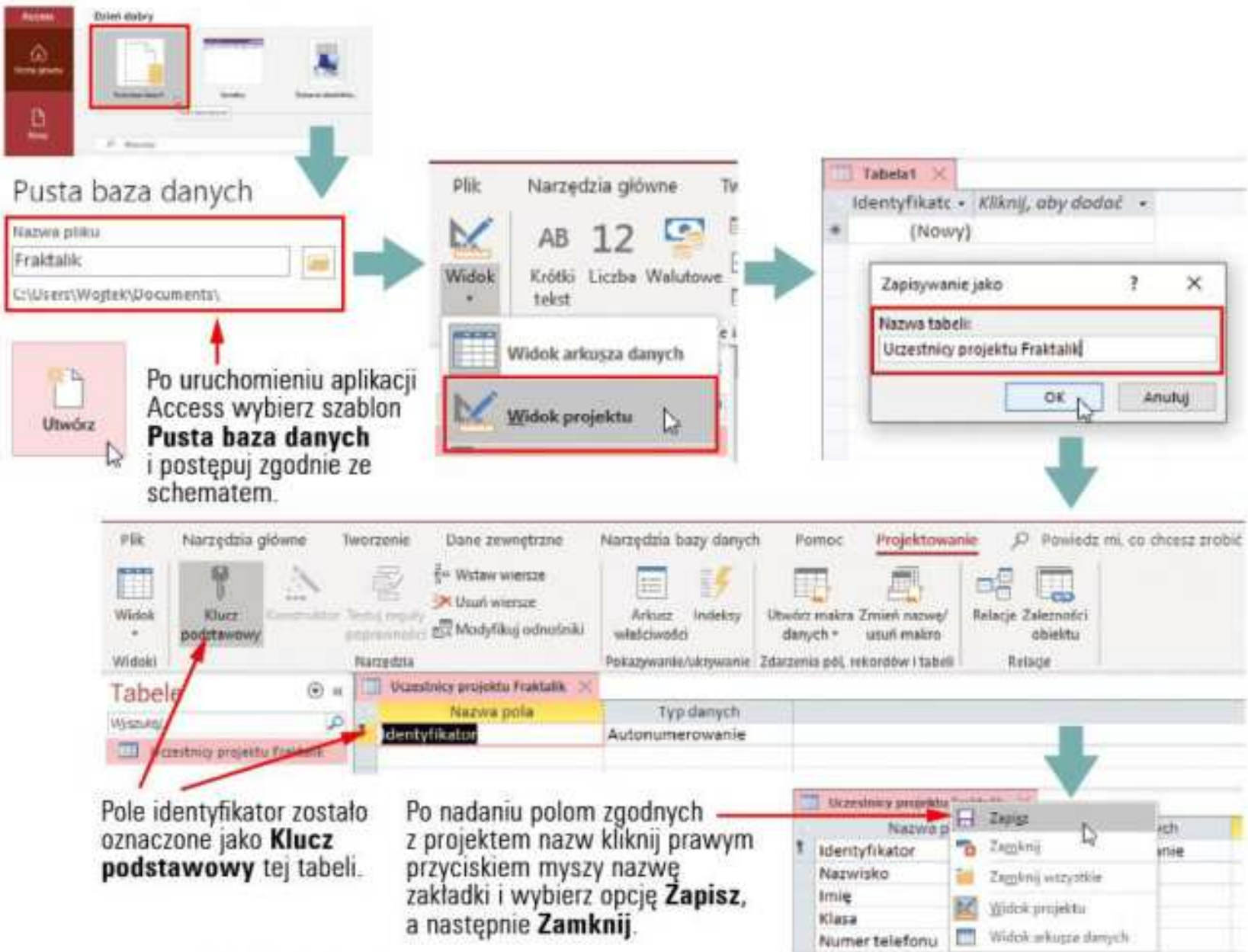
Zanim przystąpisz do pracy, warto skorzystać z opcji pomocy bazy Access i przypomnieć sobie znaczenie kluczy na podstawie przytoczonych przykładów (w tłumaczeniu na stronie jest używana nazwa **klucz podstawowy**):

Tabela może mieć tylko jeden klucz podstawowy. Klucz podstawowy składa się z pola lub kilku pól jednoznacznie identyfikujących poszczególne rekordy przechowywane w tabeli. Często funkcję klucza podstawowego pełni unikatowy numer, na przykład numer identyfikacyjny, numer seryjny lub kod. Na przykład w tabeli **klienci** do każdego klienta może być przypisany unikatowy identyfikator klienta. Pole zawierające identyfikator klienta jest kluczem podstawowym tabeli **klienci**. W przypadku gdy klucz podstawowy zawiera więcej niż jedno pole, najczęściej składa się z istniejących wcześniej pól, które łącznie tworzą unikatowe wartości. Na przykład kluczem podstawowym tabeli z danymi osób może być kombinacja nazwiska, imienia i daty urodzenia.

Tabela może mieć również **klucz obcy** lub kilka takich kluczy. Klucz obcy zawiera wartości odpowiadające wartościom w kluczu podstawowym innej tabeli. Przykładem może być tabela **zamówienia**, w której dla każdego zamówienia istnieje numer identyfikacyjny klienta odpowiadający rekordowi w tabeli **klienci**. Pole identyfikatora klienta jest kluczem obcym tabeli **zamówienia**.

źródło: Microsoft, https://support.office.com/pl-pl/article/wprowadzenie-do-tabel-78ff21ea-2f76-4fb0-8af6-c318d1ee0ea7

W naszym przykładzie dla tabeli **Uczestnicy projektu Fraktalik** kluczem podstawowym jest **Identyfikator** (nazwa nadana domyślnie – można ją zmienić). Tabela ta nie ma klucza obcego.



Rys. 16.2. Tworzenie tabeli w bazie Access

W podobny sposób utwórz pozostałe dwie tabele naszej bazy: **Cechy uczestnika** oraz **Znajomość programów i języków programowania** (rys. 16.3.).

Rola w projekcie

Nazwa pola	Typ danych
Identyfikator	Autonumerowanie
Lider	Krótki tekst
Programista	Krótki tekst
PR	Krótki tekst
Opiekun	Krótki tekst

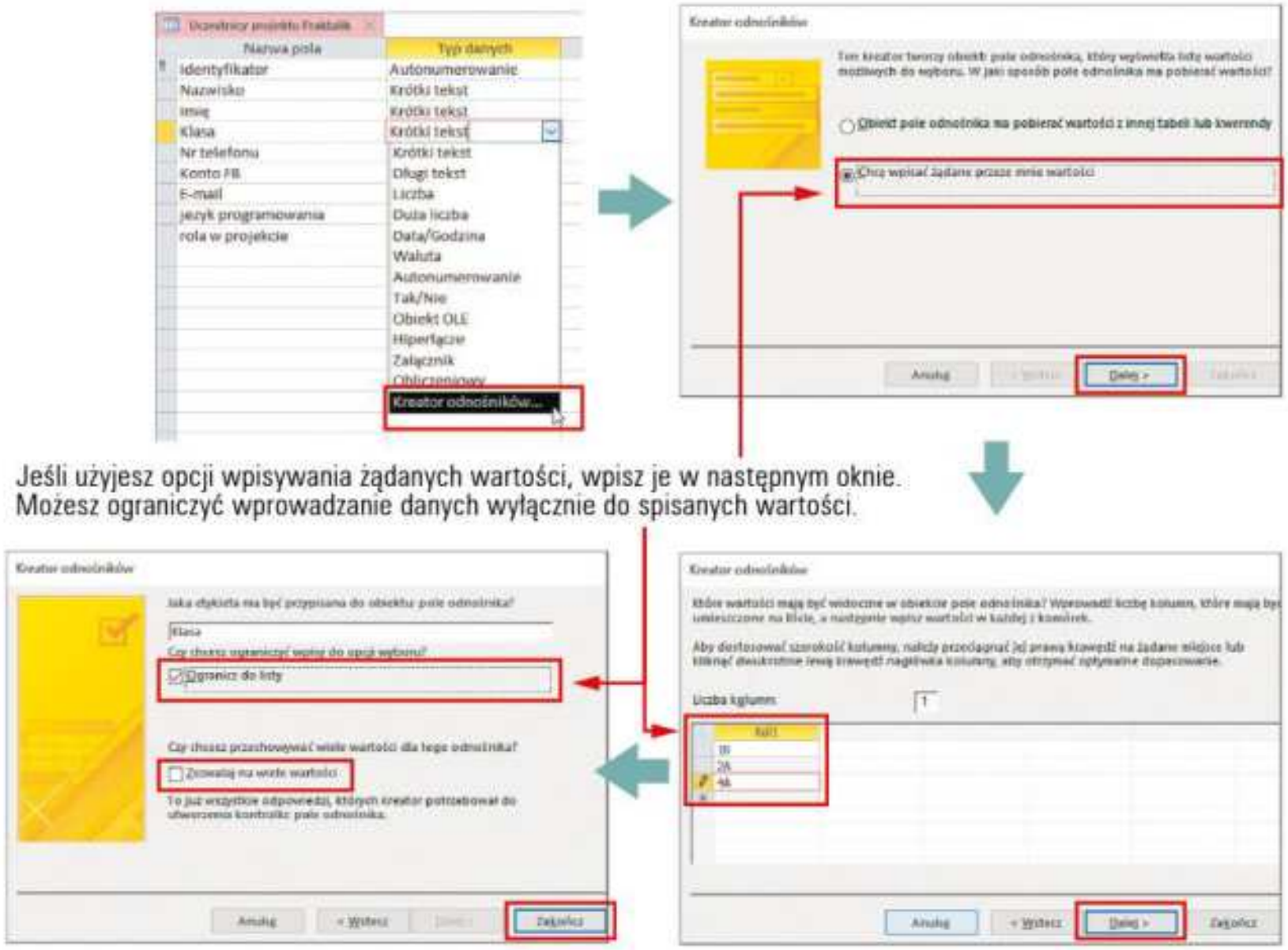
Języki programowania

Nazwa pola	Typ danych
Identyfikator	Autonumerowanie
C/C++	Krótki tekst
JavaScript	Krótki tekst
PHP	Krótki tekst
HTML 5	Krótki tekst

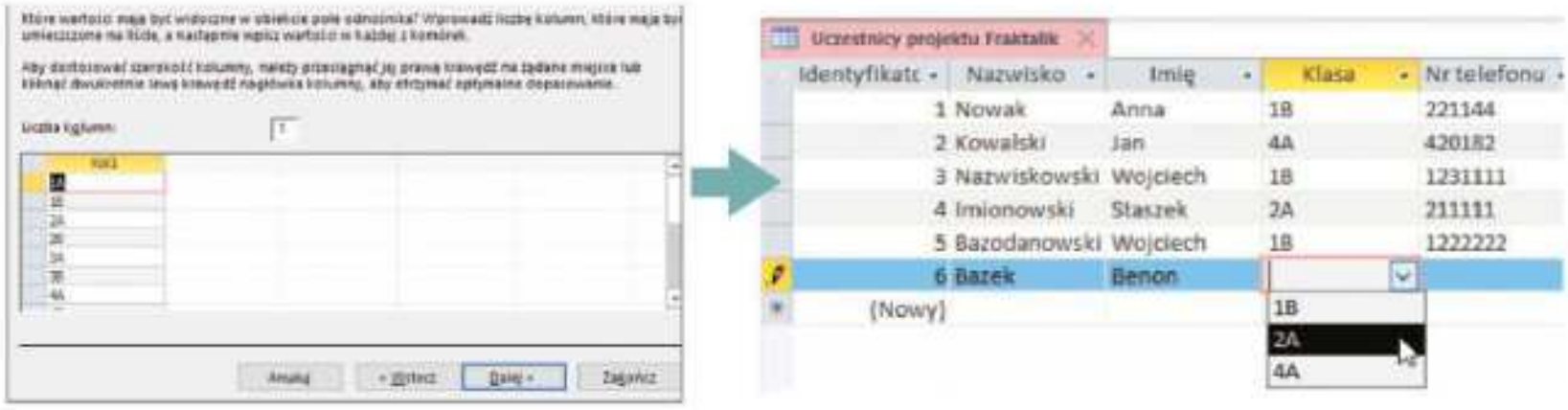
Rys. 16.3. Pozostałe tabele bazy Fraktalik zgodne z projektem

W tabelach **Rola w projekcie** oraz **Języki programowania** zapisane zostaną cechy uczestnika projektu – jego rolę w projekcie i znajomość danego języka programowania.

kreator ▸ Aby ułatwić wprowadzanie tych danych, można użyć **Kreatora odnośników** (rys. 16.4.). W naszej bazie takim polem może być Klasa. Dzięki temu ujednolicone zostaną nazwy klas.



Rys. 16.4. Użycie kreatora odnośników dla pól tabeli



Rys. 16.5. Użycie **Kreatora** odnośników dla pola **Klasa**

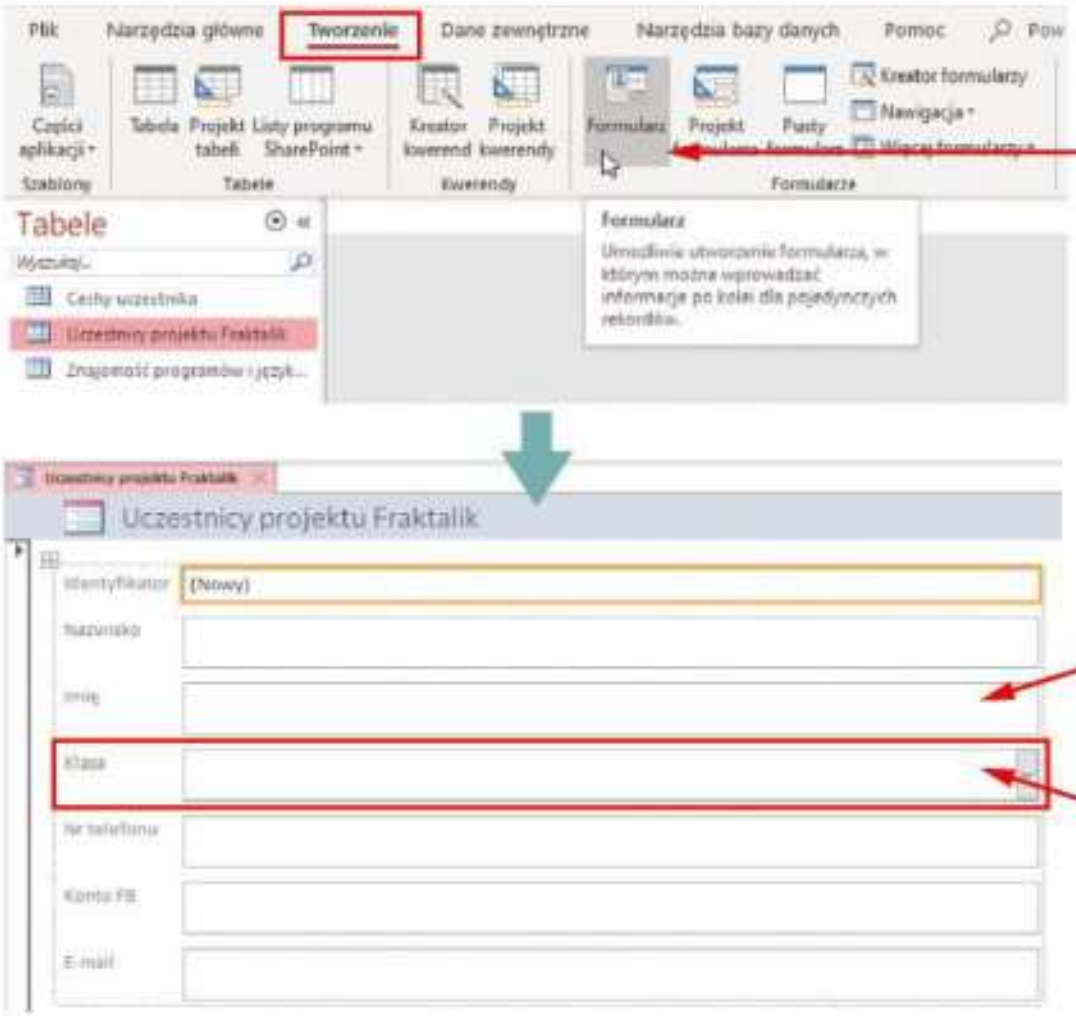
Nasze tabele są gotowe do wprowadzania danych. Możesz je wpisywać bezpośrednio w pola rekordów. Access oferuje narzędzie ułatwiające tę czynność – **Formularz**.

Jeśli użyjesz opcji **Pomoc**, znajdziesz następujące definicje formularzy:

◂ formularz

> **Formularz j**est obiektem bazy danych, który pozwala utworzyć interfejs użytkownika dla aplikacji bazy danych. **Formularz „powiązany"** to formularz bezpośrednio połączony ze źródłem danych, na przykład tabelą lub zapytaniem. Umożliwia on wprowadzanie, edytowanie i wyświetlanie danych w tym źródle danych. Można także utworzyć **formularz „niepowiązany"**, który nie jest bezpośrednio połączony ze źródłem danych, ale zawiera przyciski poleceń, etykiety i inne kontrolki potrzebne do obsługi aplikacji.
>
> źródło: Microsoft, https://support.office.com/pl-pl/article/wprowadzenie-do-formularzy-e8d47343-c937-44e8-a80f-b6a83a1fa3ae

Utwórzmy formularz dla każdej z trzech tabel (rys. 16.6.).



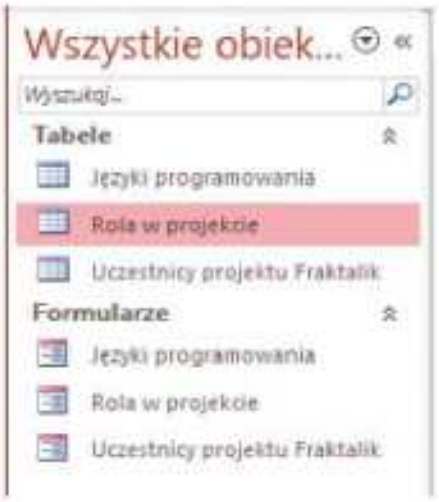
Wybierz tabelę, dla której ma powstać formularz, i kliknij przycisk narzędzia **Formularz** ze wstęgi **Tworzenie**.

Nazwy pól zostaną przeniesione automatycznie z projektu tabeli. Możesz zmieniać wielkość miejsca dla danych, a także usuwać pola.

Pole, dla którego został użyty **Kreator odnośników**, stanie się automatycznie **Polem kombi**.

Rys. 16.6. Przykład tworzenia formularza dla istniejącej tabeli

Podobne formularze utwórz dla pozostałych tabel (rys. 16.7.).



Rys. 16.7. Wszystkie dotychczas utworzone obiekty bazy danych Fraktalik

Podstawowe elementy bazy są zatem gotowe. Przed tobą faza tworzenia elementów umożliwiających dotarcie do danych bazy.

ZADANIA DO WYKONANIA

1. Zmodyfikuj tabelę, którą stworzyłeś/stworzyłaś podczas ćwiczeń. Uzupełnij ją o pola, które według ciebie będą miały znaczenie dla użytkownika bazy. Zmodyfikuj także formularz do wprowadzania danych tak, by uwzględnial te dane.
2. Zmodyfikuj wygląd formularza. Zmień jego kolor i rozmieszczenie pól oraz wielkość i rodzaj czcionki. Wypróbuj wiele ustawień i wybierz najwygodniejsze dla wprowadzania danych. Uzasadnij krótko swoją decyzję.
3. Zaprojektuj tabelę dla kolekcji filmów. Uwzględnij najważniejsze dane, takie jak tytuł, nazwisko reżysera, czas trwania itp.
4*. Utwórz bazę kolekcji plików dźwiękowych. Uwzględnij ścieżkę dostępu do nich, tytuł, wykonawcę i rok powstania utworu.

PODSUMOWANIE LEKCJI

s. 98 • Przed przystąpieniem do tworzenia bazy danych należy sporządzić jej **projekt** i ustalić podstawowe parametry, np. liczbę tabel i kolumn w tabelach oraz rodzaj danych wprowadzanych do pól.

s. 98 • W razie potrzeby skorzystaj z opcji **pomocy** lub publikacji na temat danego systemu baz.

s. 98 • W bazie danych wszystkie informacje są zapisane w **tabelach**.

s. 100 • Systemy baz danych, np. MS Access, LibreOffice Base lub OpenOffice.org Base, wspomagają tworzenie elementów bazy poprzez **kreatory**. Warto z nich korzystać.

s. 101 • Tabele można wypełniać danymi bezpośrednio lub za pośrednictwem **formularzy**.

17. Relacje i pytania, czyli jak uczynić bazę użyteczną

NA TEJ LEKCJI:

- dowiesz się, czym są kwerendy i relacje między polami tabel;
- ułożysz relacje w bazie danych.

Pierwsze kroki z bazami danych już za tobą. Wiesz już, jak tworzyć tabele, czyli miejsca przechowywania danych w bazie, oraz jak ułatwić ich wprowadzanie za pomocą formularzy. Baza danych, choćby zawierała najcenniejsze dane, jest mało przydatna, jeśli nie możemy pozyskać z niej odpowiednich informacji. Stwórzmy więc takie możliwości.

17.1. Relacje, czyli poznajemy sposoby powiązania danych z różnych tabel

Niekiedy w bazie w poszczególnych tabelach występują te same dane, np. wraz z danymi osobowymi pracownika jest przechowywany numer jego telefonu, mimo że w osobnej tabeli zebrano już numery wszystkich osób zatrudnionych w firmie.

Aby uniknąć powielania tych samych informacji w różnych tabelach, stosuje się **relacje** pozwalające na powiązanie niektórych danych. Przykładem mogą być tabele, w których w jednej występują dane pracowników wraz z numerami telefonów, a w drugiej znajduje się spis numerów telefonów. W pierwszej można zrezygnować z pola z numerami i dane te czerpać z drugiej. Warunkiem będzie oczywiście utworzenie relacji.

Relacja to połączenie logiczne pomiędzy dwiema tabelami określające, które pola są wspólne dla tych tabel. Unika się w ten sposób powielania w różnych tabelach tych samych danych, czyli ich **nadmiarowości**.

◂ relacje

W bazie Access występują trzy typy relacji pomiędzy tabelami.

Relacja jeden-do-wielu wystąpi, gdy np. jeden język programowania zna wielu uczestników projektu.

◂ relacja jeden-do-wielu

> Aby w projekcie bazy danych utworzyć relację jeden-do-wielu, należy klucz podstawowy znajdujący się po stronie „jeden" relacji dodać jako pole lub pola do tabeli po stronie „wiele" tej relacji.
>
> źródło: Microsoft, https://support.office.com/pl-pl/article/przewodnik-po-relacjach-pomi%C4%99dzy-tabelami-30446197-4fbe-457b-b992-2f6fb812b58f

W naszym przykładzie może to być połączenie relacją pola **Nazwisko** z tabeli **Uczestnicy projektu Fraktalik** z tabelą **Rola w projekcie**.

relacja wiele-do-wielu

Przykładem **relacji wiele-do-wielu** mogą być powiązania bazy towarów z zamówieniami – jedno zamówienie może przecież dotyczyć wielu towarów, a dany towar może się znaleźć w wielu różnych zamówieniach. Aby utworzyć **relację wiele-do-wielu**, należy utworzyć trzecią tabelę, która rozkłada relację wiele-do-wielu na dwie relacje typu jeden-do-wielu. Zawiera ona klucze podstawowe z obu tabel z danymi.

relacja jeden-do-jednego

> W **relacji jeden-do-jednego** z każdym rekordem w pierwszej tabeli może być związany tylko jeden pasujący rekord w drugiej tabeli, a z każdym rekordem w drugiej tabeli może być związany tylko jeden pasujący rekord w pierwszej tabeli.
>
> źródło: Microsoft, https://support.office.com/pl-pl/article/przewodnik-po-relacjach-pomi%C4%99dzy-tabelami-30446197-4fbe-457b-b992-2f6fb812b58f

Ta relacja jest stosowana najczęściej w przypadku, gdy niektóre dane, które mogły występować w jednej tabeli, należy od niej oddzielić, na przykład z przyczyn bezpieczeństwa lub ponieważ tabela zawiera zbyt wiele pól. Taka relacja wymaga, aby w obu tabelach było używane wspólne pole.

17.2. Narzędzie, czyli jak tworzyć relacje w bazie Access

Utworzenie relacji w bazie Access jest łatwe, ale należy przestrzegać pewnych reguł. W każdej z tabel, które mają być powiązane (mają być utworzone relacje), umieszcza się pola wspólne. To one gwarantują poprawność relacji. Jednak tylko w jednej z tabel wpisuje się ich dane. W pozostałych są one pobierane w ramach powiązania. Kolejność działań procesu tworzenia relacji przedstawia rysunek 17.1. Należy także zachować zgodność typów danych w powiązanych polach.

typy danych w bazie Access

Typy danych w Access określają rodzaj informacji, jakie są przechowywane w bazie, np. czy to będzie tekst, czy liczba, i określają wielkość pamięci potrzebnej do ich zapamiętania. Stosowanie typów danych pozwala na rezerwowanie w pamięci optymalnej wielkości miejsca ich przechowywania, np. **Liczba** – to 1, 2, 4, 8 lub 16 bajtów. Przy deklarowaniu typu **Duża liczba** ograniczamy się do 8 bajtów. Typ **Krótki tekst** pozwala wprowadzić niemal każdy znak (literę, symbol lub liczbę), dlatego jest stosowany najczęściej. Jednak przemyślany dobór typów danych w bazie Access pozwala na wykorzystanie większej liczby funkcji bazy, a także zwiększa dokładność przeszukiwania.

Spis i parametry typów danych w Access znajdziesz w opcji pomocy tego programu.

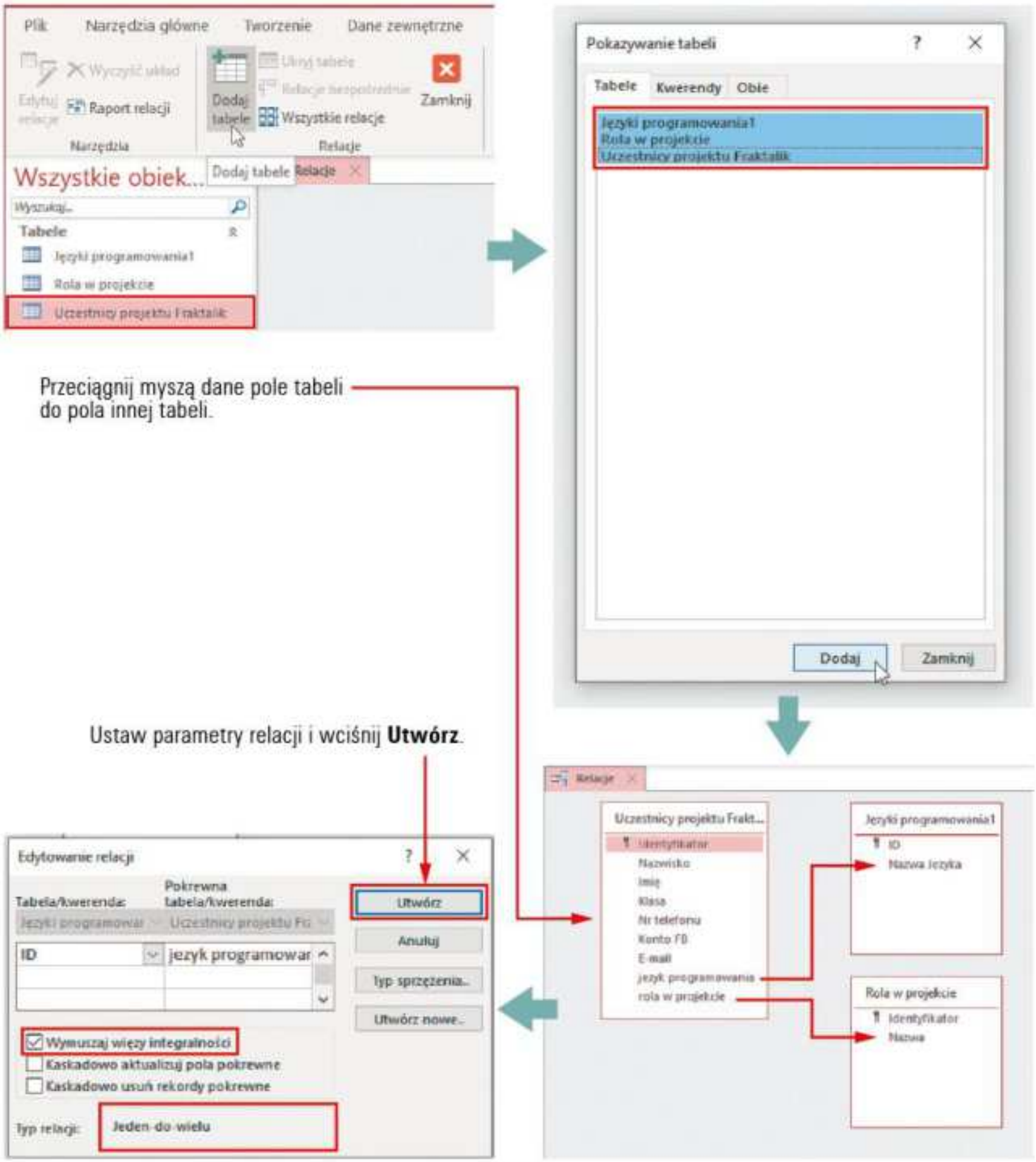


Rys. 17.1. Dodatkowe pola w tabelach bazy Fraktalik

Do tworzenia relacji w Access służy narzędzie **Relacje** (rys.17.2.). Ponieważ użyjemy relacji **jeden do wielu** wybierz ten typ w oknie dialogowym. Wymuś także **więzy integralności**. Aby uniknąć błędów, precyzyjnie operuj kursorem myszy podczas przesuwania nazwy pola tabeli.

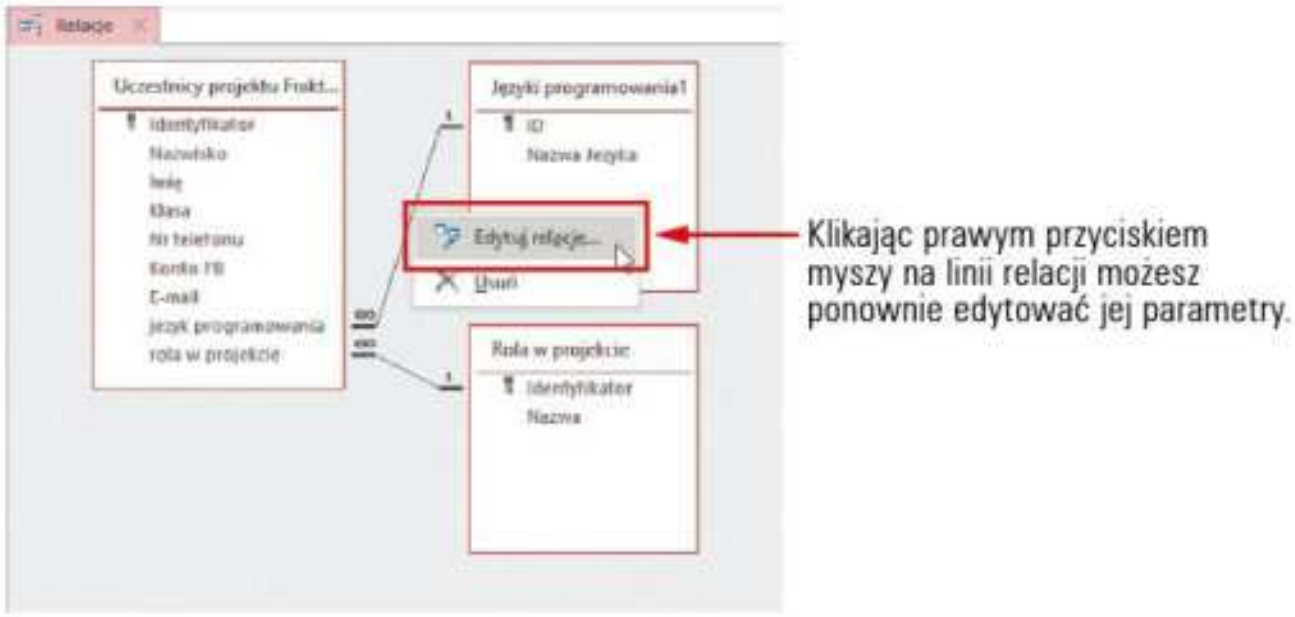
Więzy integralności służą do zapobiegania powstawaniu rekordów odwołujących się do rekordów, które nie istnieją. Jeśli więzy integralności są wymuszane, Access odrzuca takie operacje.

◂ więzy integralności



Rys. 17.2. Tworzenie relacji pomiędzy tabelami za pomocą okna **Relacje**

Utworzone relacje będą widoczne w oknie **Relacje** (rys. 17.3.).



Rys. 17.3. Relacje widoczne w oknie **Relacje**

ZADANIA DO WYKONANIA

1. Ułóż algorytm opisujący wykorzystanie Kreatora odnośników do łączenia tabel relacjami w celu ułatwienia wypełniania pól w Access. Zapisz go w postaci schematu blokowego. Wykorzystaj jeden z darmowych edytorów schematów blokowych, np. DiagramDesigner.
2. Ułóż algorytm tworzenia relacji między tabelami w bazie Access. Zapisz go w postaci listy kroków.
3. Stwórz prezentację, w której na przykładach opiszesz funkcje klucza podstawowego i klucza obcego.

4*. Utwórz bazę danych w programie Access wzorowaną na spisie kontaktów z twojego telefonu. Uwzględnij wszystkie pola spisu. Przygotuj dwie tabele – jedną do gromadzenia nazwisk, a drugą z numerami telefonów i innymi danymi. Utwórz relacje umożliwiające łatwe wpisywanie numerów odpowiadających kontaktom.

PODSUMOWANIE LEKCJI

s. 103 • Każdy **obiekt bazy** po edycji powinien być zapisany i zamknięty.

s. 103 • **Relacja** to połączenie logiczne pomiędzy dwiema tabelami określające, które pola są wspólne dla tych tabel.

s. 104 • W bazie Access są możliwe trzy rodzaje relacji – **jeden-do-wielu**, **wiele-do-wielu** i **jeden-do-jednego**.

s. 104 • Do tworzenia relacji w bazie Access służy narzędzie **Relacje**.

s. 105 • **Więzy integralności** służą do zapobiegania powstawaniu rekordów odwołujących się do rekordów, które nie istnieją.

s. 107 • **Typy danych** w Access określają rodzaj informacji, jakie są przechowywane w bazie, np. czy to będzie tekst, czy liczba, i określają wielkość pamięci potrzebnej do ich zapamiętania.

18. Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access

NA TEJ LEKCJI:

- dowiesz się, czym są kwerendy i raporty oraz nauczysz się ich używać;
- nauczysz się używać filtrowania odczytanych danych.

Po kilku godzinach spędzonych z bazą Access wiesz już, że to bardzo rozbudowany program. Ćwiczenia, które do tej pory wykonaliśmy, nie uwzględniały wielu opcji użytych narzędzi. Jeśli chcesz zgłębić tajniki bazy Access, sięgnij do tutoriali, opcji **Pomoc** i książek obszernie opisujących wszystkie jej możliwości. W tym rozdziale zostaną omówione narzędzia i mechanizmy pozwalające na wygodne z niej korzystanie.

18.1. Kwerendy, czyli odczytujemy dane z bazy Access

◂ kwerenda

Kwerendy stosuje się dla ułatwienia wyświetlania, zmiany danych oraz ich dodawania i usuwania z bazy. Podczas wyszukiwania informacji w bazie Access z zastosowaniem kwerend można stosować filtrowanie według założonych kryteriów. W naszej bazie istnieje wiele typów kwerend. Na początku użyjemy **wybierającej**, by w jednej tabeli wyjściowej zgromadzić interesujące nas dane. Zapewne zastanawiasz się, jaki jest sens ich stosowania, skoro i tak można wyświetlić zawartość tabel. Otóż w każdej z tabel gromadzimy często wiele informacji. Nie zawsze potrzebujemy wszystkich – np. gdy nauczyciel wpisuje ocenę do elektronicznego dziennika (rozbudowanej bazy danych), nie musi odczytywać wszystkich danych ucznia (daty urodzenia, adresu itp.). Wystarczy pozycja na liście lub nazwisko z imieniem. Ponadto odczytywanie danych z wielu tabel jest niewygodne i może być źródłem pomyłek. Kwerenda pozwoli na „zadanie bazie pytania" o konkretne informacje z różnych tabel. Odpowiedź otrzymamy w jednej tabeli wyjściowej. Zyskamy także możliwość **filtrowania**, które jest jednym z kryteriów kwerend.

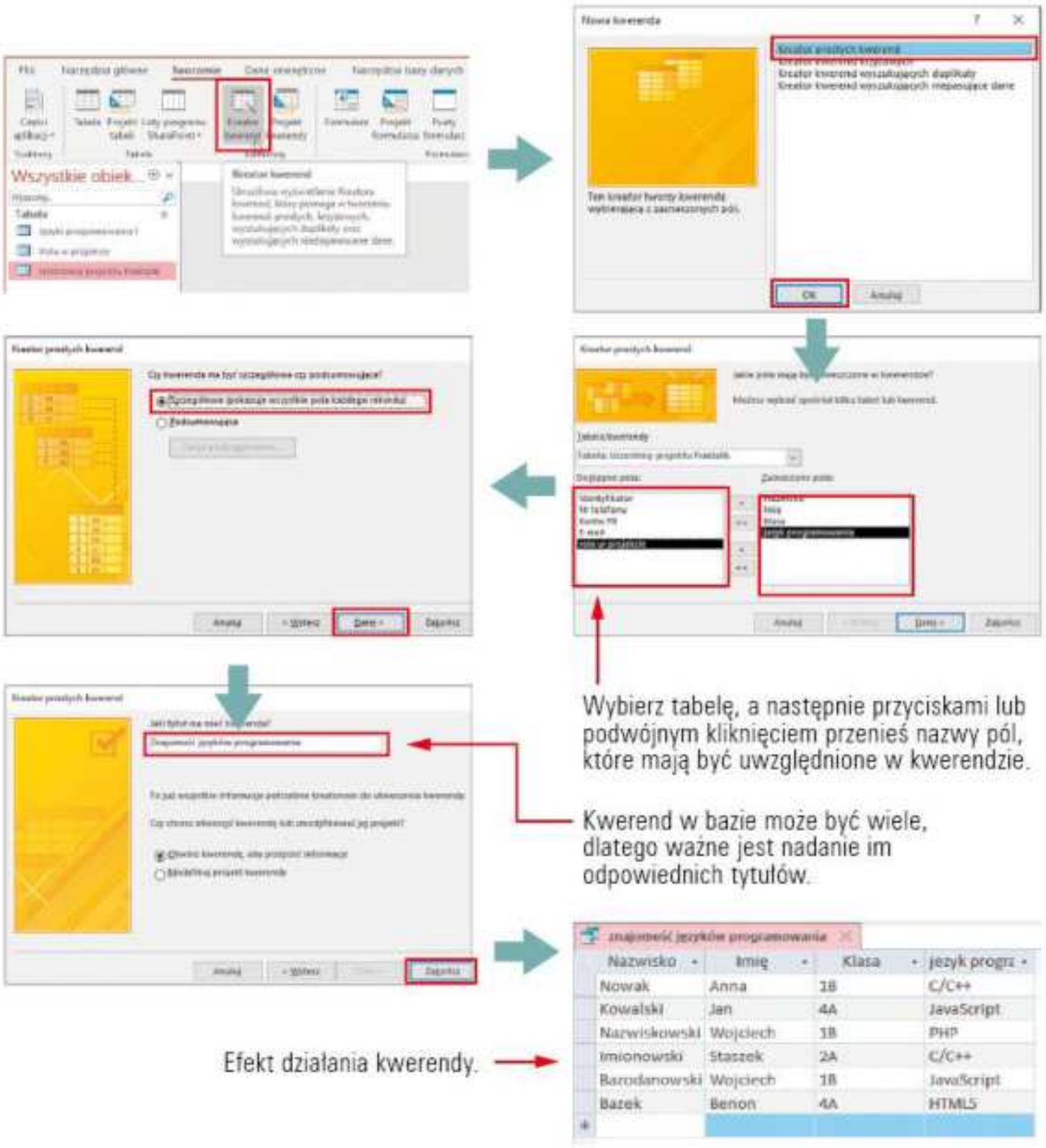
◂ filtrowanie

Filtrowanie jest jednym z kryteriów zapytania (kwerendy) stawianemu bazie. Może mieć postać formuły ułożonej z użyciem operatorów, np. <, >, =, lub wyrażeń, z którymi będą porównywane dane z pól tabel.

Użyjmy bazy Fraktalik. Załóżmy, że interesuje nas informacja w jakich językach programują członkowie zespołu projektowego. Kwerenda może być zbudowana za pomocą kreatora (rys. 18.1.). Oczywiście użyte w kwerendzie tabele są ze sobą powiązane relacjami.

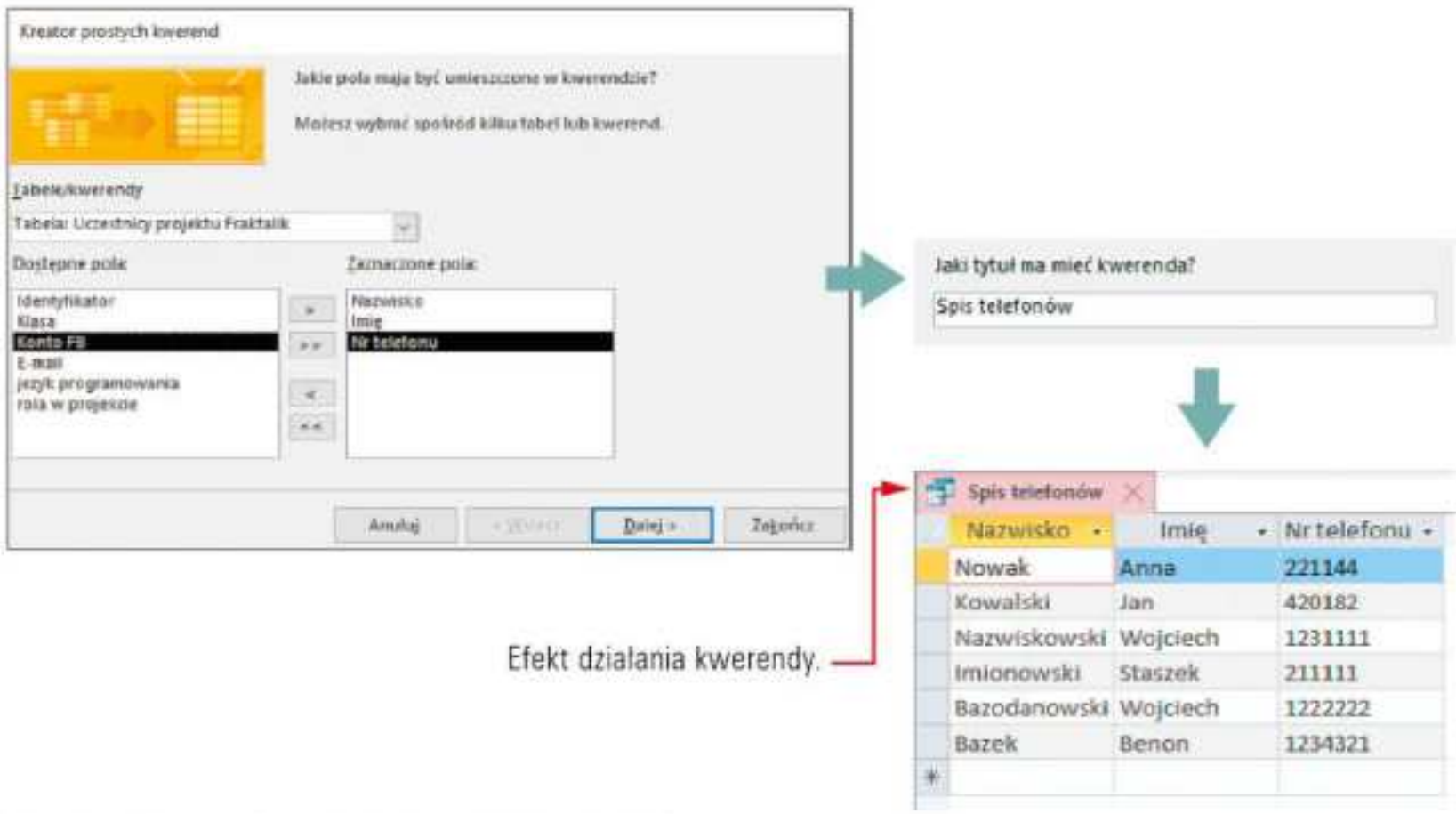
Uwaga!

Podczas tworzenia kolejnych istniejące kwerendy znajdą się w spisie dostępnych tabel – ich pola mogą zostać w nich użyte.



Rys. 18.1. Tworzenie kwerendy prostej w bazie Access za pomocą kreatora

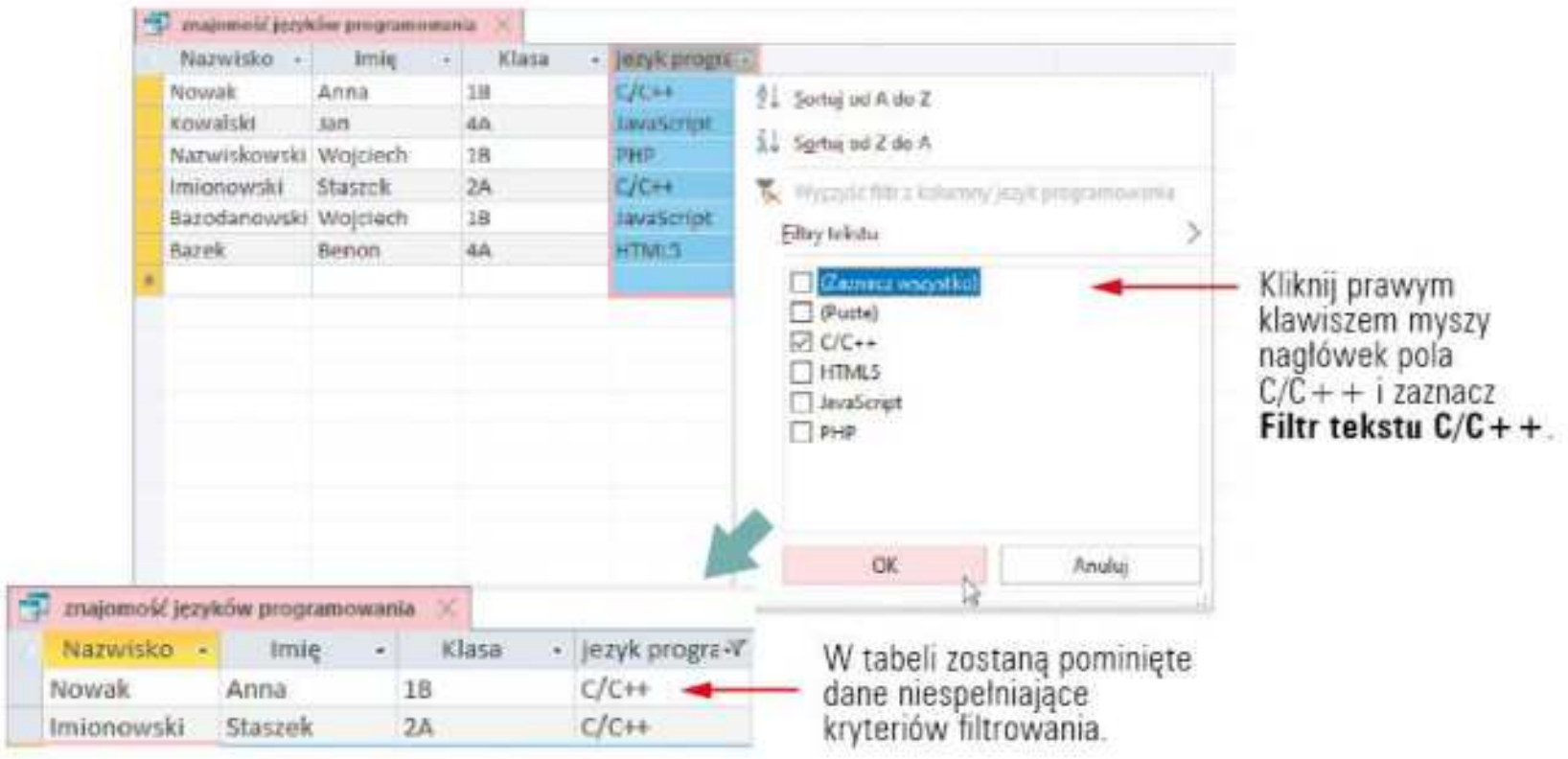
W naszej bazie możesz utworzyć wiele kwerend – także takich, które wyświetlają dane z jednej tabeli. Przykładem może być kwerenda **Spis telefonów** (rys. 18.2.).



Rys. 18.2. Kwerenda zbudowana z pól jednej tabeli

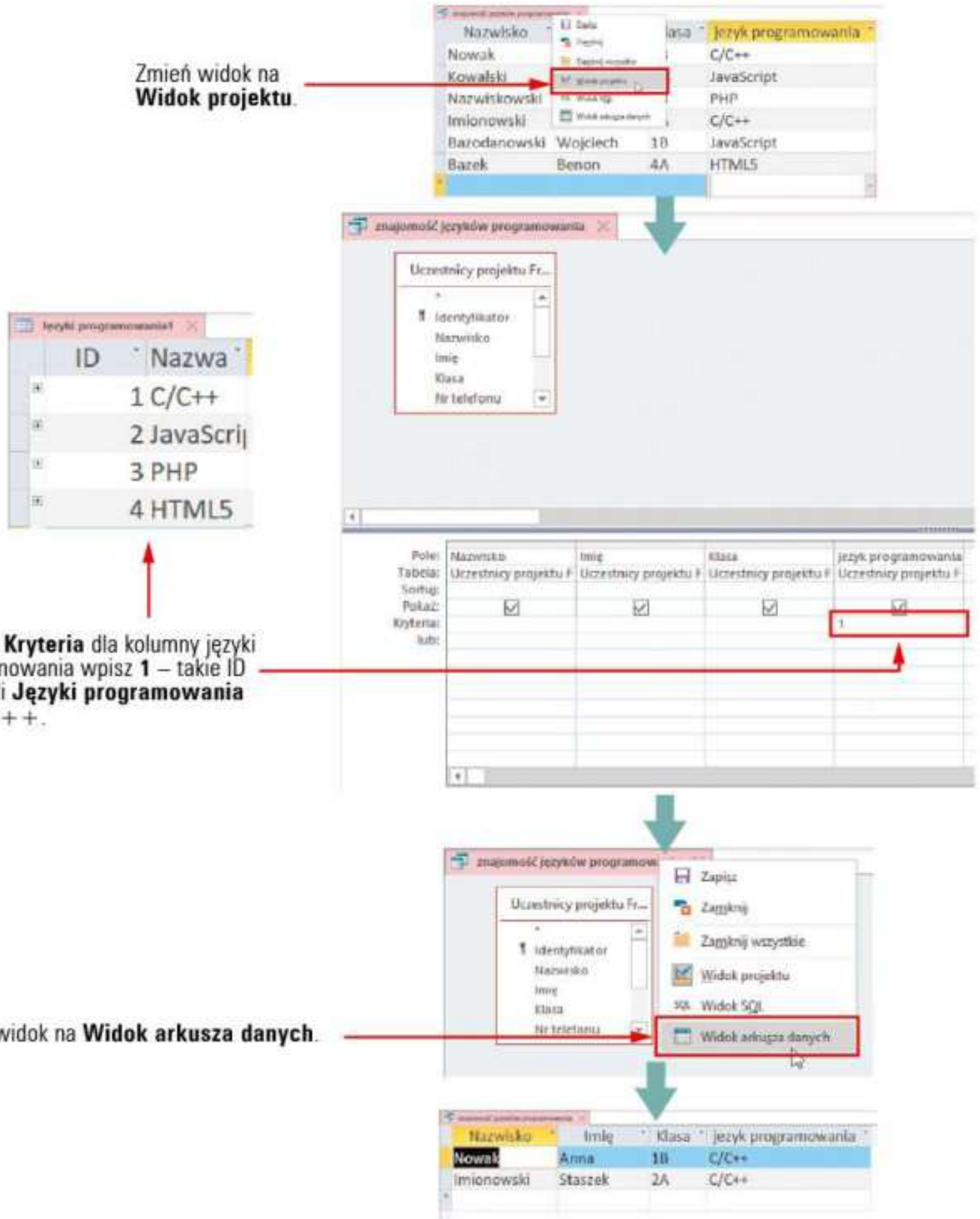
18.2. Nie wszystko nas interesuje, czyli filtrowanie informacji

W tabeli, którą zestawiła nam kwerenda (rys. 18.1.), otrzymaliśmy informacje o tym, jakie języki programowania znają poszczególni członkowie zespołu. Użyjmy filtrowania, by dowiedzieć się, który z nich zna język C lub C++ (rys.18.3.).



Rys. 18.3. Filtrowanie danych za pomocą filtru tekstu dla wybranego pola

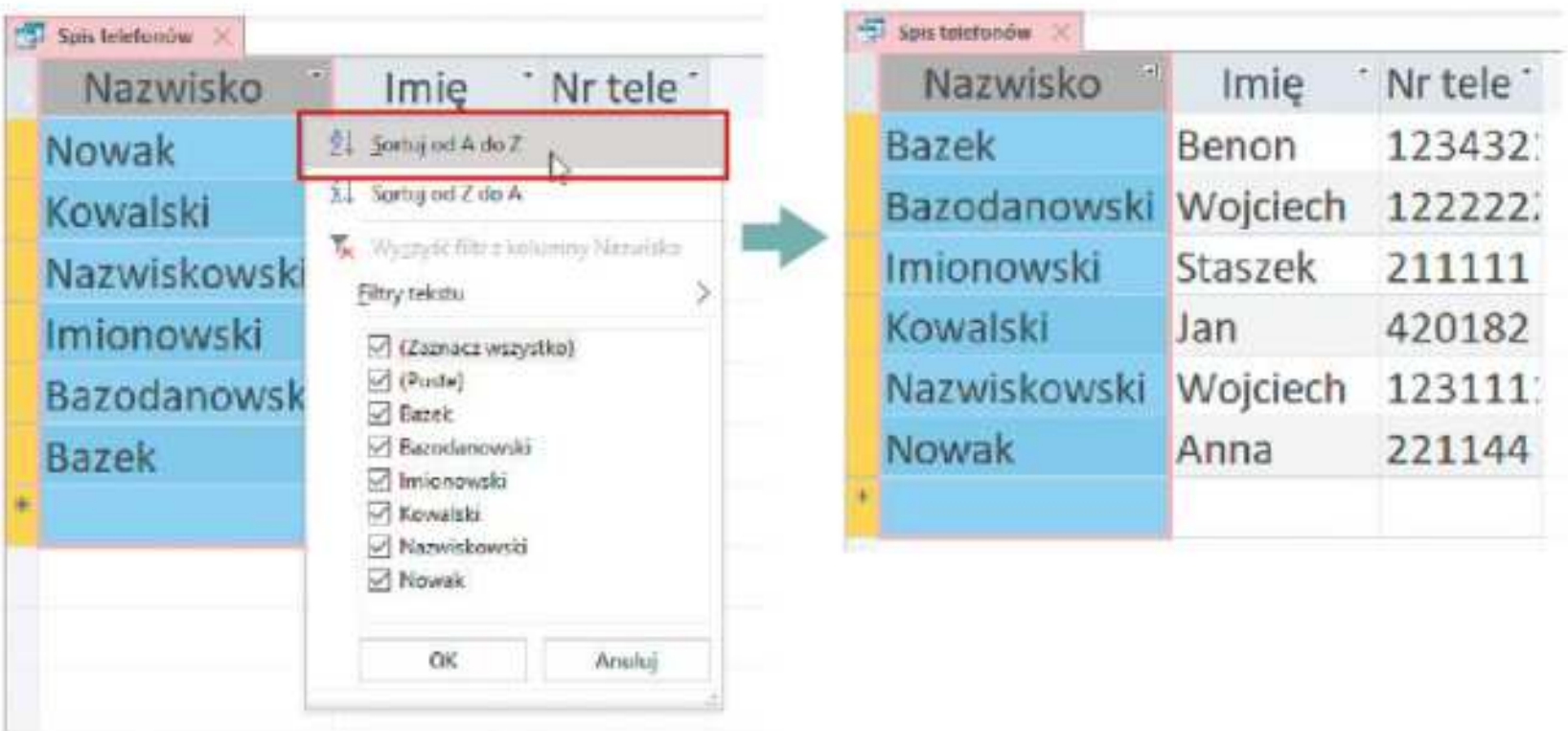
Filtrowanie za pomocą wybierania kryteriów z listy jest skuteczne, ale wymaga wykonania kilku czynności. Zmodyfikujmy kwerendę tak, by wyświetlała jedynie dane uczestników umiejących programować w C lub C++ (rys. 18.4.). Po zmianie efekt powinien być identyczny jak na rysunku 18.3. Co ważne, jest on osiągnięty bez konieczności używania filtrów z listy.



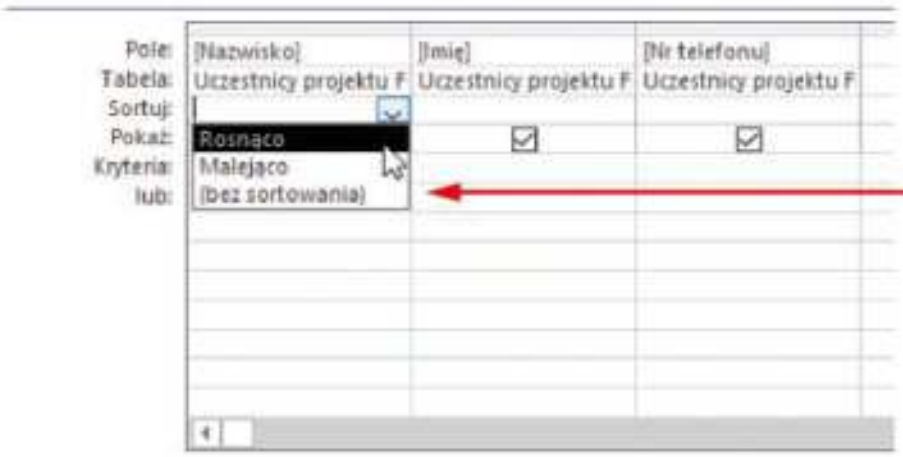
Rys. 18.4. Modyfikacja kwerendy polegająca na dodaniu kryteriów filtrowania

18.3. Zmiana kolejności, czyli sortujemy dane w kwerendzie

Domyślasz się, że sortowanie w tabeli kwerendy można wykonać za pomocą menu (rys. 18.5.) lub uwzględnić w kryteriach (rys. 18.6.). Druga opcja oczywiście nie eliminuje późniejszego użycia menu kontekstowego.



Rys. 18.5. Sortowanie z zastosowaniem menu

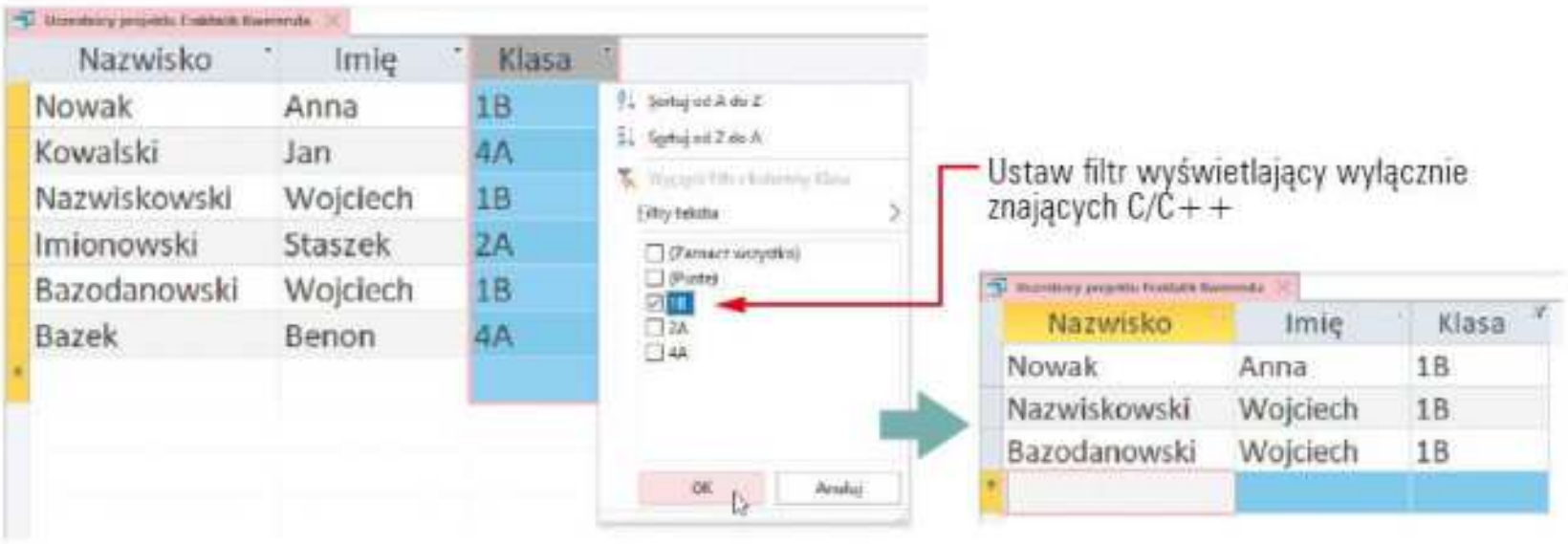


Rys. 18.6. Ustalenie sortowania w projekcie kwerendy

18.4. Raportowanie, czyli drukujemy dane z bazy

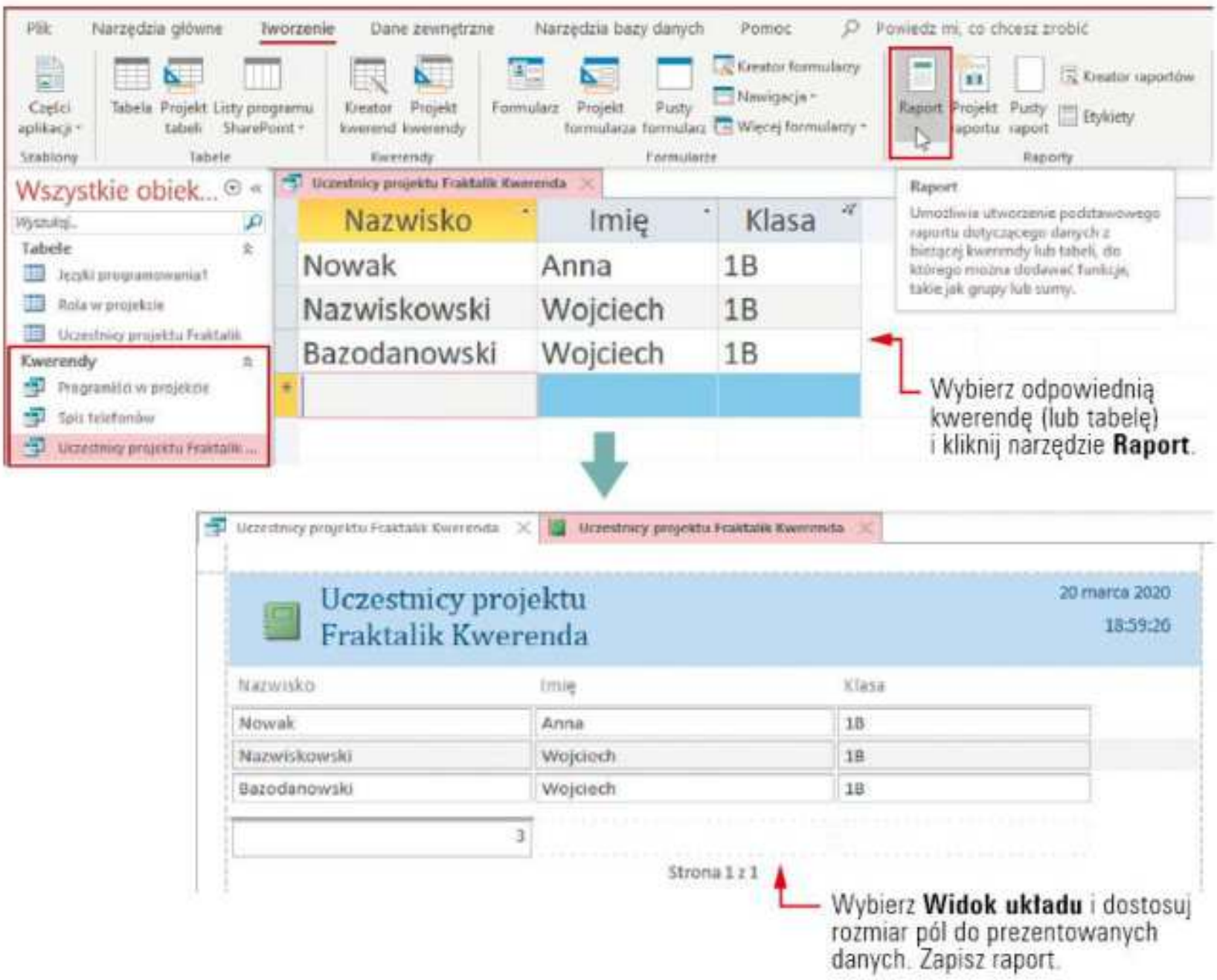
Raport jest obiektem bazy danych przeznaczonym do prezentowania odczytanych z niej informacji. Raporty mogą być powiązane z tabelami lub kwerendami. ◂ raport

Zaprojektuj raport, w którym zostaną wyszczególnione nazwiska i imiona uczestników projektu chodzących do klasy 1B. W tym celu ułóż kwerendę **Uczestnicy projektu Fraktalik** (rys. 18.7.).



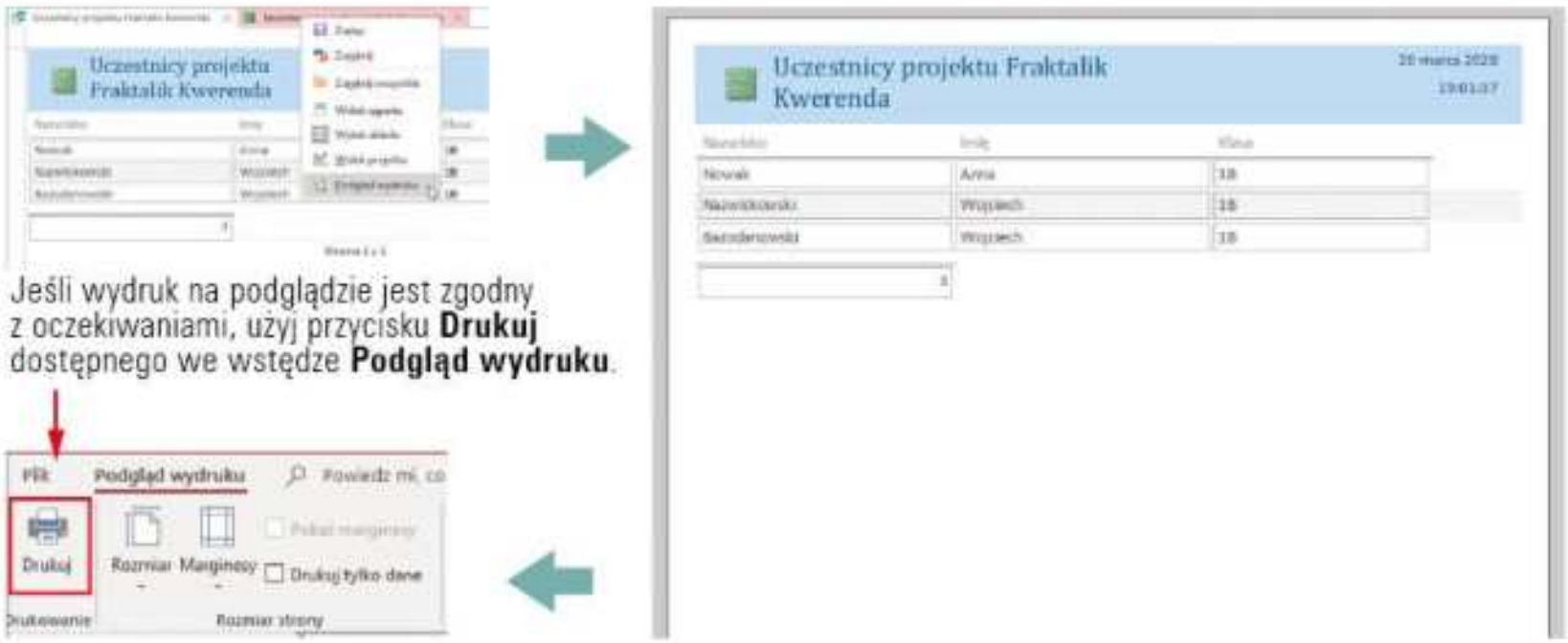
Rys. 18.7. Projekt kwerendy do budowy raportu o założonych parametrach

Gdy kwerenda jest już gotowa, sprawdź poprawność jej działania. Do utworzenia **Raportu** posłuży nam kreator (rys. 18.8.).



Rys. 18.8. Tworzenie raportu za pomocą kreatora

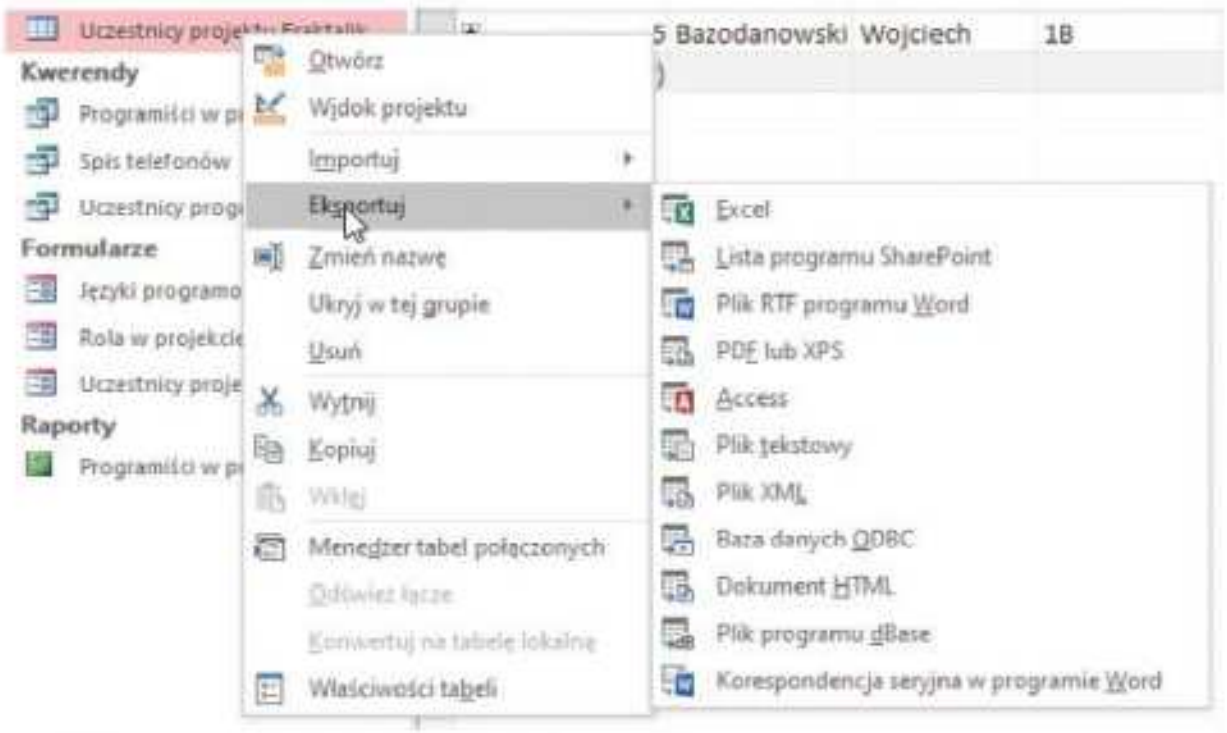
Raporty można modyfikować za pomocą narzędzi ze wstęg **Projektowanie**, **Rozmieszczenie**, **Formatowanie** i **Ustawienia strony**. Ta ostatnia dotyczy parametrów drukowania raportów. Warto z nich skorzystać, gdy baza danych należy do firmy i raporty powinny być opatrzone jej logo lub nazwą. Ostateczny wygląd wydruku raportu możesz sprawdzić, zmieniając widok (rys. 18.9.).



Rys. 18.9. Podgląd wydruku raportu w bazie Access

Bogactwo opcji i narzędzi programu Access pozwala nie tylko na tworzenie funkcjonalnych i sprawnie działających baz, lecz także na atrakcyjne prezentowanie danych i szybkie do nich dotarcie.

Każdą tabelę bazy możesz wyeksportować do innych formatów (rys. 18.10.), np. Excel.



Rys. 18.10. Eksport tabeli do różnych formatów

Podczas pracy z bazą Access warto korzystać z dobrze opracowanych opisów w opcji **Pomoc**.

ZADANIA DO WYKONANIA

1. Utwórz prezentację, w której przedstawisz definicję i przykłady zastosowania kwerend w programie Access. Skorzystaj z tutoriali z opcji **Pomoc**, w tym także wideo. Do wykonania prezentacji użyj dowolnego programu, np. PowerPoint.
2. Dla dowolnej tabeli bazy Access zastosuj różne rodzaje filtrowania. Opisz ich działanie w dokumencie tekstowym. Plik zapisz w formacie PDF. Wykorzystaj zrzuty ekranowe i opcje eksportu tabel w programie Access.
3. Dla tabeli **Uczestnicy projektu Fraktalik** zastosuj różne rodzaje sortowania danych. Wybierz najodpowiedniejszą z punktu widzenia prezentowanych danych. Zastosuj to sortowanie jako domyślne dla tej tabeli.
4. Utwórz raport prezentujący nazwiska, imiona i spis numerów telefonów uczestników będących programistami. Sformatuj odpowiednio pola, by można było wydrukować raport na drukarce A4. Szerokość pól dostosuj do danych w nich wyświetlanych.

5*. Zmodyfikuj raport z zadania 4., dodając wybrane logo, przykładową nazwę firmy i tło. Pozostałe parametry zmień według uznania, by wydruki były atrakcyjne i czytelne. Dokonaj próbnego wydruku.

PODSUMOWANIE LEKCJI

s. 107 • **Kwerendy** stosuje się dla ułatwienia wyświetlania i zmiany danych oraz ich dodawania i usuwania z bazy.

s. 107 • W kwerendach i tabelach można stosować **filtrowanie** lub **sortowanie**. **Filtrowanie** jest jednym z kryteriów zapytania. Może mieć postać formuły ułożonej z użyciem operatorów np. <, >, =, lub wyrażeń, z którymi będą porównywane dane z pól tabel.

s. 111 • **Raport** jest obiektem bazy danych przeznaczonym do prezentowania odczytanych z niej informacji. Raporty mogą być powiązane z tabelami lub kwerendami.

19. Formularze, czyli ułatwiamy sobie wprowadzanie informacji do bazy danych

NA TEJ LEKCJI:

- poznasz definicję formularza;
- nauczysz się tworzyć podstawowe formularze, również te składające się z wielu elementów;
- dowiesz się, jak korzystać z kreatora formularza;
- nauczysz się tworzyć indywidualny formularz od podstaw.

Formularze umożliwiają stworzenie interfejsu użytkownika, dzięki któremu łatwiej jest wprowadzać i edytować dane. Dzięki nim możemy określić sposoby wykorzystania przez użytkowników danych zawartych w bazie. Większość użytkowników baz woli posługiwać się formularzami niż wprowadzać, przeglądać i edytować dane bezpośrednio w tabelach.

19.1. Tworzymy formularz

◂ formularz

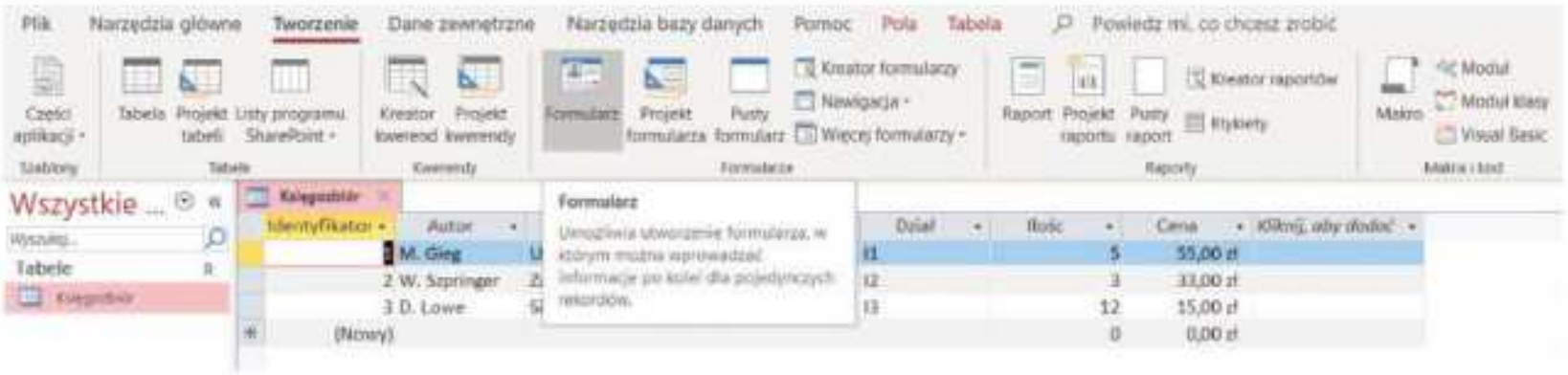
Formularz jest jednym z obiektów bazy danych, który umożliwia wprowadzanie danych oraz wyświetlanie zawartych w niej informacji. Informacjami, które wprowadzamy za pomocą formularza, mogą być dane alfanumeryczne (zawierające tekst zapisany przy użyciu liter i cyfr) przechowywane w tabelach oraz informacje innego typu, jak grafika czy dźwięk, które są przechowywane w plikach poza bazą danych.

Tworzenie formularza przedstawimy na przykładzie bazy danych **Biblioteka**, która będzie miała jedną tabelę o nazwie **Księgozbiór**. Tabela **Księgozbiór** powinna składać się z pól, które przedstawiono w tabeli 19.1.

Nazwa pola	Typ danych	Rozmiar pola	Wymagane
Identyfikator	Autonumerowanie	Liczba całkowita długa	
Autor	Krótki tekst	255	TAK
Tytuł	Krótki tekst	255	TAK
Dział	Krótki tekst	5	TAK
Ilość	Liczba	Liczba całkowita długa	TAK
Cena	Waluta		TAK

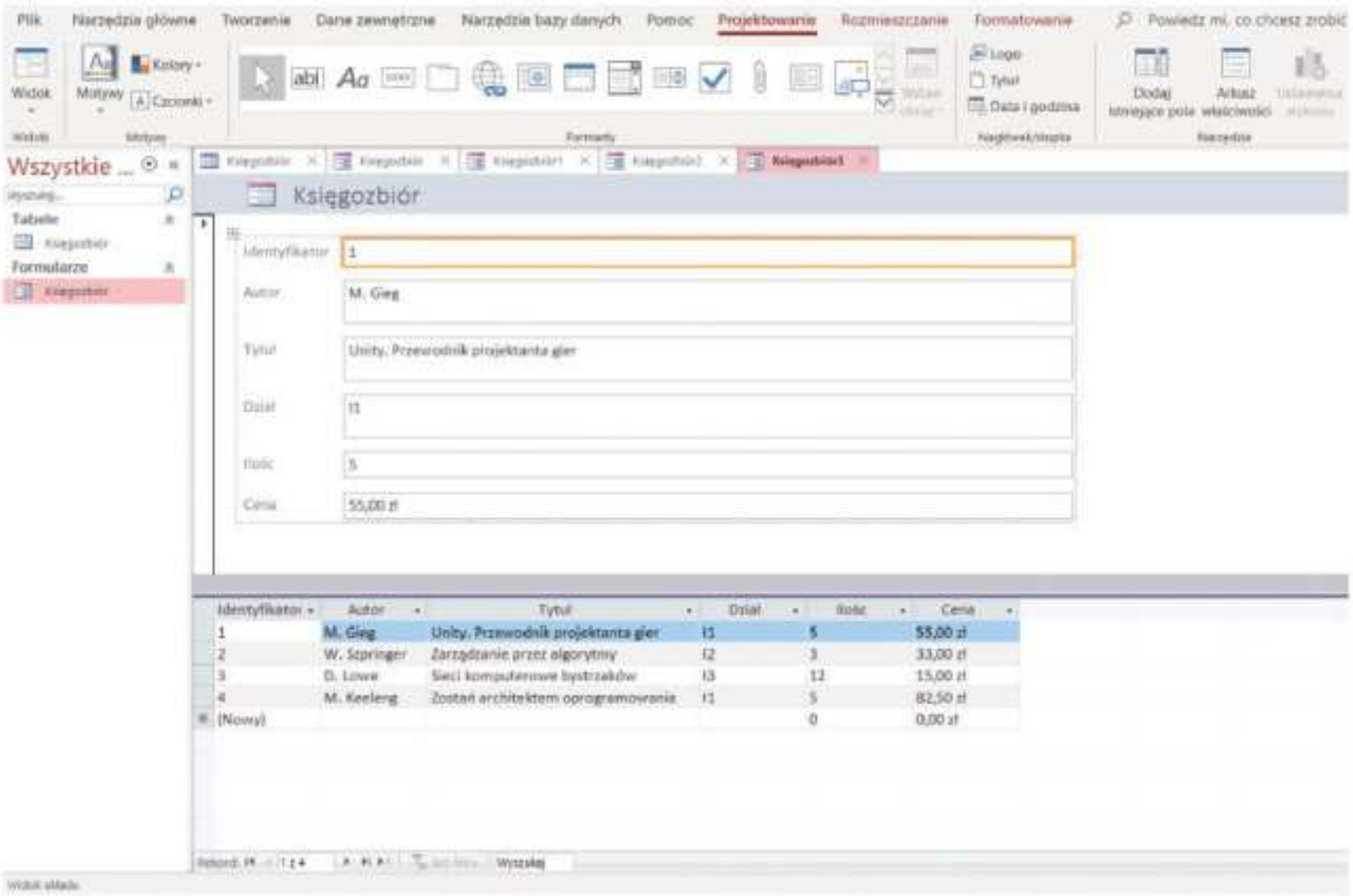
Tab. 19.1. Tabela **Księgozbiór** bazy danych **Biblioteka**

Z karty **Tworzenie** wybieramy **Formularz** (rys. 19.1.).



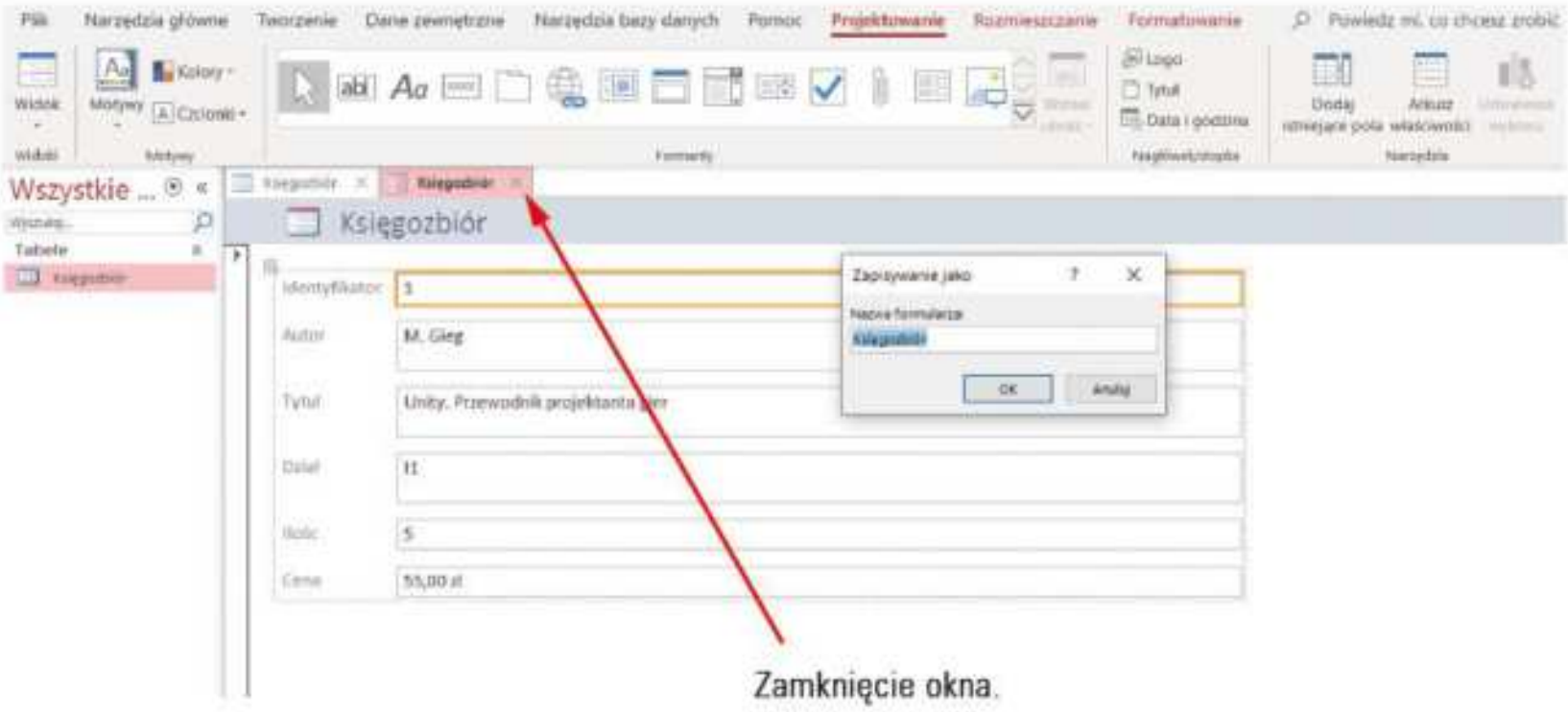
Rys. 19.1. Karta **Tworzenie** – polecenie **Formularz**

Po wybraniu tego polecenia automatycznie otworzy się okno formularza dostosowane do naszej bazy danych. Zawiera ona już pewne informacje, które również wyświetlą się w naszym formularzu (rys. 19.2.).



Rys. 19.2. Okno formularza stworzone na podstawie tabeli **Księgozbiór**

Tak stworzony formularz jest już gotowy do pracy. Aby go zapisać, wystarczy go zamknąć przyciskiem **X**, a następnie kliknąć przycisk **TAK** na oknie dialogowym, które się ukaże. Akceptacja zapisu spowoduje wyświetlenie okna, w którym będziemy mogli nadać nazwę naszemu formularzowi (rys. 19.3.).

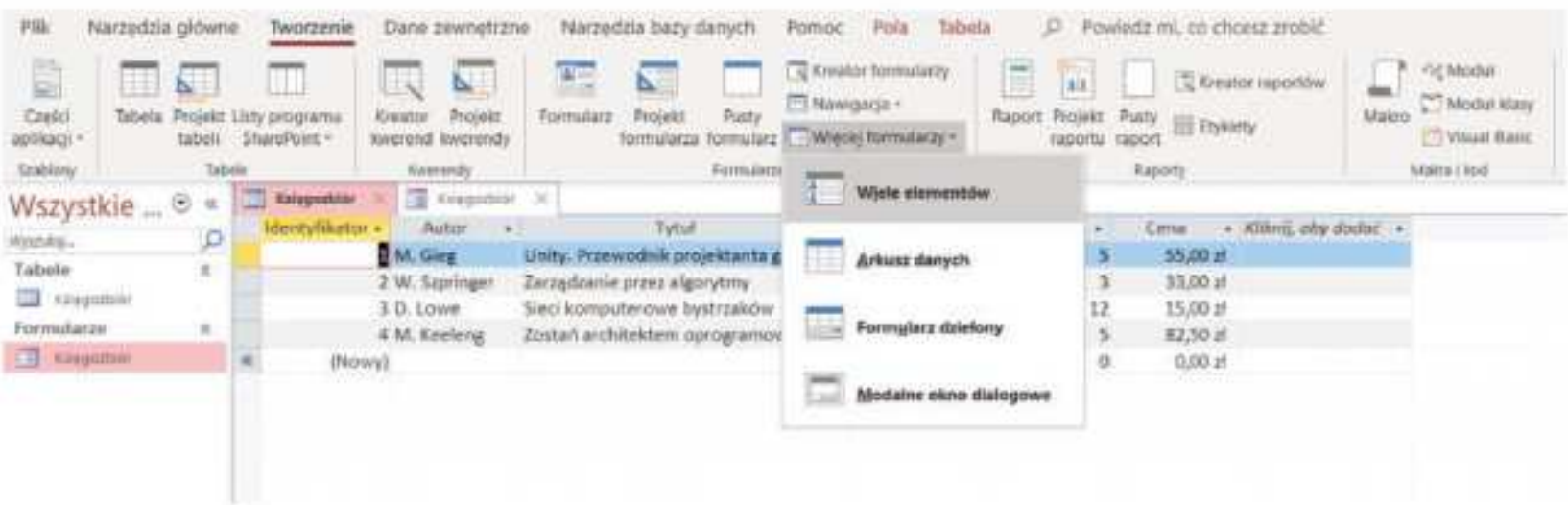


Rys. 19.3. Zapisanie nazwy formularza

19.2. Formularz – wiele elementów

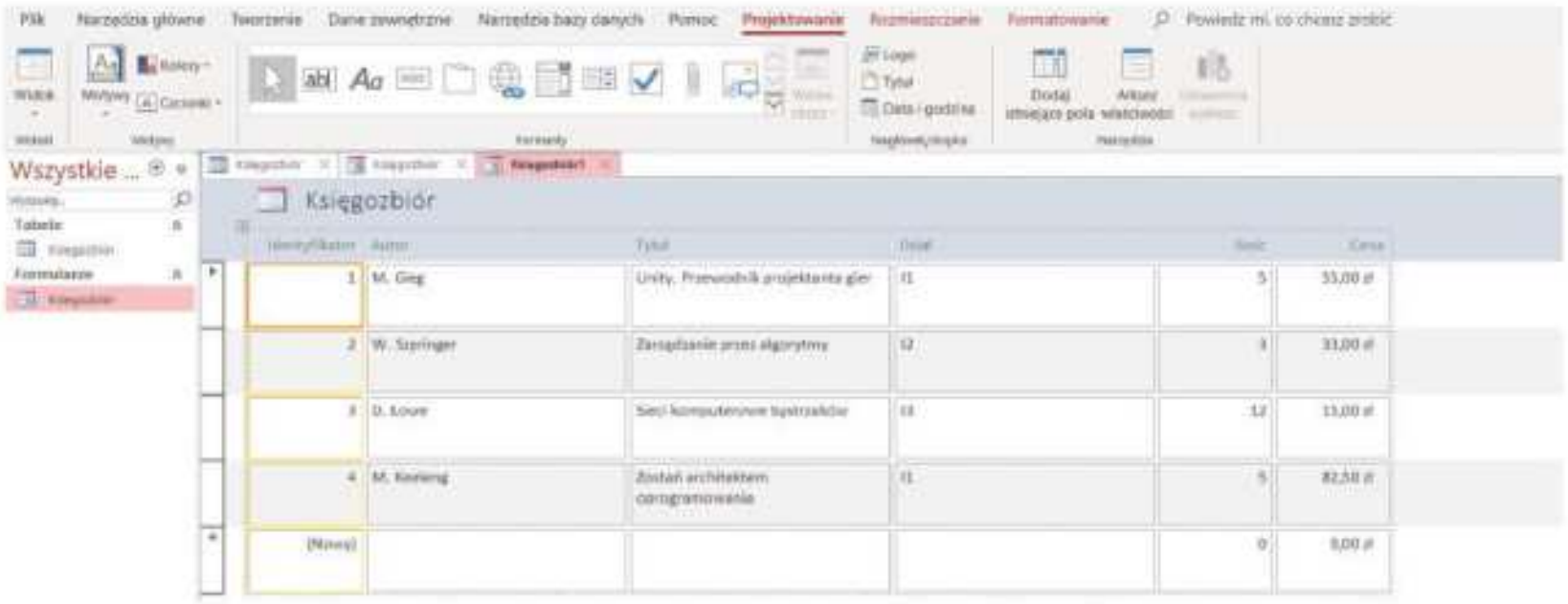
wiele elementów

Formularz z wieloma elementami umożliwia przedstawienie formatki z wieloma rekordami jednocześnie. Poprzedni formularz wyświetlał tylko jeden rekord. Typ formularza **Wiele elementów** to formularz typu tabelarycznego. Aby go utworzyć, należy tak jak poprzednio wybrać kartę **Tworzenie**, ale tym razem z grupy **Formularze** wybieramy **Więcej formularzy** i następnie **Wiele elementów**.



Rys. 19.4. Tworzenie formularza – wiele elementów

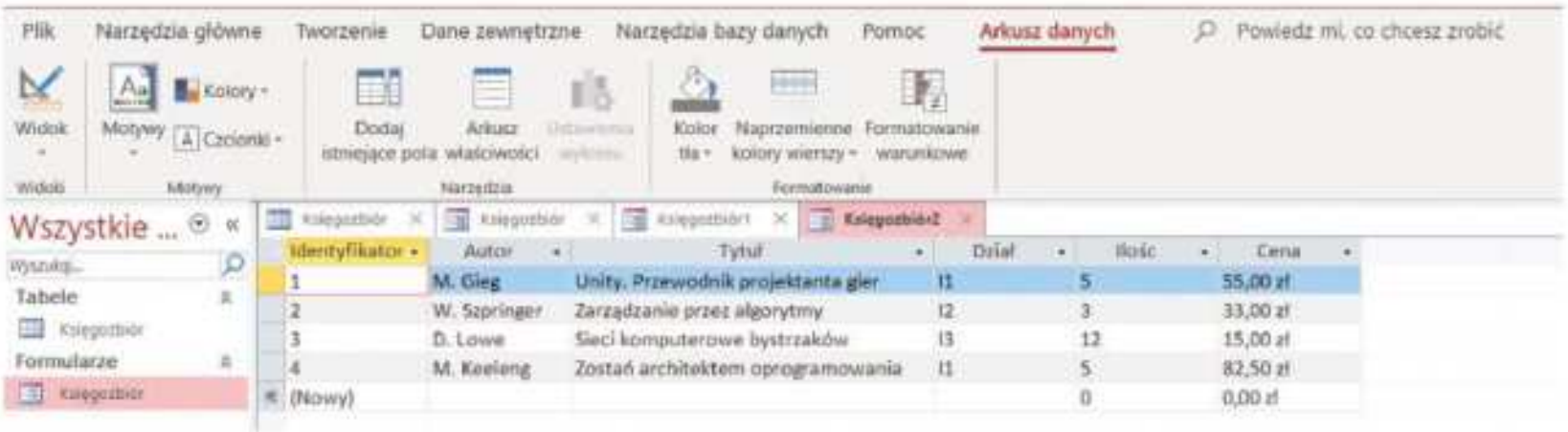
Poruszanie się po formularzu typu tabelarycznego pozwala na bezpośredni dostęp do wyszukiwanych przez nas danych (rys. 19.5.).



Rys. 19.5. Wygląd formularza – wiele elementów

19.3. Formularz – arkusz danych

arkusz danych ▸ **Arkusz danych** umożliwia utworzenie formularza pokazującego wiele rekordów w arkuszu danych, po jednym rekordzie na wiersz – jest on odmianą formularza z wieloma danymi. Aby go uruchomić, klikamy kartę **Tworzenie** i w grupie **Formularze** wybieramy **Więcej formularzy**, a następnie **Arkusze danych** (rys. 19.6.).



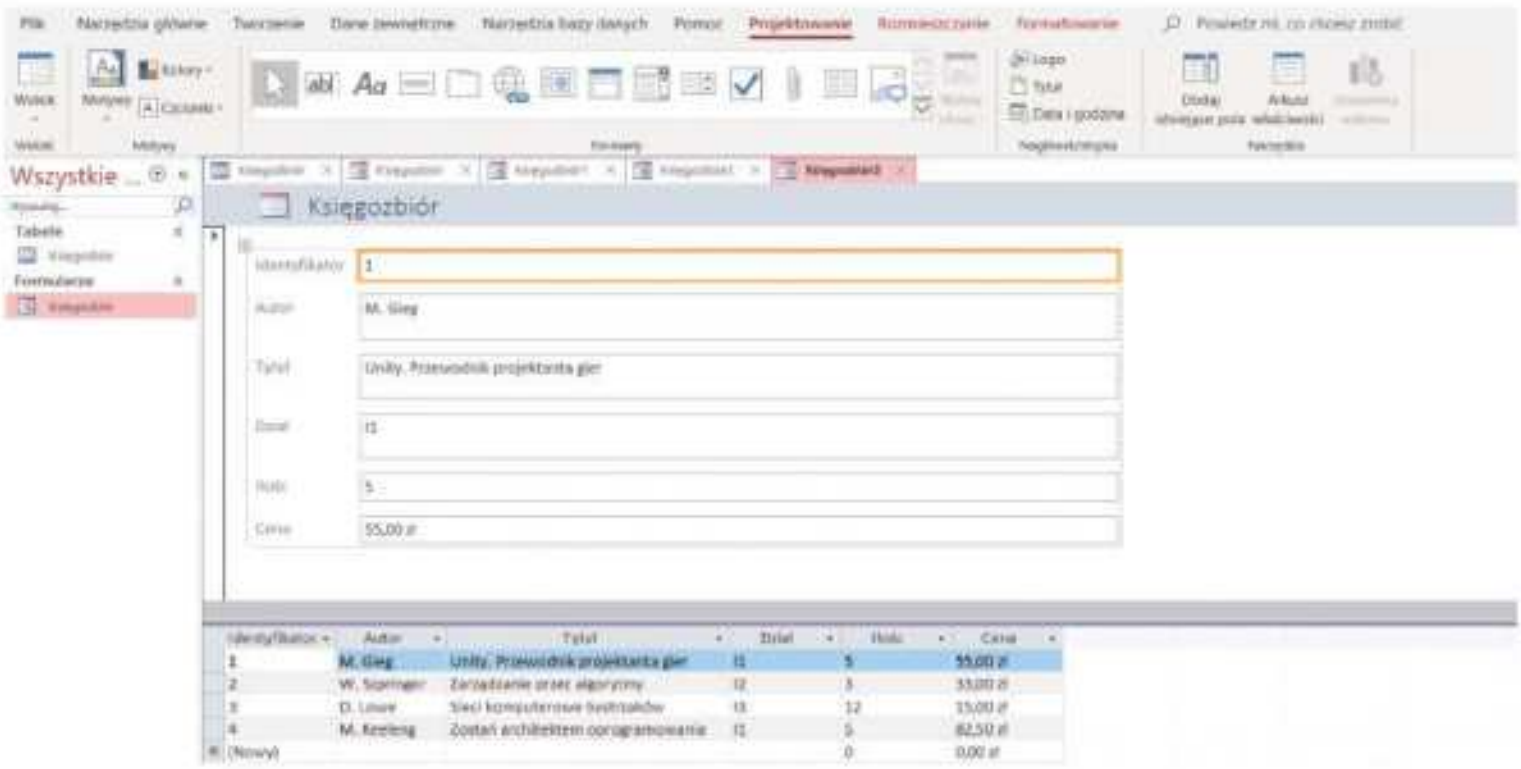
Rys. 19.6. Formularz – arkusz danych

Wygląd tego arkusza prawie niczym nie różni się od wyglądu tabeli w bazie danych. Wszystkie procedury prezentacji danych, jakie występują w tabelach, mają również zastosowanie przy tym formularzu.

19.4. Formularz dzielony

formularz dzielony ▸ **Formularz dzielony** umożliwia utworzenie formularza, w którym w górnej sekcji jest pokazany arkusz danych, a w dolnej sekcji znajduje się formularz do wprowadzania informacji dotyczących rekordu wybranego w arkuszu danych.

Formularz dzielony uruchamiamy jak wcześniejsze, czyli z karty **Tworzenie** i w grupie **Formularze** wybieramy **Więcej formularzy**, a następnie **Formularz dzielony** (rys. 19.7.).



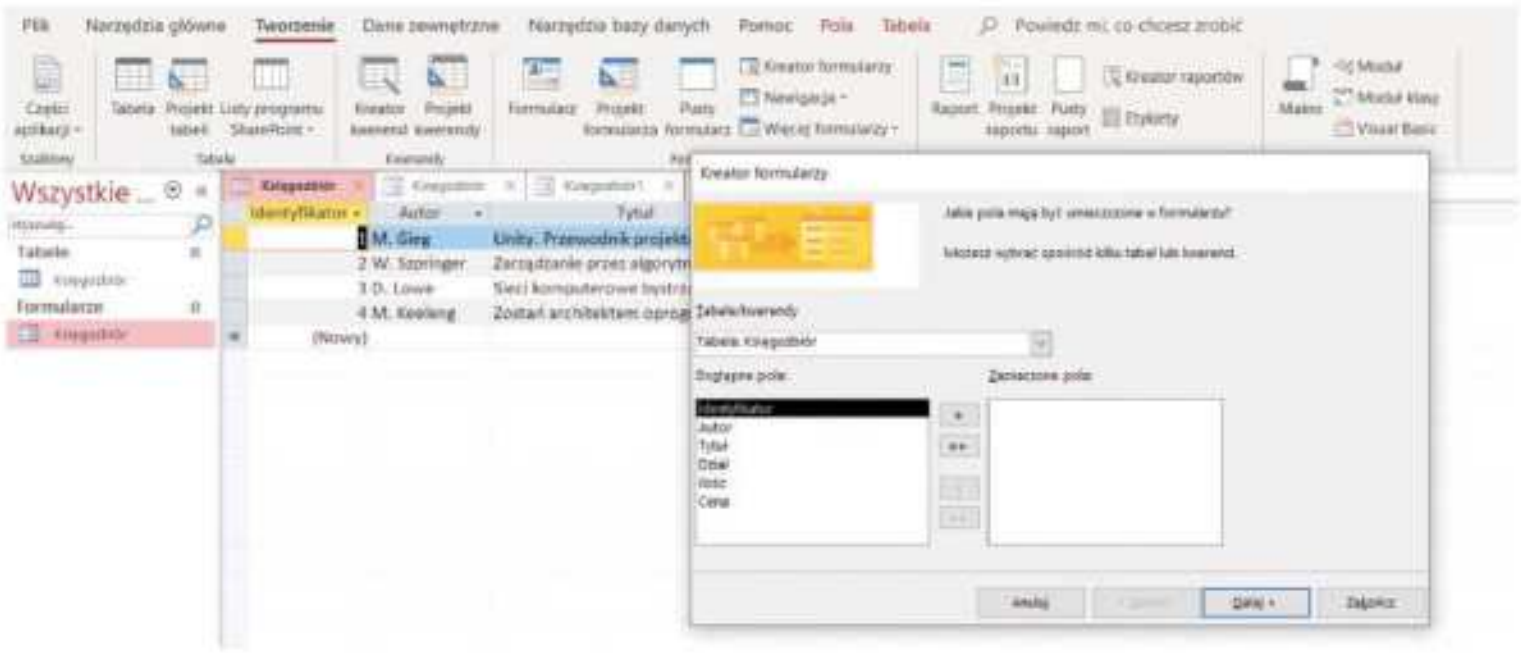
Rys. 19.7. Formularz dzielony

19.5. Kreator formularza

kreator formularza

Tworzenie formularza za pomocą kreatora umożliwia stworzenie prostych formularzy dostosowanych do naszych potrzeb. Tworzymy go w następujących krokach:

1. Na karcie **Tworzenie** w grupie **Formularze** wybieramy **Kreator formularzy** (rys. 19.8.).



Rys. 19.8. Uruchomienie kreatora formularzy

2. Zostanie uruchomione okno, w którym znajdują się następujące pola:

- lista rozwijana, która umożliwia wybór źródła danych (tabele, kwerendy),
- lista wyświetlająca dostępne pola danych w wybranym przez nas źródle danych,
- lista wyświetlająca wybrane pola danych ze źródła danych,
- przycisk przeniesienia pojedynczego pola danych z pola **Dostępne pola** na listę **Zaznaczone pola**,

- przycisk przeniesienia wszystkich pól z pola **Dostępne pola** na listę **Zaznaczone pola**,
- przycisk przeniesienia pojedynczego pola z listy **Zaznaczone pola** na listę **Dostępne pola**,
- przycisk przeniesienia wszystkich pól z listy **Zaznaczone pola** na listę **Dostępne pola**,
- przycisk anulowania pracy kreatora,
- przycisk przejścia do poprzedniego okna,
- przycisk przejścia do następnego okna,
- przycisk zakończenia pracy kreatora.

3. Ostatnim krokiem jest związanie formularza ze źródłem danych poprzez określenie tabeli lub kwerendy za pomocą listy rozwijanej, zaznaczenie na liście **Dostępne pola** nazwy pól tabeli, a następnie przeniesienie ich na listę **Zaznaczone pola**.

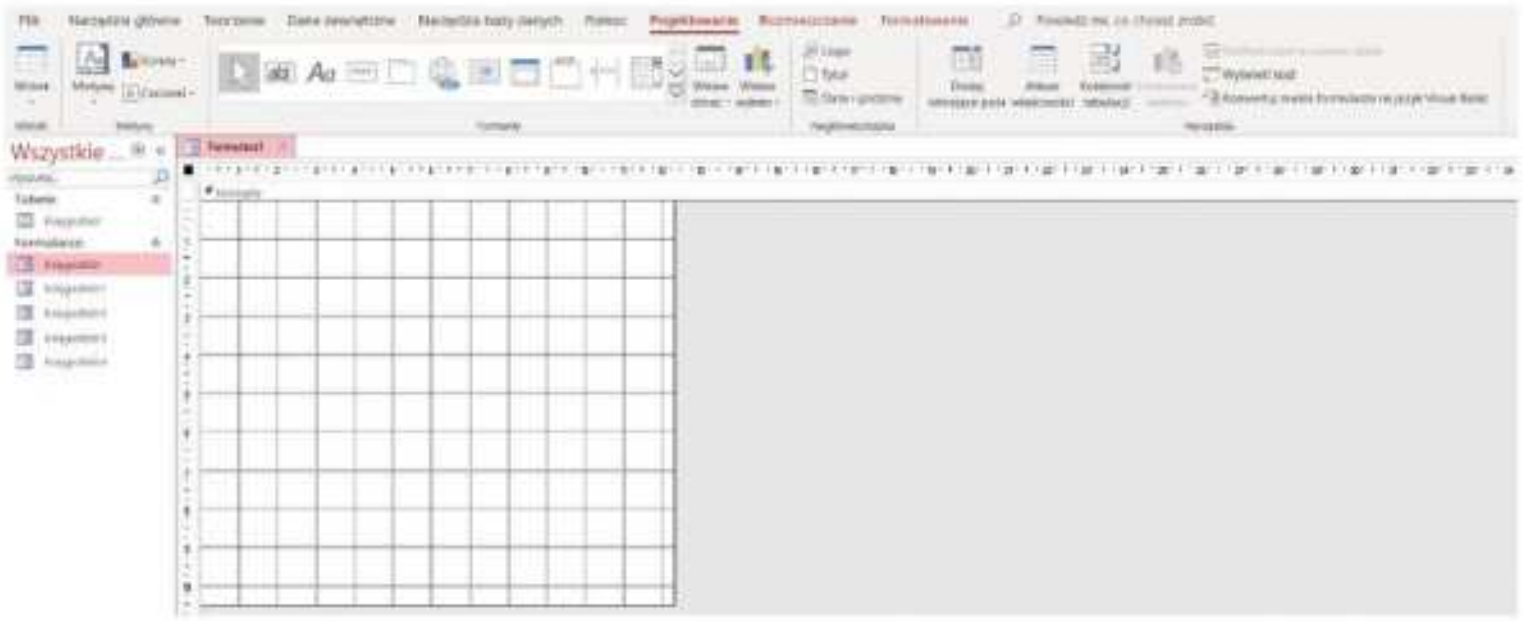
Po związaniu formularza ze źródłem danych naciskamy przycisk **Dalej** – spowoduje to przejście do kolejnego okna kreatora, w którym wybierzemy układ naszego formularza. Po jego wyborze klikamy ponownie **Dalej** – otworzy się okno, w którym będziemy mogli nadać nazwę formularzowi oraz wybrać, czy mamy otworzyć formularz, czy też chcemy zmodyfikować projekt. Pozostawiamy domyślnie zaznaczone **Otwórz formularz**, aby przejrzeć lub wprowadzić informacje, i klikamy **Zakończ**.

19.6. Samodzielne konstruowanie formularza z widoku projektu

widok projektu ▸ Jeżeli żadna z wcześniejszych metod nam nie odpowiada, to możemy sami skonstruować formularz, wykorzystując do tego widok projektu.

Na karcie **Tworzenie** w grupie **Formularze** wybieramy **Projekt formularza**.

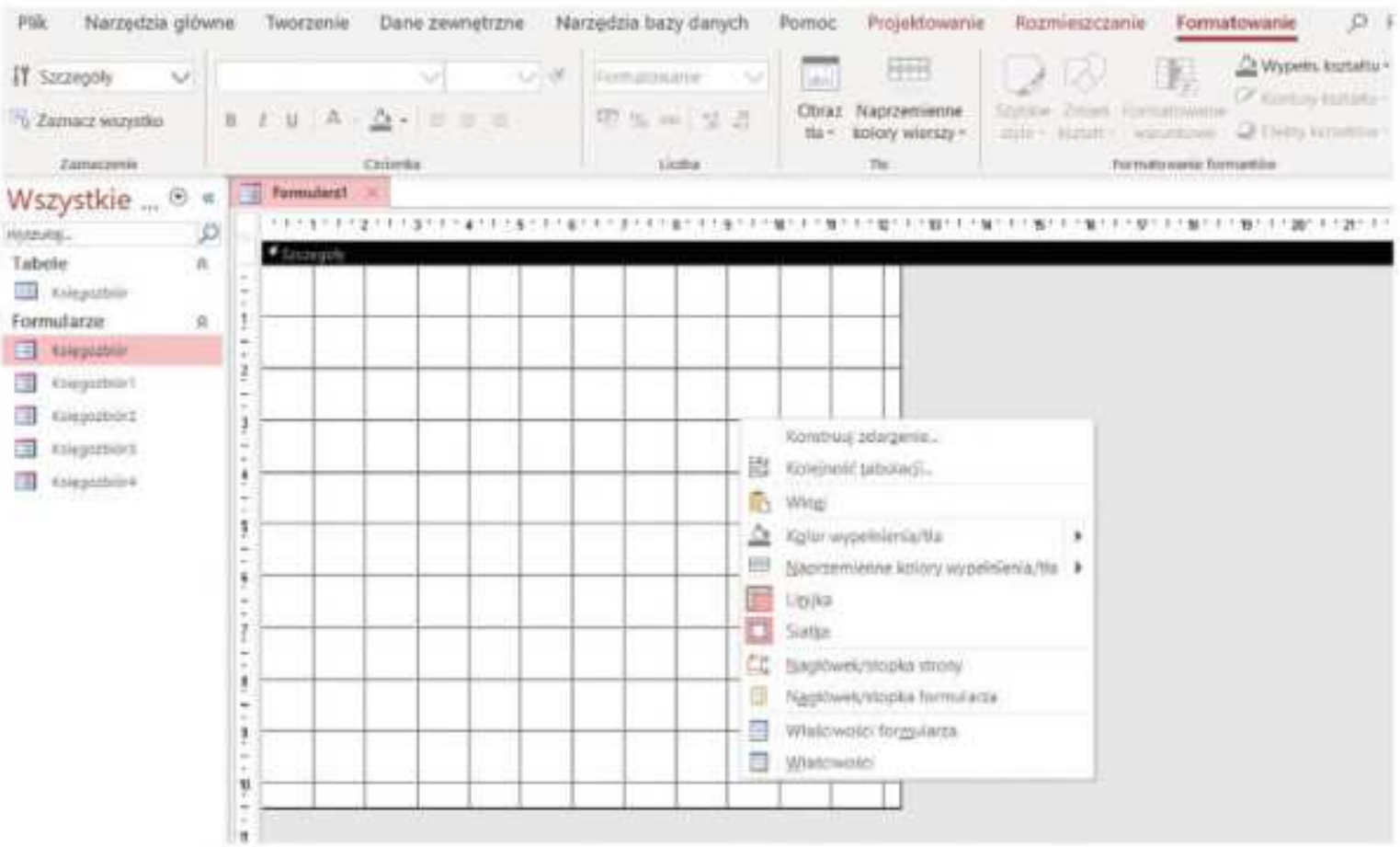
Po jego wybraniu uaktywni się okno formularza (rys. 19.9.).



Rys. 19.9. Okno formularza w widoku projektu

Po wyświetleniu się okna formularza musimy je powiązać z tabelą lub kwerendą. Wstawiamy formanty, które są powiązane z określonymi polami tabeli lub kwerendy (np. pole tekstowe, przyciski opcji) umożliwiające nam wyświetlenie żądanych informacji oraz niezbędne elementy, za pomocą których będzie realizowany proces przetwarzania danych w bazie.

Informacje, które znajdują się w tabelach lub kwerendach, umieścimy w sekcjach formularza. Sekcję utworzymy, klikając lewym przyciskiem myszy dowolne miejsce formularza. Aby dodać nagłówek lub stopkę, klikamy prawym przyciskiem myszy w pole formularza i wybieramy odpowiednią opcję z menu kontekstowego (rys. 19.10.).

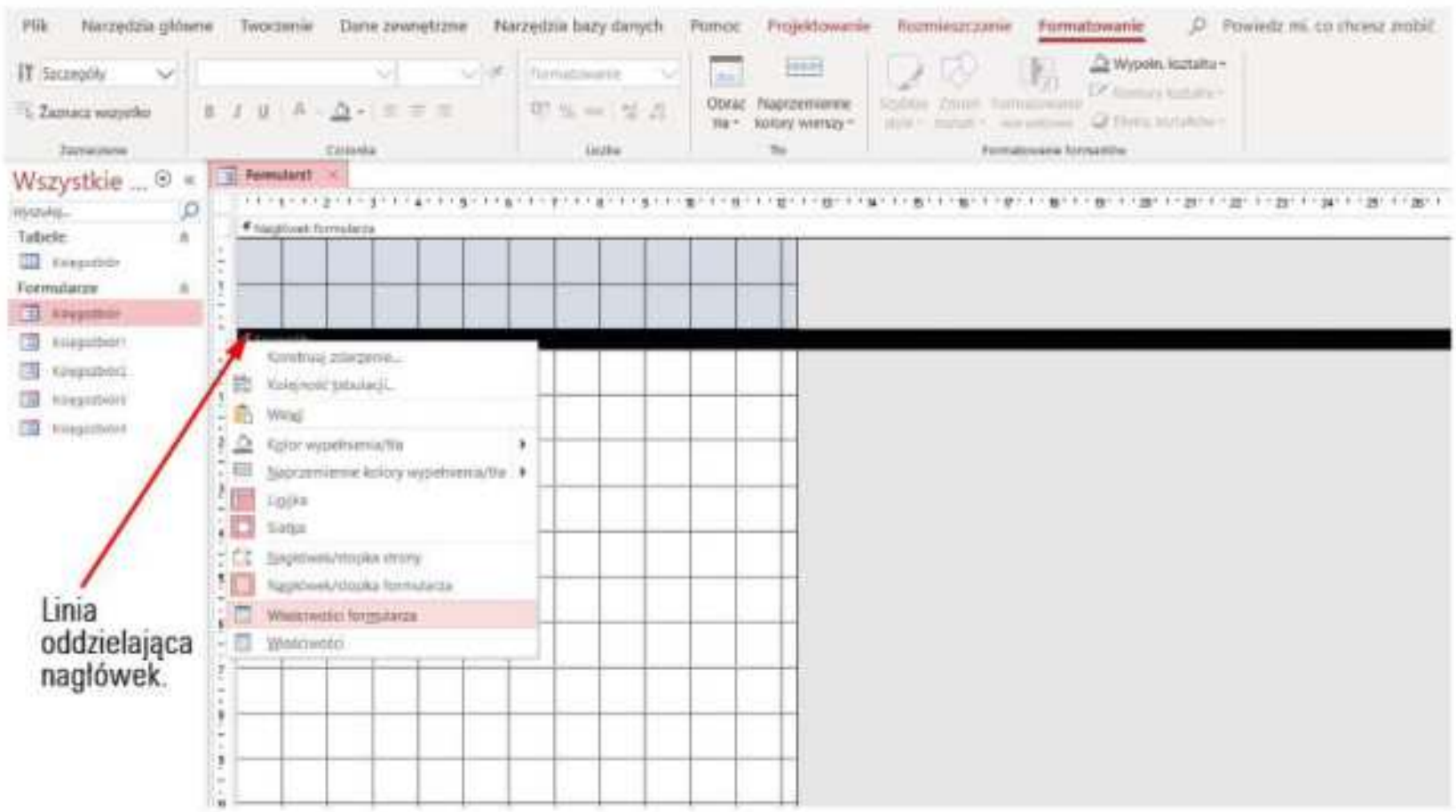


Rys. 19.10. Wstawianie sekcji oraz nagłówka i stopki w formularzu

W nagłówku możemy umieścić informacje, które pojawią się na wydruku, np. tytuł formularza. Nagłówek może także zawierać nagłówki kolumn, dat lub numerów stron. Informacje te pozostaną niezmienne dla każdego rekordu. Stopka natomiast służy do wyświetlenia podsumowania strony, numerów stron, przycisków nawigacyjnych itp. Do prezentacji danych w rekordach służy natomiast sekcja **Szczegóły**, w której rekordy mogą być wyświetlane pojedynczo lub grupowo.

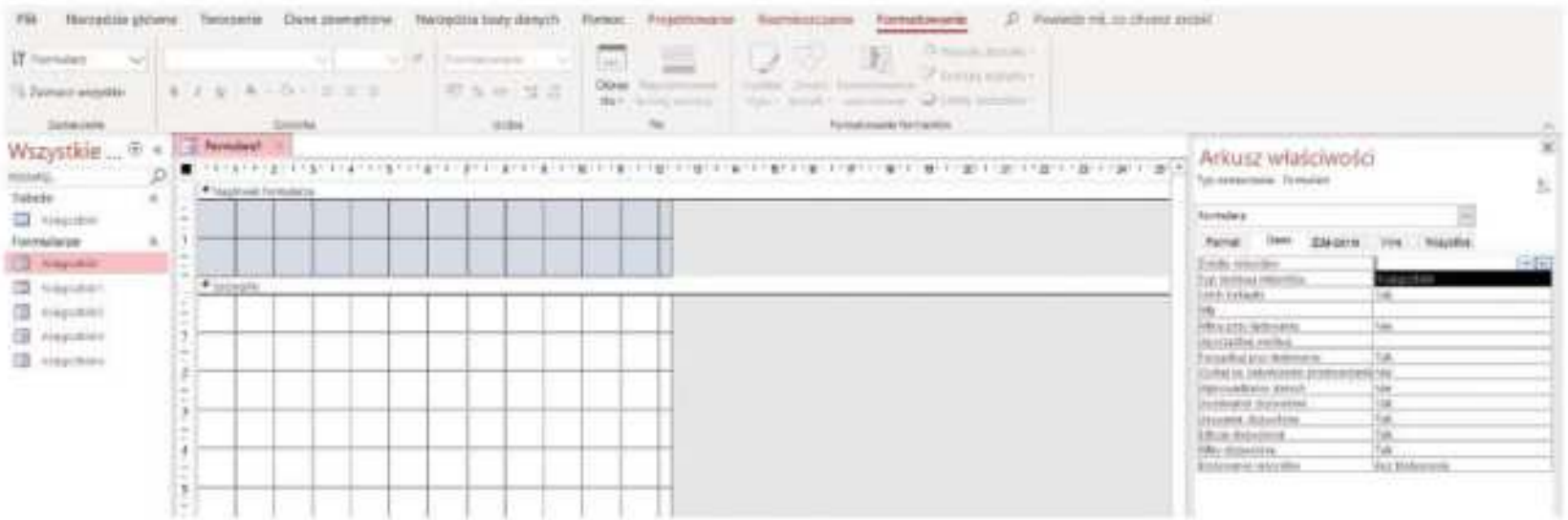
Jeżeli chcemy, aby nasz formularz służył do prezentacji czy aktualizacji danych zawartych w tabelach lub kwerendach, musimy go związać ze źródłem danych. Aby tego dokonać, postępujmy według poniższych kroków:

1. Klikamy w linię oddzielającą nagłówek od szczegółów formularza, a następnie z menu podręcznego wybieramy polecenie **Właściwości formularza** (rys. 19.11.).



Rys. 19.11. Wybór właściwości w menu podręcznym

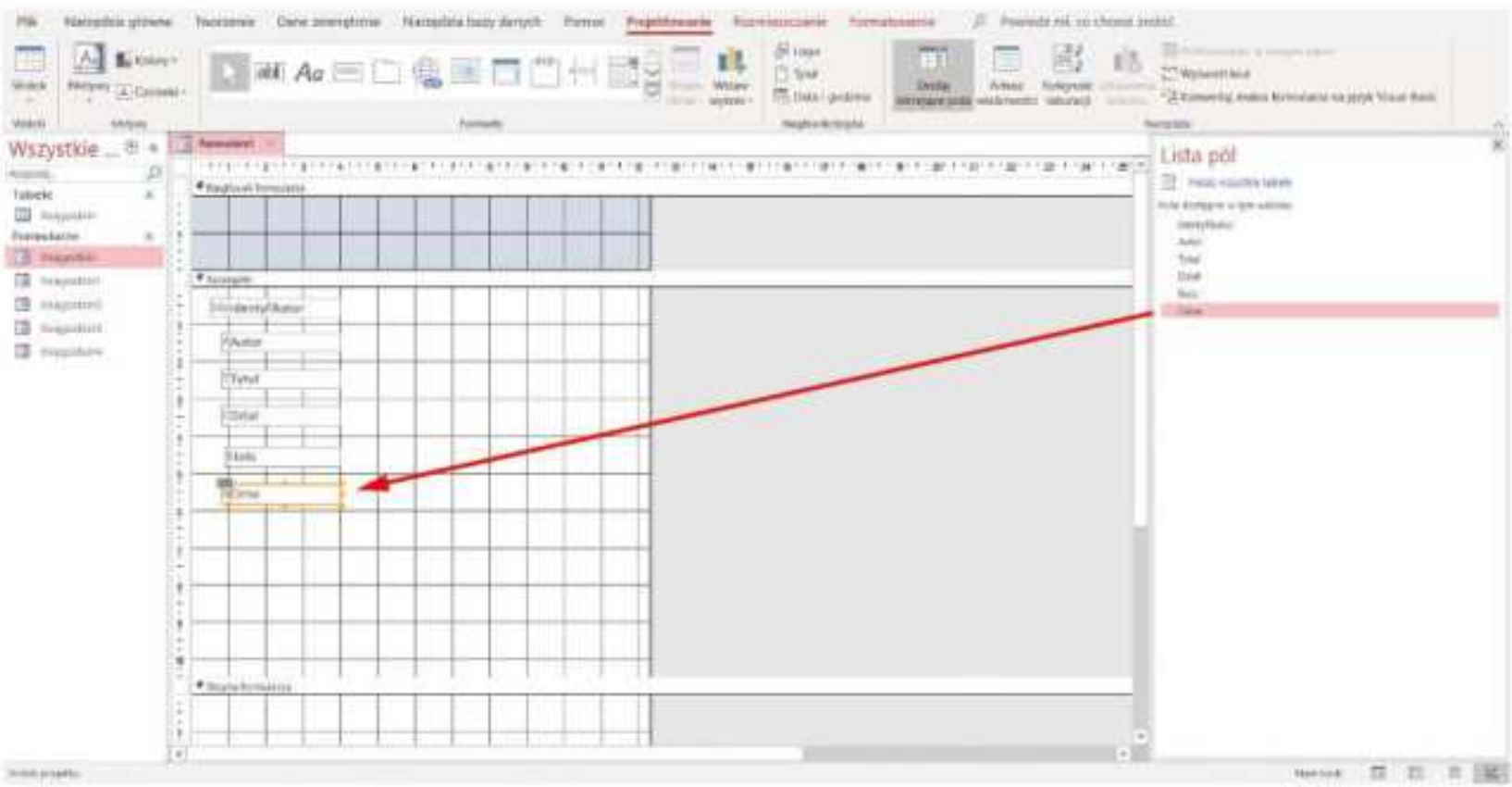
2. Wybór **Właściwości formularza** spowoduje uruchomienie okna **Arkusz właściwości**, w którym wybieramy **Dane formularza**. W polu **Źródło rekordów** klikamy symbol strzałki. Z rozwiniętej listy wybieramy jedną z wartości – tą czynnością dokonamy związania formularza z tabelą lub kwerendą (rys. 19.12.).



Rys. 19.12. Związanie formularza z tabelą **Księgozbiór**

3. Po związaniu formularza z tabelą lub kwerendą aktywujemy listę pól, na której wyświetlą się nazwy wszystkich kolumn dostępnych w tabeli **Księgozbiór**.

Aby umieścić konkretne pole z tabeli w formularzu, przeciągamy je z listy do formularza (rys. 19.13.).



Rys. 19.13. Umieszczanie pól tabeli w formularzu

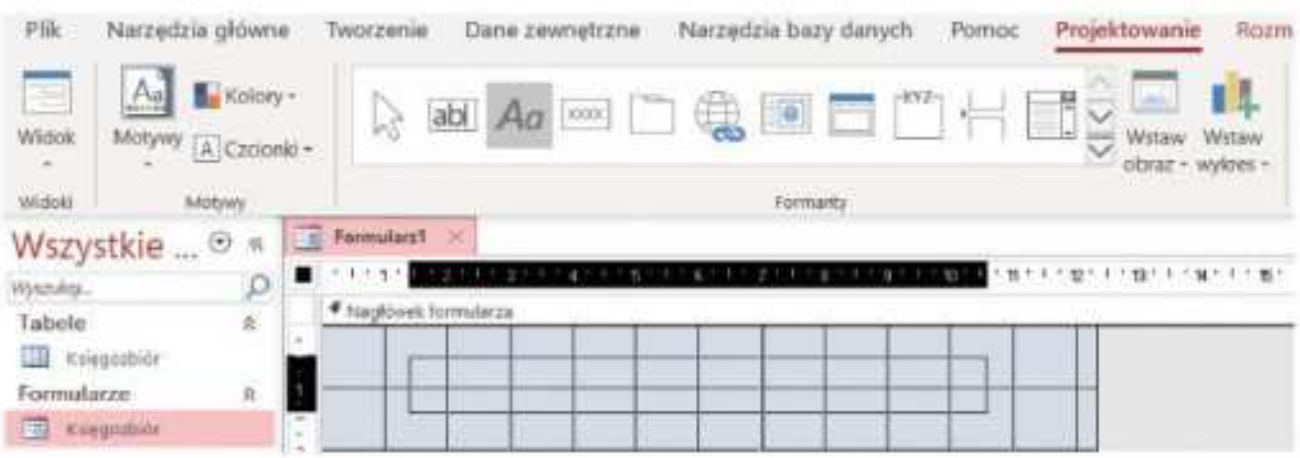
19.7. Formanty

Aby dokończyć tworzenie formularza w widoku projektu, musimy użyć formantów, czyli obiektów aplikacji MS Access, które służą do: ◂ formanty

- przechowywania, wyświetlania lub wprowadzania danych alfanumerycznych,
- przechowywania i wyświetlania grafiki, wideo, dźwięku itp.,
- wykonywania akcji.

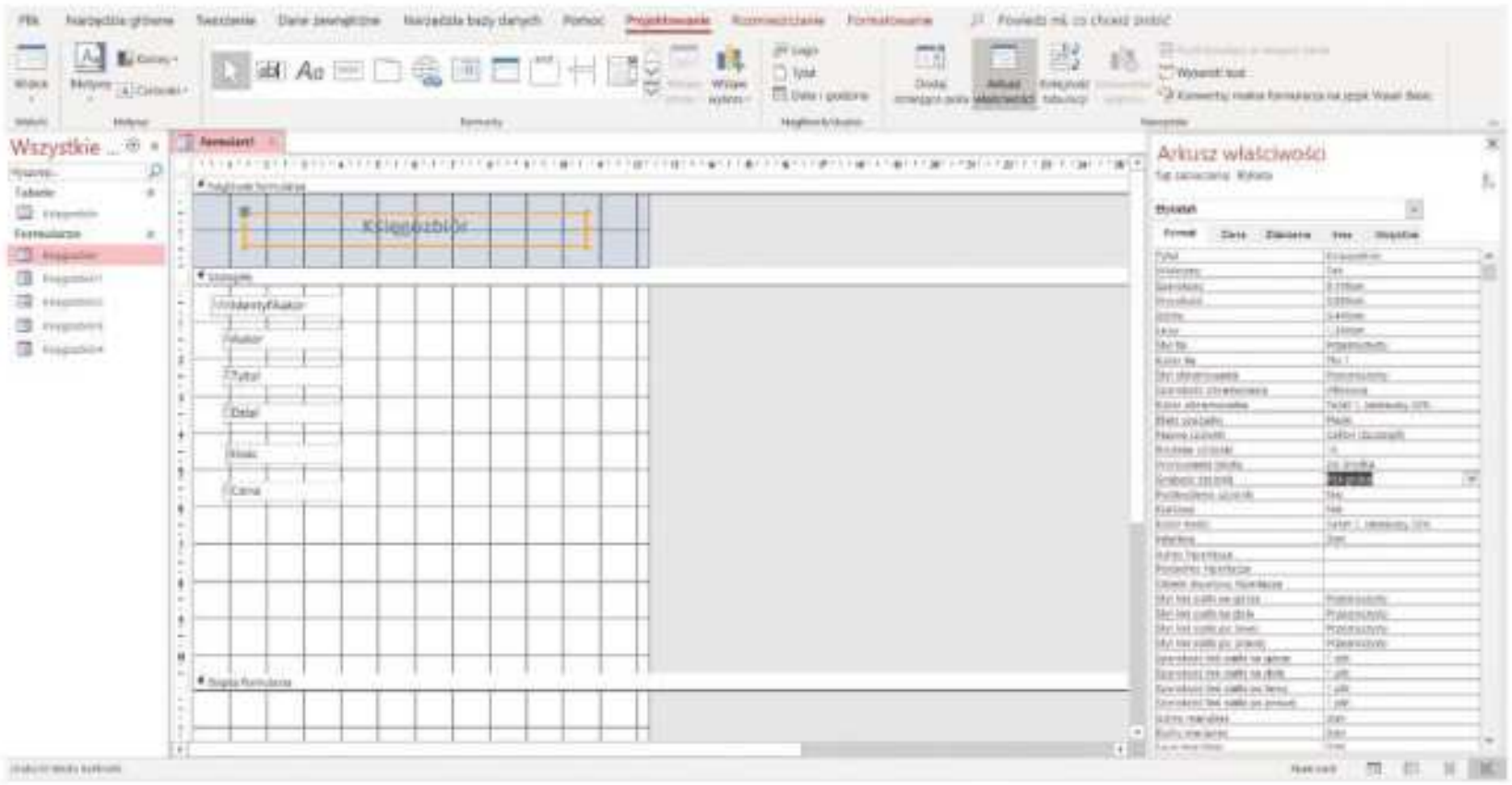
Aby umieścić formant na formularzu, musimy wykonać następujące czynności:

1. Otwieramy formularz w widoku projektu, a następnie z karty **Projektowanie** w grupie **Formanty** wybieramy interesujący nas formant. W wybranej sekcji formularza umieszczamy wskaźnik myszy – w miejscu, w którym ma się znajdować górny prawy róg naszego formantu. Wciskając lewy przycisk myszy, zaznaczamy obszar, który ten formant ma zajmować. Po osiągnięciu celu puszczamy przycisk myszy (rys. 19.14.).



Rys. 19.14. Wstawianie formantu

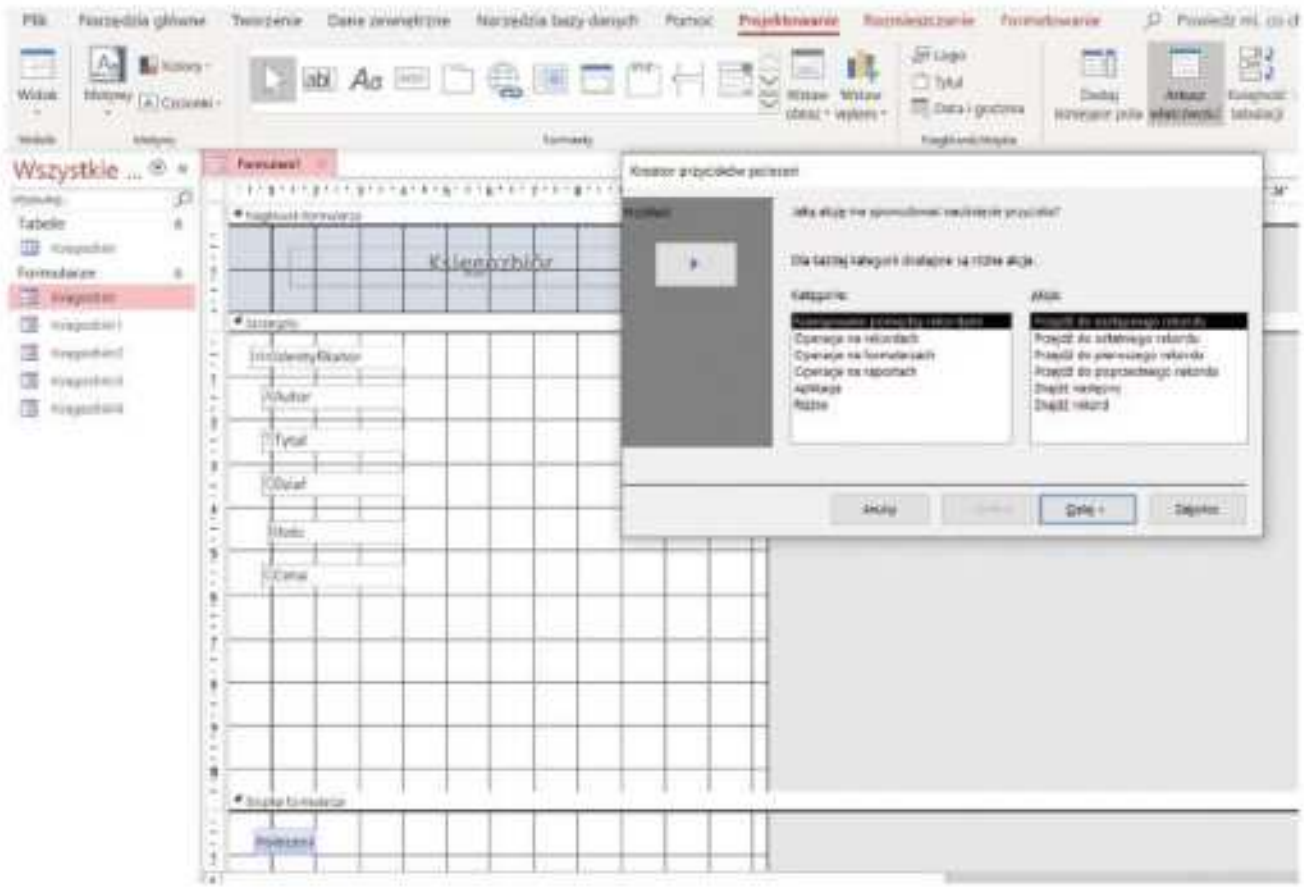
2. Po wstawieniu formantu uruchomi się okno z właściwościami, w którym możemy wybrać kategorię właściwości – **Format**, **Dane** lub **Zdarzenie** (rys. 19.15.).



Rys. 19.15. Formatowanie formantu

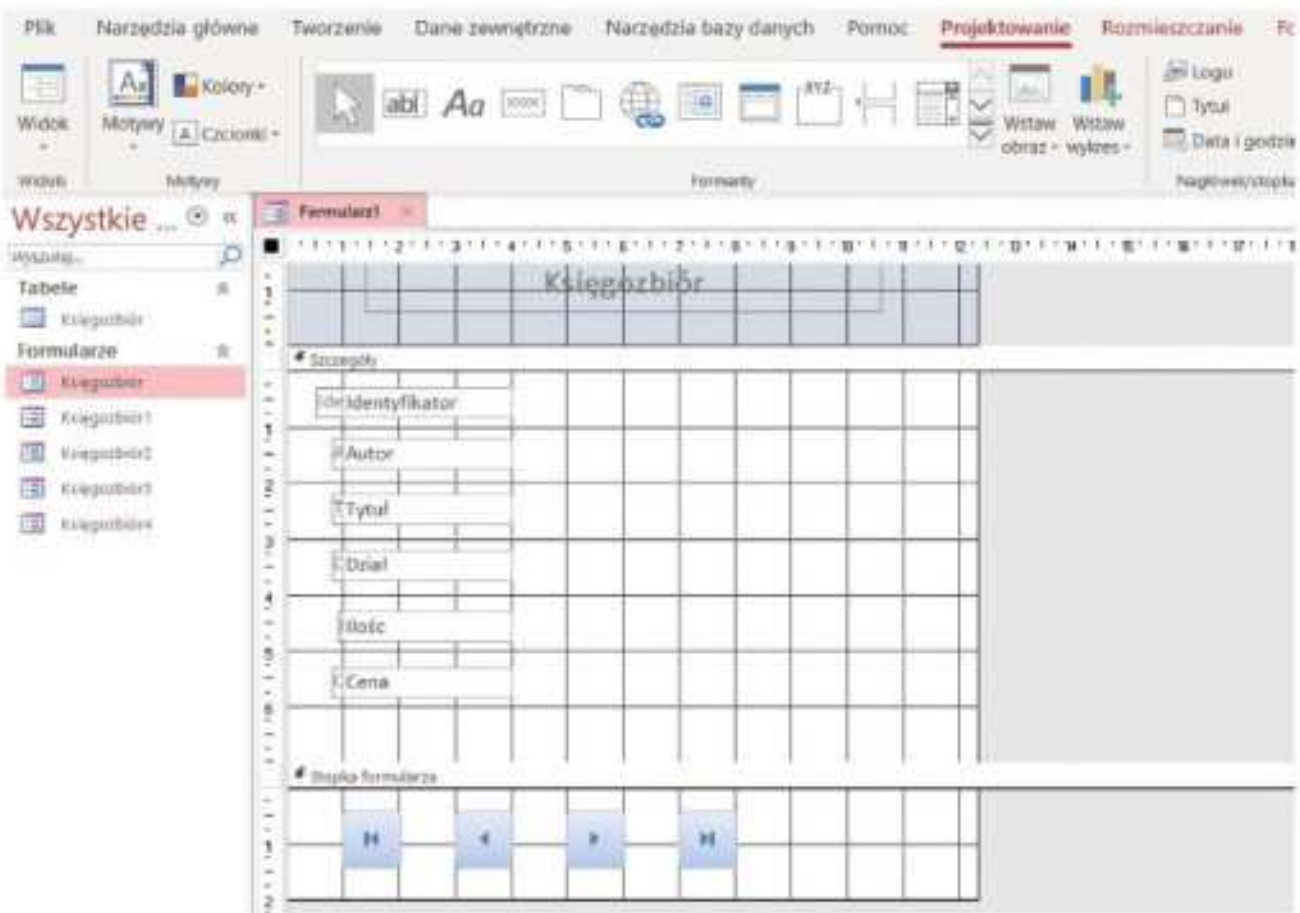
Jeżeli chcielibyśmy w stopce wstawić przyciski nawigacyjne, musimy wykonać następujące czynności:

1. Wybieramy formant **Przycisk** i umieszczamy go w stopce. Uruchomi się kreator przycisków poleceń (rys. 19.16.).

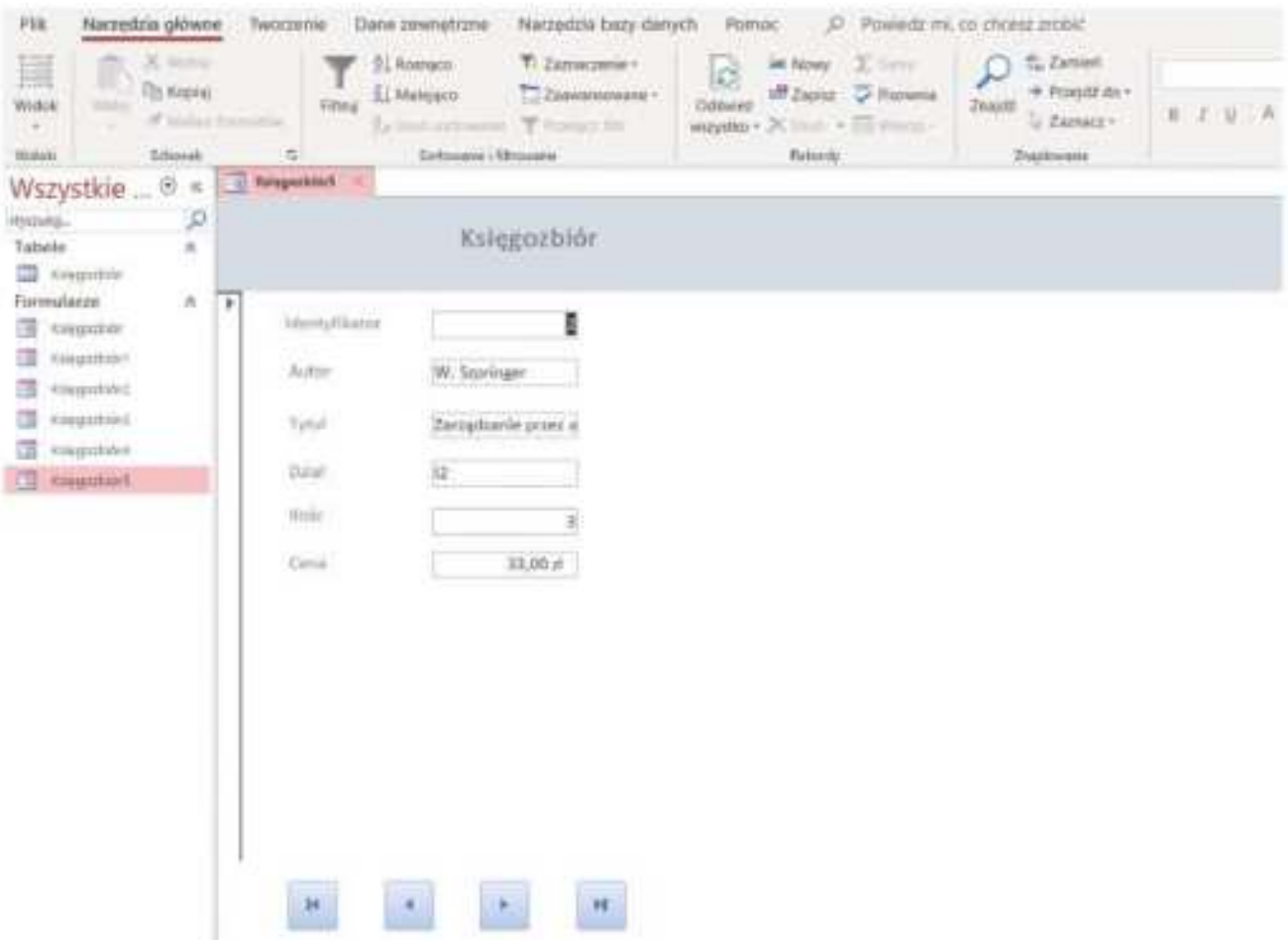


Rys. 19.16. Wstawianie przycisku nawigacyjnego

2. Wybierając odpowiednie **Kategorie** i akcje, formatujemy nasz przycisk (rys. 19.17. i 19.18.).



Rys. 19.17. Przyciski nawigacyjne



Rys. 19.18. Formularz utworzony w widoku projektu

ZADANIA DO WYKONANIA

1. Utwórz prezentację, w której przedstawisz definicję i przykłady zastosowania kwerend w programie Access. Skorzystaj z tutoriali z opcji **Pomoc**, w tym także wideo. Do wykonania prezentacji użyj dowolnego programu, np. PowerPoint.
2. Korzystając z bazy danych utworzonej w poprzednim ćwiczeniu, skonstruuj formularz – wiele elementów.
3. Utwórz bazę danych **księgarnia** z tabelą **książki**:

Nazwa pola	Typ danych	Rozmiar pola	Wymagane
Identyfikator	Autonumerowanie	Liczba całkowita długa	
autor	Krótki tekst	16	TAK
tytul	Krótki tekst	16	TAK
liczba_stron	Liczba	Liczba całkowita długa	TAK
wydawnictwo	Krótki tekst	255	TAK
rok_wydania	Liczba	Liczba całkowita długa	TAK
isbn	Liczba	Liczba całkowita długa	TAK

Wypełnij tabelę danymi. Wstaw pięć rekordów. Następnie skonstruuj formularz, korzystając z kreatora formularzy.

4. Korzystając z bazy danych z poprzedniego ćwiczenia, skonstruuj formularz w widoku projektu.

PODSUMOWANIE LEKCJI

s. 115 ▸ • Formularz jest jednym z obiektów bazy danych, który umożliwia wprowadzanie danych oraz wyświetlanie zawartych w niej informacji. Dzięki nim możemy określić sposoby interakcji użytkowników z danymi zawartymi w bazie danych.

s. 117 ▸ • Formularz z wieloma elementami umożliwia przedstawienie formularza z wieloma rekordami jednocześnie (zwykły formularz wyświetlał tylko jeden rekord). Wiele elementów jest formularzem typu tabelarycznego.

s. 118 ▸ • Arkusz danych umożliwia utworzenie formularza pokazującego wiele rekordów w arkuszu danych, po jednym rekordzie na wiersz – jest on odmianą formularza z wieloma danymi.

s. 118 ▸ • Formularz dzielony umożliwia nam utworzenie formularza, w którym w górnej sekcji jest pokazany arkusz danych, a w dolnej sekcji znajduje się formularz do wprowadzania informacji dotyczących rekordu wybranego w arkuszu danych.

- Kreator formularza umożliwia stworzenie prostych formularzy krok po kroku. ◂ s. 119
- Aby stworzyć indywidualny formularz od podstaw, całkowicie dostosowany w układzie i wyglądzie do naszych potrzeb, trzeba skorzystać z możliwości tworzenia formularza w widoku projektu. ◂ s. 120
- Formanty są powiązane z określonymi polami tabeli lub kwerendy (np. pole tekstowe, przyciski opcji) i umożliwiają wyświetlenie żądanych informacji oraz niezbędnych elementów, za pomocą których będzie realizowany proces przetwarzania danych w bazie. ◂ s. 123

20. Czy to możliwe, czyli baza danych w arkuszu Excel

NA TEJ LEKCJI:

- dowiesz się, jak wprowadzać i gromadzić dane w arkuszu Excel;
- zbudujesz prostą bazę w arkuszu;
- zaimportujesz tabelę z bazy Access.

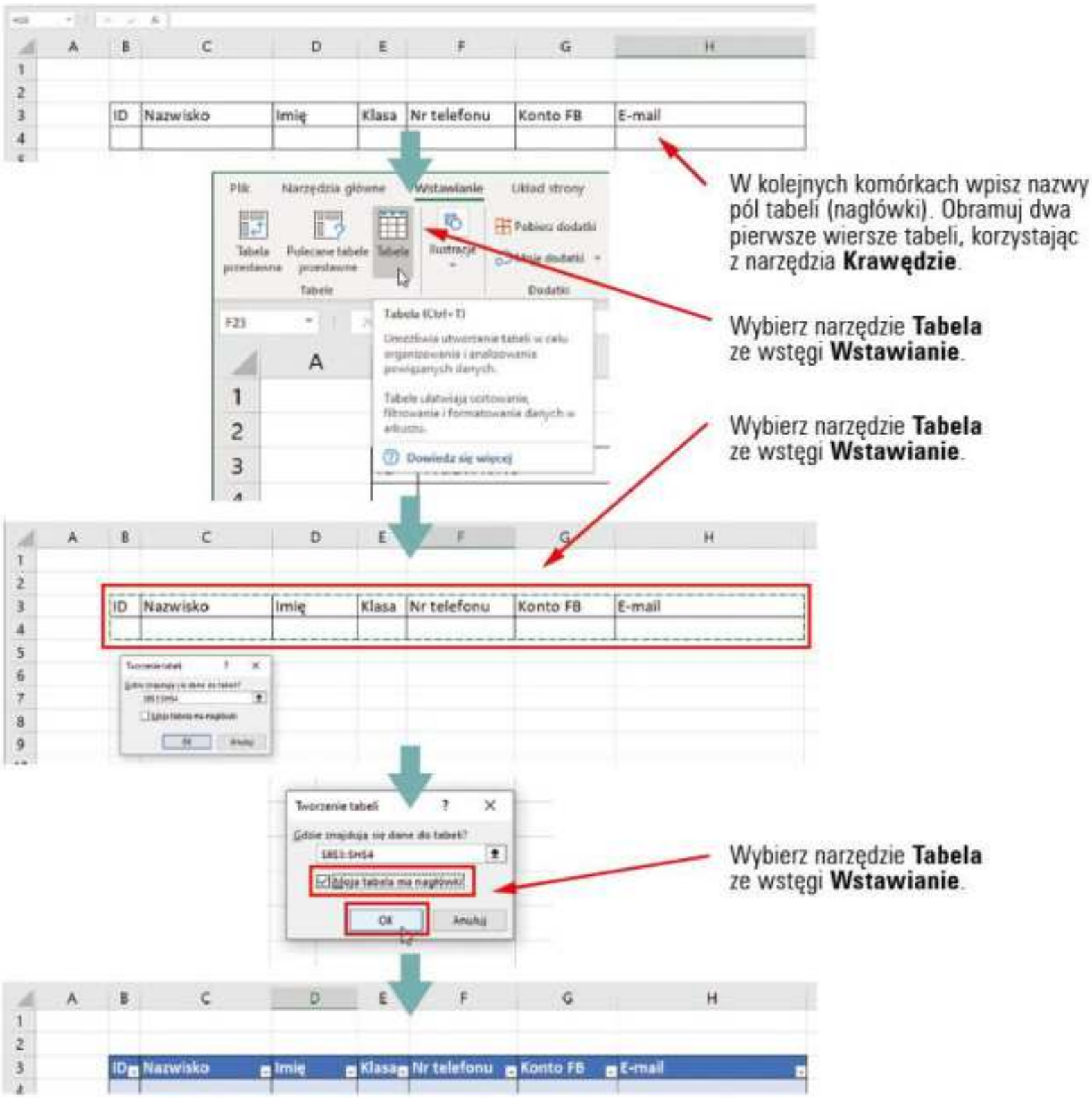
Wiesz, że baza gromadzi dane w tabelach. Skoro arkusz programu Excel (i innych tego typu programów) składa się z kolumn oraz wierszy i w każdym możemy tworzyć wiele tabel, to nasuwa się pytanie, czy mogą się one stać elementami bazy danych.

20.1. Analogie, czyli dane w tabelach programu Excel

Program Excel jest bardzo rozbudowanym narzędziem używanym w wielu dziedzinach, np. w księgowości (ze względu na swoje możliwości prowadzenia obliczeń), planowaniu i symulacjach ekonomicznych, a także w gromadzeniu danych. Oczywiście tworzenie bazy danych w programie Excel wiąże się z zachowaniem pewnych zasad, które poznasz w trakcie kolejnych ćwiczeń w tym rozdziale.

Zacznijmy od utworzenia w arkuszu odpowiednio zaprojektowanej tabeli (rys. 20.1.). W przykładzie posłużymy się takimi samymi wytycznymi, jak podczas budowania bazy Access w poprzednich rozdziałach.

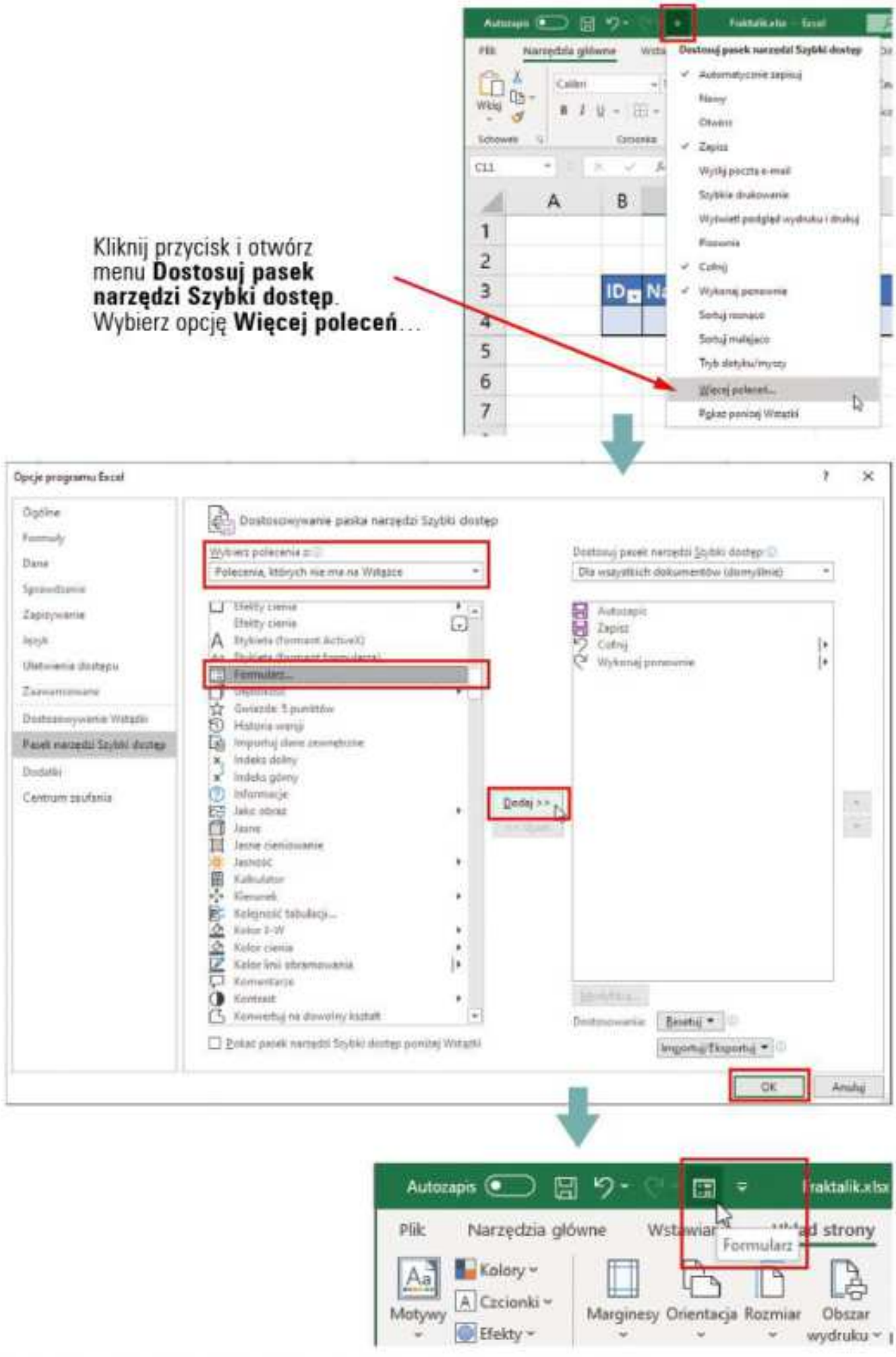
Uwaga! Prezentowany sposób nie jest jedynym rozwiązaniem problemu tworzenia tabel. Możesz również użyć narzędzia **Formatuj jako tabelę** z wstęgi **Narzędzia główne**.



Rys. 20.1. Tworzenie tabeli w programie Excel

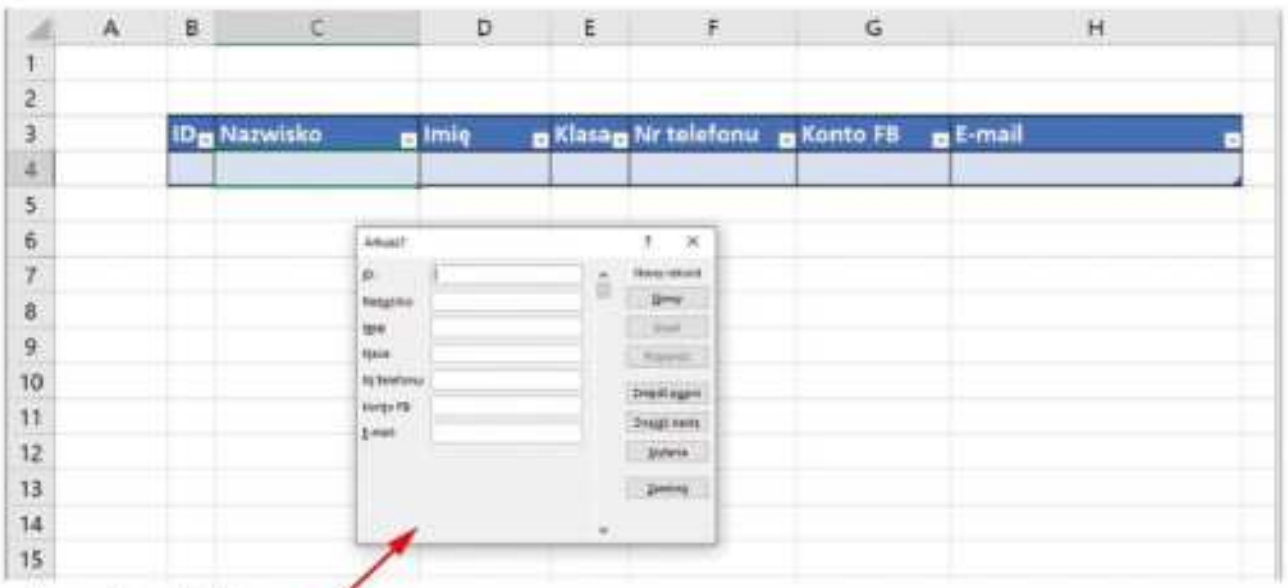
narzędzie Formularz

Skoro mamy już tabelę, powinniśmy wypełnić ją danymi. Podobnie jak w bazie Access można to zrobić bezpośrednio w komórkach (polach) tabeli lub utworzyć formularz. W programie Excel istnieje specjalne narzędzie do tworzenia formularzy. Nie znajdziesz go jednak na wstęgach. Wprowadźmy skrót do tego narzędzia na pasek narzędzi **Szybki dostęp** (rys. 20.2.).



Rys. 20.2. Wstawianie narzędzia do paska narzędzi **Szybki dostęp**

Skoro narzędzie jest już dostępne, możesz przystąpić do tworzenia formularza (rys. 20.3.). Zauważ, że twoja tabela ma wiersz z nagłówkami i jeden wiersz na dane. Jeśli za pomocą formularza będziemy dodawać kolejne rekordy, będą one powstawać samoczynnie i otrzymają takie formatowanie, jak pierwszy wiersz.



Kliknij dowolne pole tabeli i wybierz narzędzie **Formularz**. Po chwili wyświetli się okno formularza zawierające nazwy wszystkich nagłówków tabeli.

Po wpisaniu danych w odpowiednie pola wybierz przycisk **Nowy**. Dane zostaną zapisane w wierszu, a formularz będzie gotowy do wpisywania następnych. Formatowanie wierszy będzie zgodne ze stylem.



ID	Nazwisko	Imię	Klasa	Nr telefonu	Konto FB	E-mail
1	Nowak	Anna		221144	NowAnna	na@12345.eu
2	Kowalski	Jan		420182	KowalJa	kj@qwertyu.eu
3	Nazwiskowski	Wojciech		1231111	Nazwiskowski	nw@10040.eu
4	Imionowski	Staszek		211111	Stach	stach@10040.eu
5	Bazodanowski	Wojciech		1222222	Bazodan	bazodan@oi12.eu

Rys. 20.3. Używanie formularza do wpisywania danych do tabeli

Uwaga!

Każde pole tabeli (komórkę) możesz formatować, czyli wskazać, jakiej kategorii dane mogą być do niej wpisywane (liczbowe, tekstowe, daty itp.). Domyślnie w programie Excel pola przechowują dane kategorii **Ogólne**. W naszym ćwiczeniu nie musisz tego zmieniać. Jeśli jednak uznasz to za konieczne, np. w innej tabeli, zmień kategorię, klikając komórkę prawym przyciskiem i wybierając z menu kontekstowego opcję **Formatuj komórki...**

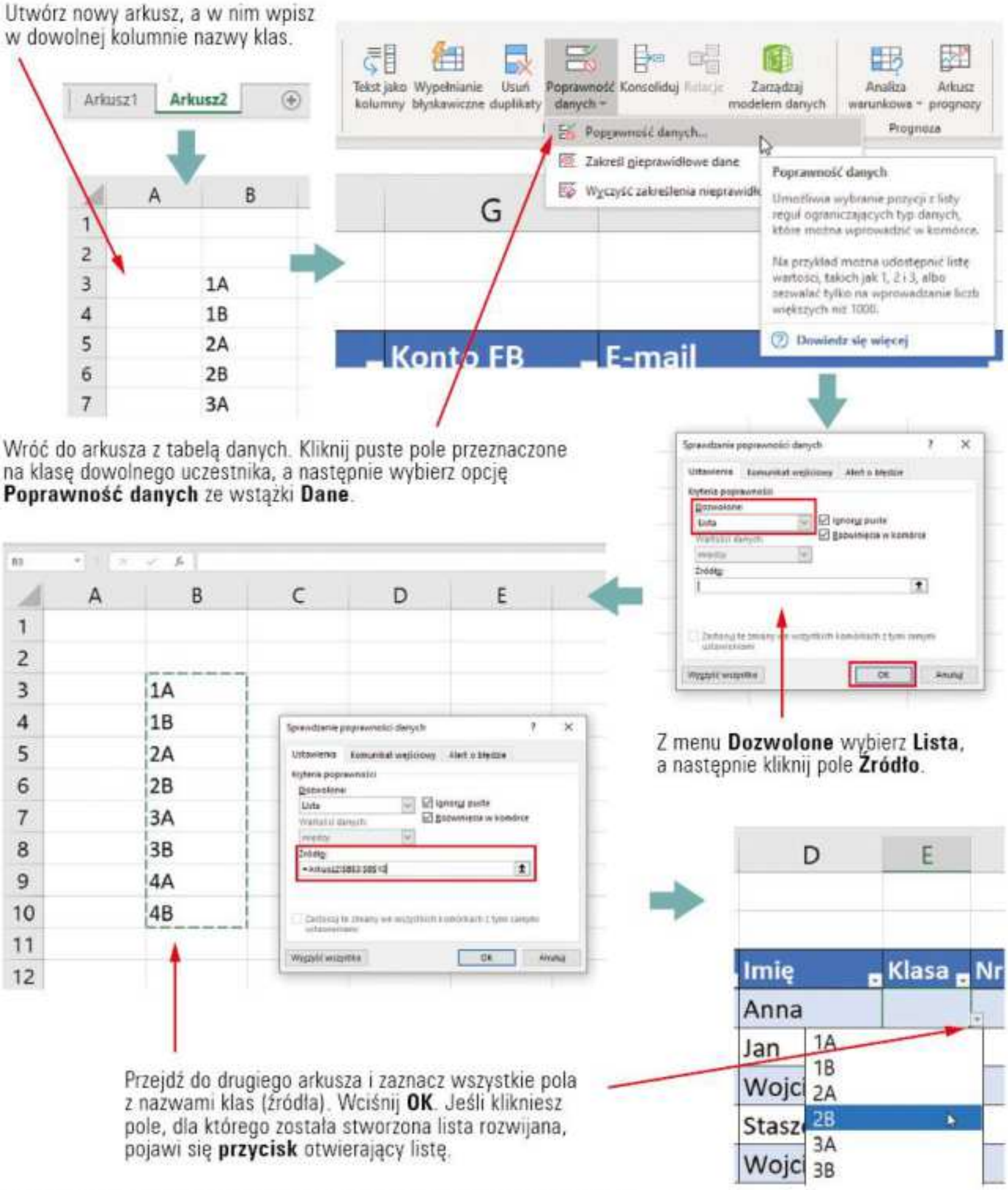
Uwaga!

Pola ID będą pełnić podobną funkcję jak identyfikatory, dlatego powinny zostać wypełnione kolejnymi liczbami naturalnymi.

Uwaga!

Udostępniając dane z bazy, zachowaj zasady prawa określone przez Unię Europejską. Znajdziesz je w internecie.

Nietrudno zauważyć, że w przykładzie nie wpisano klas. Przypomnij sobie, w jaki sposób wprowadzaliśmy te dane do tabeli w bazie Access. Zróbmy taką **listę rozwijaną** także w naszej tabeli programu Excel (rys. 20.4.).

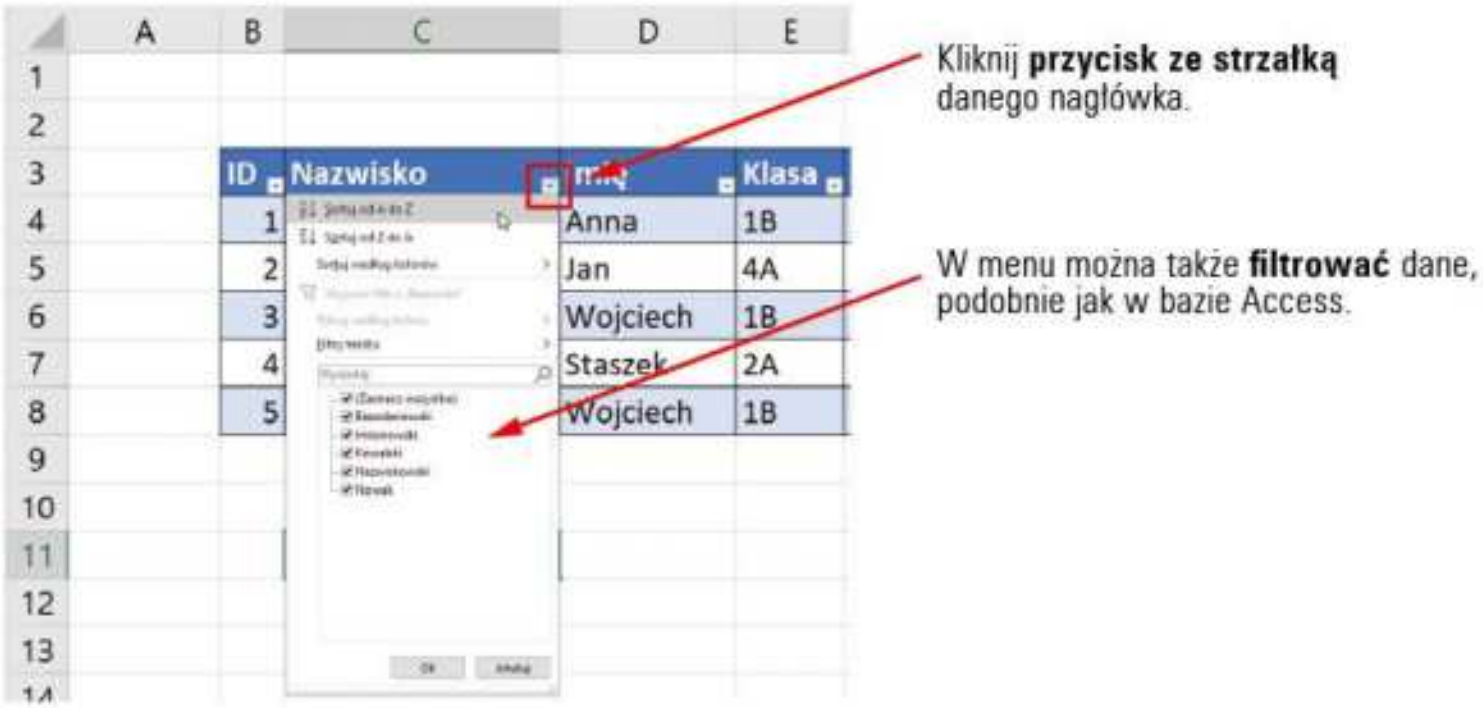


Rys. 20.4. Tworzenie listy rozwijanej dla pola w tabeli programu Excel

Ponieważ tej samej listy będziemy używać do wszystkich pól, w których trzeba wpisać klasę, wystarczy użyć opcji kopiowania pola (ze zdefiniowaną listą) do schowka i wkleić do pozostałych.

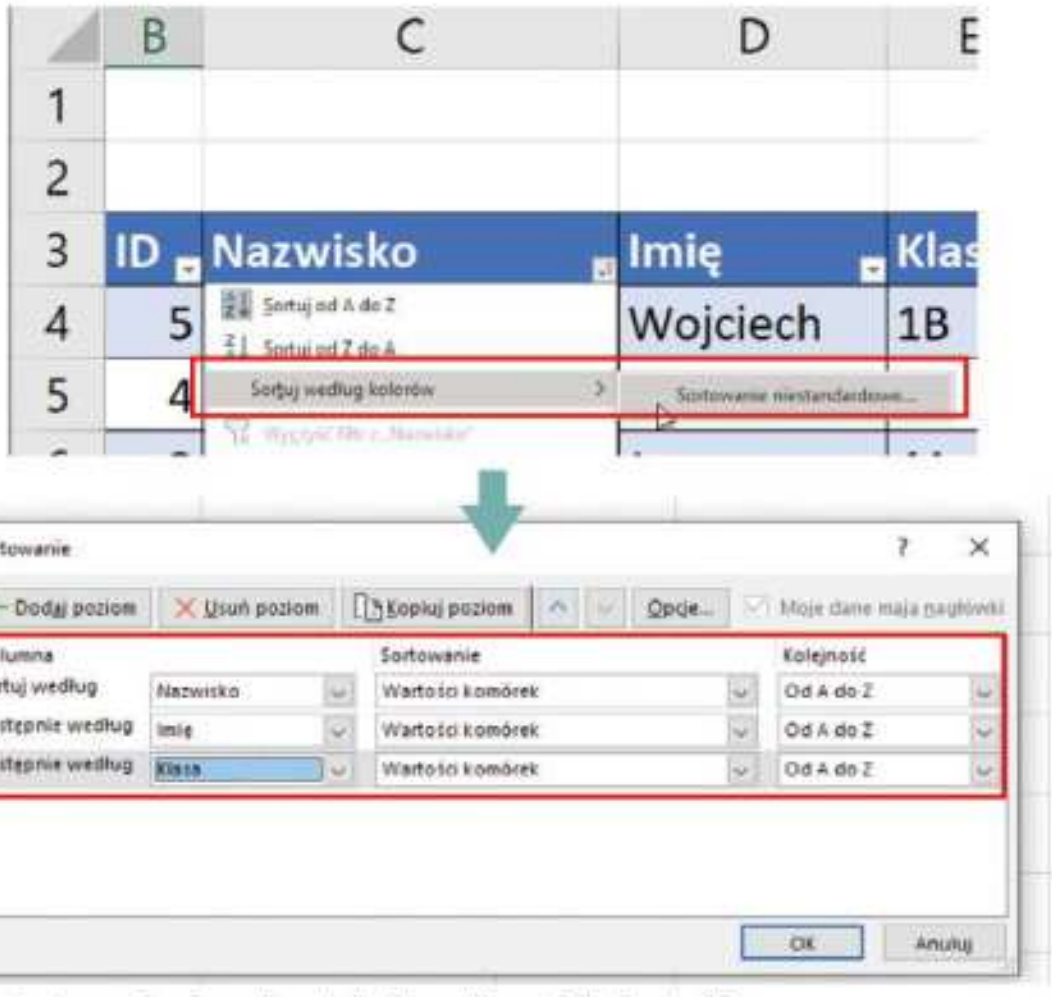
20.2. Ustawiamy kolejność, czyli sortowanie i filtrowanie danych w tabelach programu Excel

sortowanie danych ▸ Sortowanie danych w tabelach programu Excel odbywa się tak samo jak w bazie Access. Warunkiem jest sformatowanie wiersza nagłówków (rys. 20.5.). W naszym przypadku nagłówki powstały automatycznie podczas tworzenia tabeli.



Rys. 20.5. Sortowanie i filtrowanie rekordów (wierszy tabeli) w programie Excel

Dane w programie Excel można sortować także według kilku kryteriów (rys. 20.6.). Jest to szczególnie przydatne, gdy dane w kilku polach są identyczne, np. nazwiska. Wtedy można ustalić następne kryteria, np. imię, klasa itd.



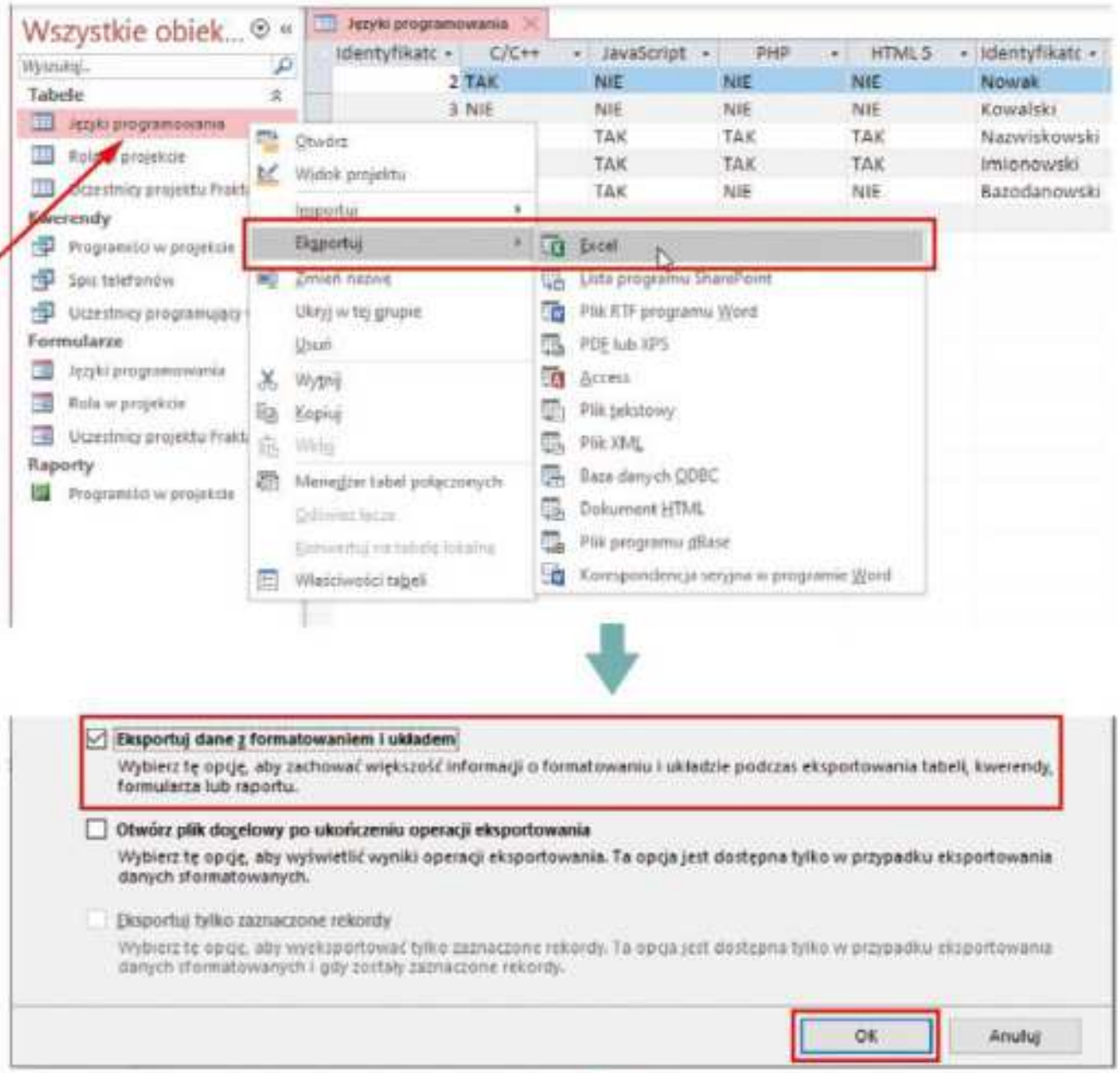
Rys. 20.6. Sortowanie danych w tabeli według wielu kryteriów

20.3. Z bazy Access do programu Excel, czyli importowanie tabel

Jeśli chcesz wkleić kompletną tabelę z bazy danych Access do programu Excel, musisz ją zapisać w odpowiednim formacie (rys. 20.7.).

eksport danych z bazy Access

W bazie **Access** kliknij prawym przyciskiem nazwę tabeli i z menu wybierz **Eksportuj**.



Plik z tabelą odczytany w programie Excel.

	A	B	C	D	E	F
1	Identyfikator	C/C++	JavaScript	PHP	HTML 5	Identyfikator uczestnika
2	2	TAK	NIE	NIE	NIE	Nowak
3	3	NIE	NIE	NIE	NIE	Kowalski
4	4	TAK	TAK	TAK	TAK	Nazwiskowski
5	5	NIE	TAK	TAK	TAK	Imionowski
6	6	TAK	TAK	NIE	NIE	Bazodanowski
7						

Rys. 20.7. Proces eksportu tabeli z bazy **Access** do formatu **Excel**

ZADANIA DO WYKONANIA

1. Wczytaj do arkusza tabelę z bazy danych Access **Fraktalik**. Zmodyfikuj ją w taki sposób, by stała się identyczna z prezentowaną w rozdziale.
2. Utwórz tabelę z fikcyjnymi nazwiskami, imionami i nickami. Sortuj dane według trzech kryteriów. Zmieniaj kolejność ich ważności i obserwuj zachowanie się rekordów. Omów je i powiąż ze zmianami hierarchii pól sortowania.
3. W ćwiczeniu z rozdziału narzędzie **Formularz** umieszczono w pasku narzędzi **Szybki dostęp**. Umieść to narzędzie w jednej ze wstążek. Wykonaj zrzuty ekranowe kolejnych czynności i użyj ich do opisania wykonanych czynności. Opis utwórz w edytorze tekstu i zapisz w formacie PDF.

PODSUMOWANIE LEKCJI

s. 132 • W programie Excel zgromadzone dane można **sortować** zgodnie z jednym kryterium – lub wieloma.

s. 128 • Do wpisywania danych można użyć narzędzia **Formularz**. Należy go przedtem dołączyć do wstążki lub paska narzędzi **Szybki dostęp**.

s. 133 • Każdą tabelę w bazie Access można **wyeksportować** do formatu Excel. Zostanie ona zapisana jako samodzielny arkusz.

21. Wiedza w sieci, czyli internet mądrych ludzi

NA TEJ LEKCJI:

- dowiesz się, jak działają wyszukiwarki w internetowych bazach danych;
- nauczysz się skutecznie wyszukiwać dane w internecie;
- poznasz cenne internetowe bazy danych.

Jak znaleźć informację w sieci? Tylko z pozoru to banalne pytanie. Oczywiście większość potrzebnych danych znajdziesz za pośrednictwem wyszukiwarek, ale czy na pewno umiesz się nimi sprawnie posługiwać? Czy zawsze w wyniku ich działania otrzymujesz to, czego szukasz? A może zamiast kilku odpowiedzi wyszukiwarka wyświetla kilka tysięcy? Jak wśród nich znaleźć tę właściwą i wiarygodną? Co o tym decyduje? Jeśli nie wiesz, czas zweryfikować swoją wiedzę na temat wyszukiwania informacji.

21.1. Odpowiedź w ułamku sekundy, czyli jak działają wyszukiwarki

Wyobraź sobie, że chcesz poszukać informacji na dany temat. Co zrobisz? Oczywiście użyjesz wyszukiwarki, wpisując w odpowiednim miejscu słowa związane z danym zagadnieniem. Co dzieje się dalej? Czy teraz wyszukiwarka zacznie przeglądać cały internet?

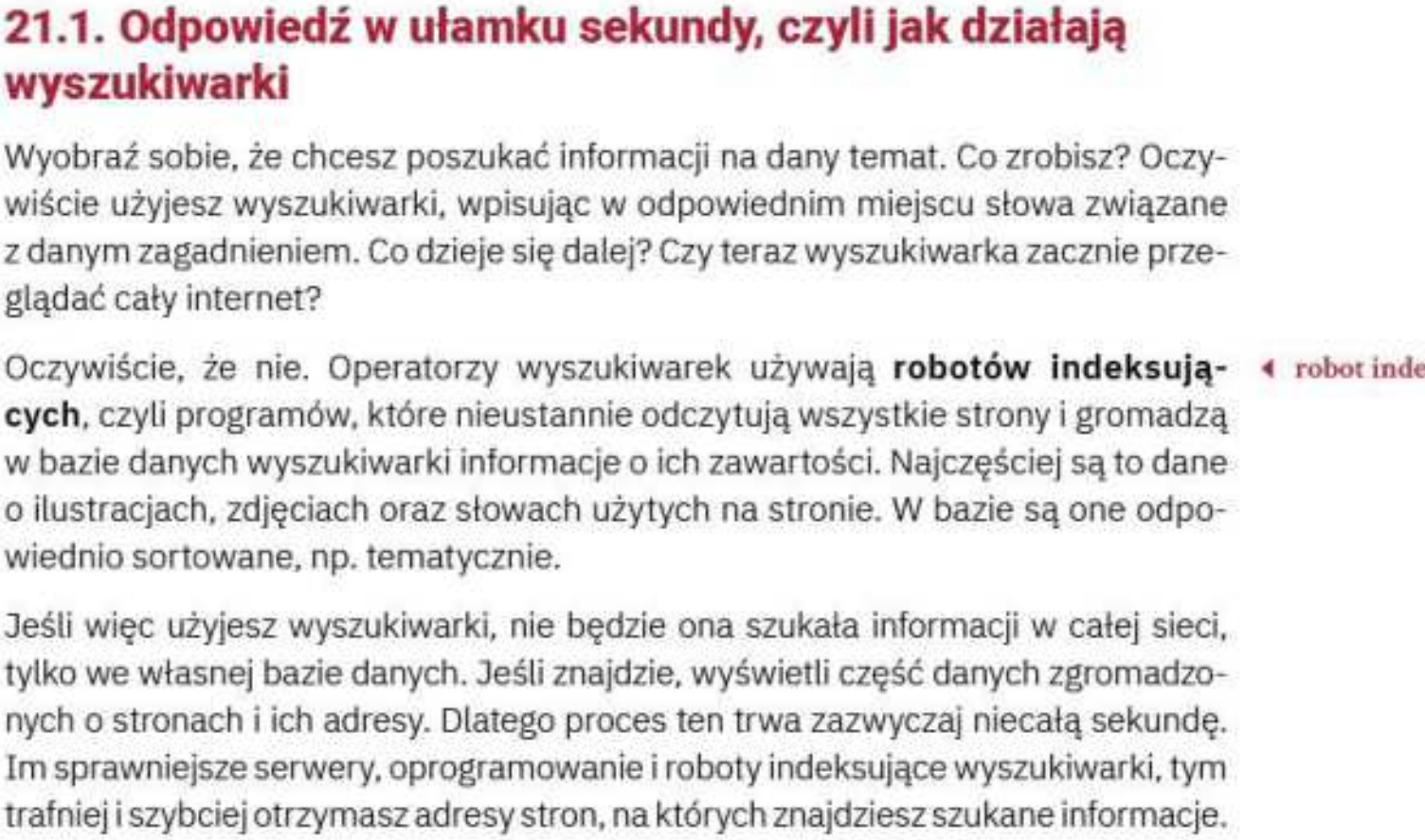
Oczywiście, że nie. Operatorzy wyszukiwarek używają **robotów indeksujących**, czyli programów, które nieustannie odczytują wszystkie strony i gromadzą w bazie danych wyszukiwarki informacje o ich zawartości. Najczęściej są to dane o ilustracjach, zdjęciach oraz słowach użytych na stronie. W bazie są one odpowiednio sortowane, np. tematycznie.

◀ robot indeksujący

Jeśli więc użyjesz wyszukiwarki, nie będzie ona szukała informacji w całej sieci, tylko we własnej bazie danych. Jeśli znajdzie, wyświetli część danych zgromadzonych o stronach i ich adresy. Dlatego proces ten trwa zazwyczaj niecałą sekundę. Im sprawniejsze serwery, oprogramowanie i roboty indeksujące wyszukiwarki, tym trafniej i szybciej otrzymasz adresy stron, na których znajdziesz szukane informacje.

Jeden obiekt, np. wyraz, może być opatrzony wieloma indeksami. Na przykład słowo *handel* może, dzięki indeksom, prowadzić do informacji o handlu (kupnie i sprzedaży) lub Jerzym Fryderyku Händlu (czyt. hendlu) – wielkim kompozytorze.

Uruchom przeglądarkę internetową. Wczytaj stronę wyszukiwarki *google.pl* i wpisz słowo *handel*. W odpowiedzi otrzymasz ponad 100 milionów adresów stron (rys. 21.1.). Oczywiście to zbyt wiele, by mówić o skutecznym wyszukiwaniu. Programy wyszukiwarki pozycjonują strony w taki sposób, by na pierwszych miejscach pokazały się najczęściej odwiedzane adresy. Jak zwiększyć trafność wyszukiwania, dowiesz się w dalszej części rozdziału.



Rys. 21.1. Wyniki wyszukiwania przez Google słowa „handel"

metaopis ▸ **Metaopis** to krótka informacja o stronie ukazująca się pod adresem znalezionym przez wyszukiwarkę.

Zwróć uwagę na niepozorne – ale bardzo istotne z punktu widzenia wyszukiwania – menu Google. Wybierz z niego opcję **Grafika**.

Zauważ, że gdy aktywna była domyślna opcja **Wszystko**, wśród wyświetlonych adresów żaden nie dotyczył kompozytora Jerzego Fryderyka Händla. Jednak gdy wybierzesz opcję **Grafika**, sytuacja się odwróci (rys. 21.2.). Jest to logiczne, ponieważ o wiele częściej są wyszukiwane portrety kompozytora niż zdjęcia związane z handlem. Ilustracje o handlu zostały więc pogrupowane i dostępne po wybraniu odpowiedniej ikony.



Rys. 21.2. Wyniki wyszukiwania dla hasła „handel" w opcji **Grafika**

Użyłeś/Użyłaś domyślnej opcji **Wszystko** i opcji **Grafika**. Sprawdź, jakie treści zostały wyszukane w postaci filmu, wiadomości i map. Użyj także opcji **Więcej**.

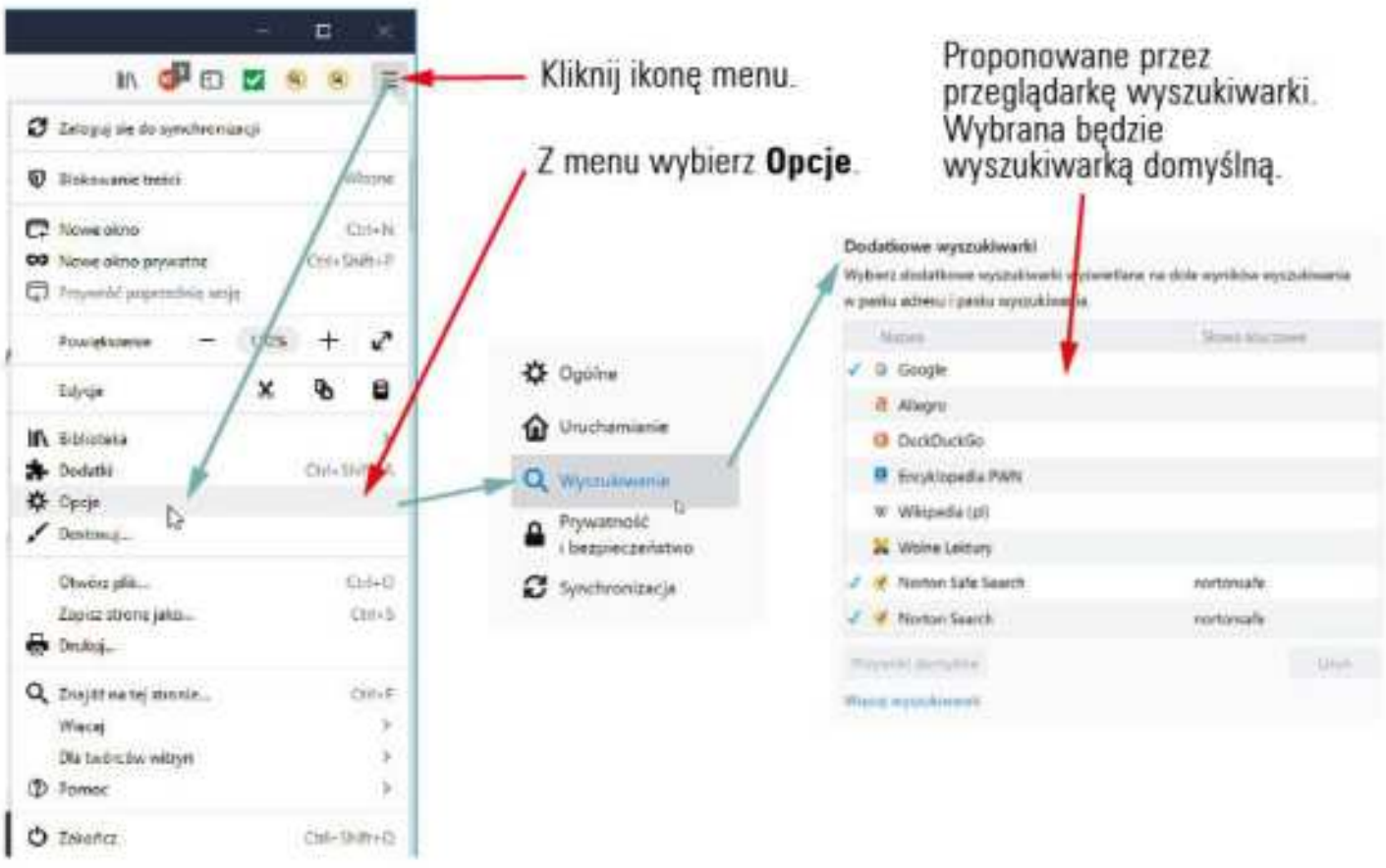
21.2. Wyszukiwarek jest wiele, czyli szukaj, a znajdziesz

Google jest bardzo popularną wyszukiwarką, lecz nie jedyną. Sprawdź, jakie proponuje twoja przeglądarka stron, np. Microsoft Edge (czyt. majkrosoft edż, rys. 21.3.) lub Firefox (czyt. fajerfoks, rys. 21.4.).

domyślna wyszukiwarka przeglądarki ▸ W większości przeglądarek możesz ustawić **wyszukiwarkę domyślną**. Można ją wybrać ze spisu zawierającego najpopularniejsze wyszukiwarki.



Rys. 21.3. Wybór domyślnej wyszukiwarki w przeglądarce Microsoft Edge

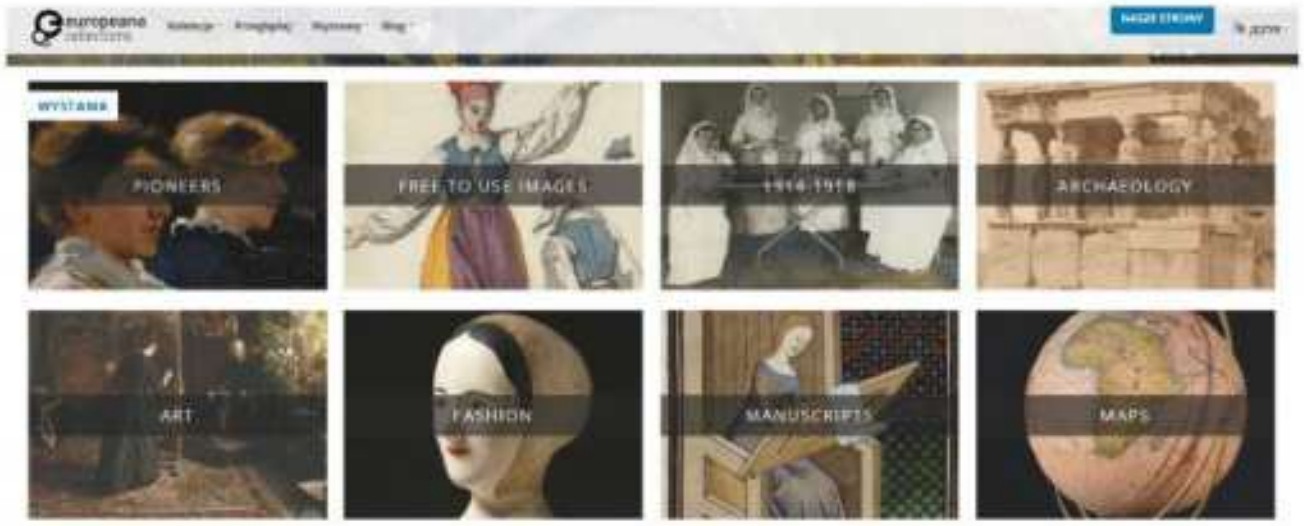


Rys. 21.4. Wybór domyślnej wyszukiwarki w przeglądarce Firefox

Wśród wyszukiwarek znajdziesz także takie, które nie są bardzo popularne, jednak pozwalają skutecznie znaleźć informacje z różnych dziedzin.

◂ Europeana

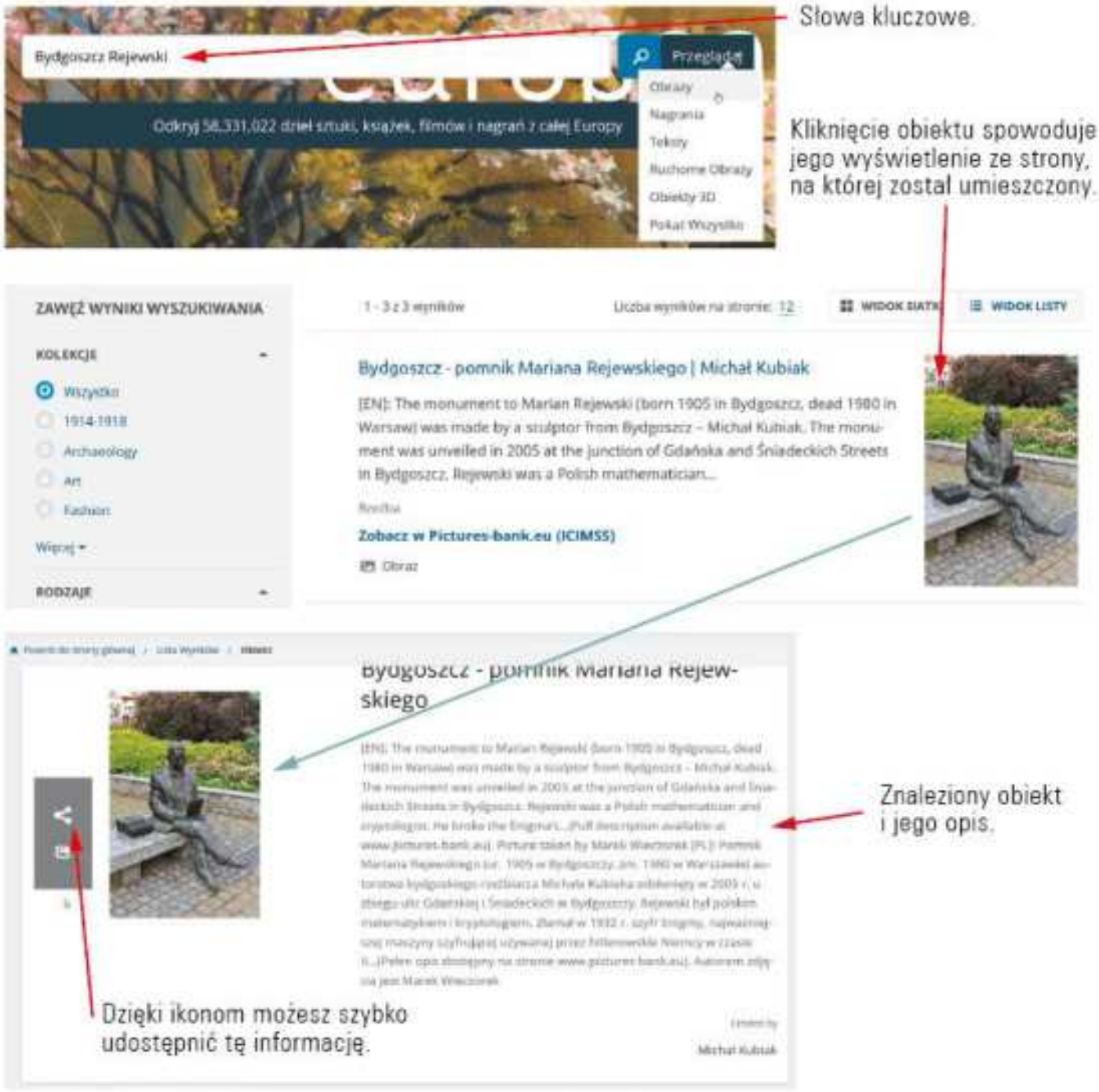
Jedną z nich jest *europeana.eu* (rys. 21.5.) dająca dostęp do zasobów bibliotek, muzeów, organizacji kulturalnych, a także zbiorów prywatnych z całej Europy. Obecnie zawiera ona bazę prawie 54 milionów dóbr kultury europejskiej. Mimo że nie jest typową wyszukiwarką, bogactwo zgromadzonych danych dostępowych do zbiorów i łatwość ich odnajdywania czyni ją nieocenionym źródłem informacji o kulturze europejskiej. Każdy z obiektów Europeany jest archiwizowany z ogromną dbałością o jakość. Zdjęcia i kopie cyfrowe wykonano w dużej rozdzielczości.



Rys. 21.5. Fragment strony *europeana.eu*

Wczytaj stronę *europeana.eu* i sprawdź, jakie obiekty kultury związane z twoim regionem lub miejscowością możesz znaleźć w indeksowanej bazie obiektów. Użyj odpowiednich **słów kluczowych**, dzięki którym wyszukiwarka będzie mogła dokładniej spełnić twoje oczekiwania.

W przykładzie (rys. 21.6.) szukano rzeźby przedstawiającej Mariana Rejewskiego.



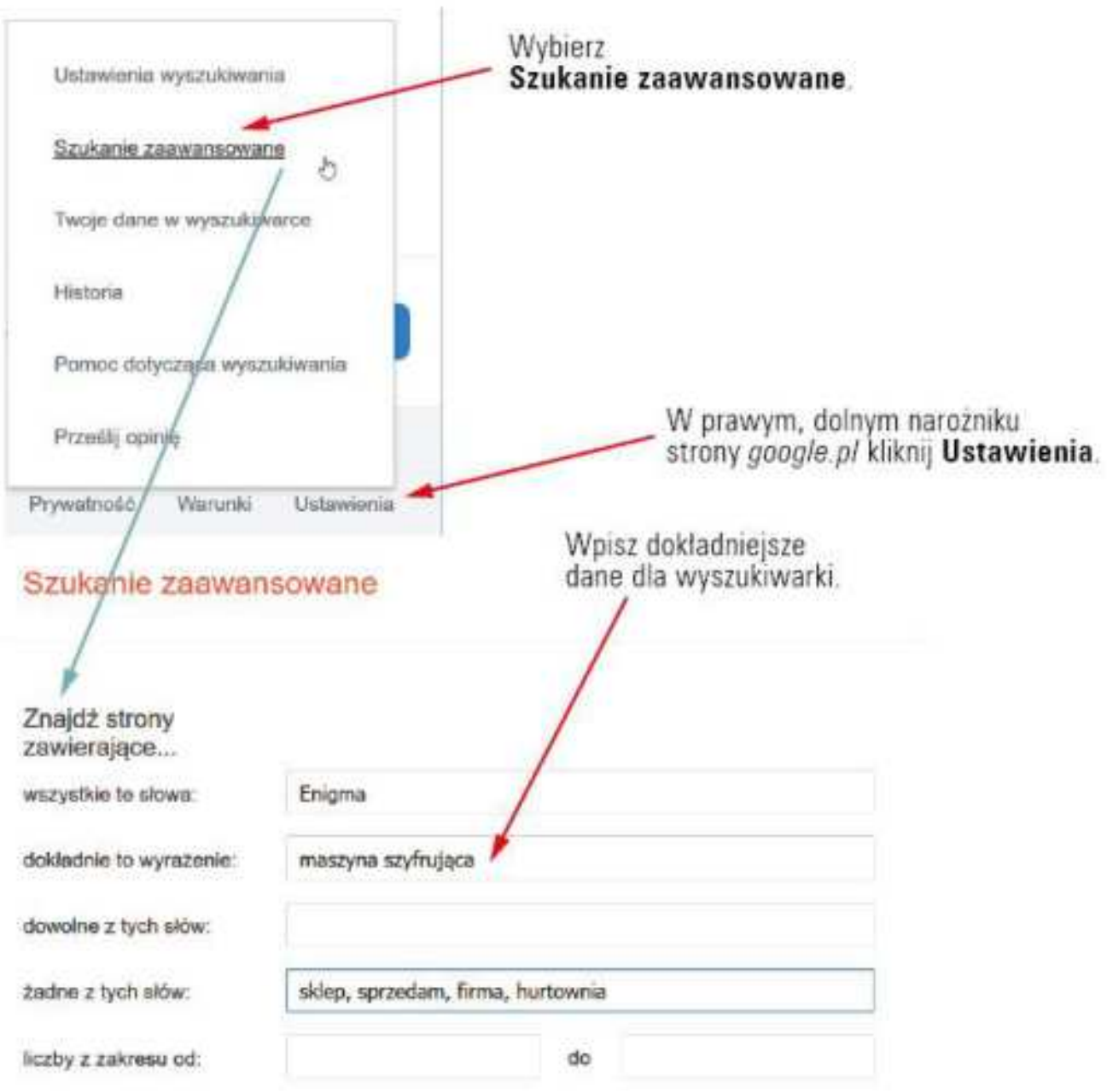
Rys. 21.6. Pomnik Mariana Rejewskiego ze zbiorów *europeana.euu*

Marian Rejewski to wybitny polski matematyk i kryptolog, który wraz z Henrykiem Zygalskim i Jerzym Różyckim złamał szyfr Enigmy – niemieckiej maszyny szyfrującej używanej przez faszystowskie Niemcy w czasie drugiej wojny światowej. Przyczyniło się to znacznie do pokonania III Rzeszy Niemieckiej i zwycięstwa aliantów.



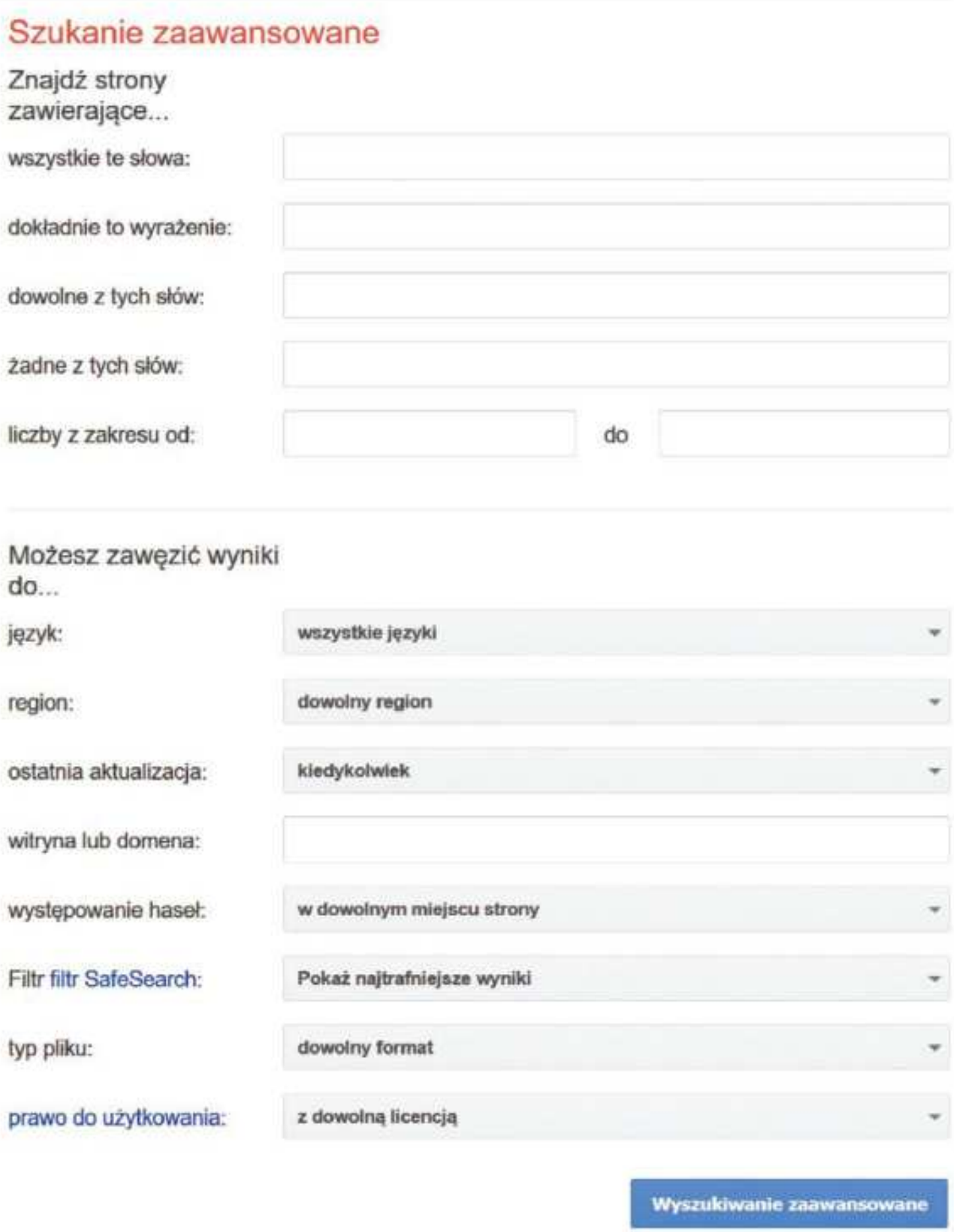
21.3. Tajemnice wyszukiwarek, czyli jak wyszukiwać dokładniej

Znasz już możliwości wyszukiwarek. Czy umiesz korzystać z ich wszystkich możliwości? Posługiwanie się zaawansowanymi opcjami przeglądarki znacznie ułatwi szybkie dotarcie do właściwych informacji i odrzucenie tych niepotrzebnych już w procesie wyszukiwania. Rozpatrzmy przykład *google.pl*. Poszukaj informacji o maszynie szyfrującej Enigma. Ponieważ wiele firm i sklepów przyjęło taką nazwę, odrzućmy ich strony już na etapie wyszukiwania (rys. 21.7.).



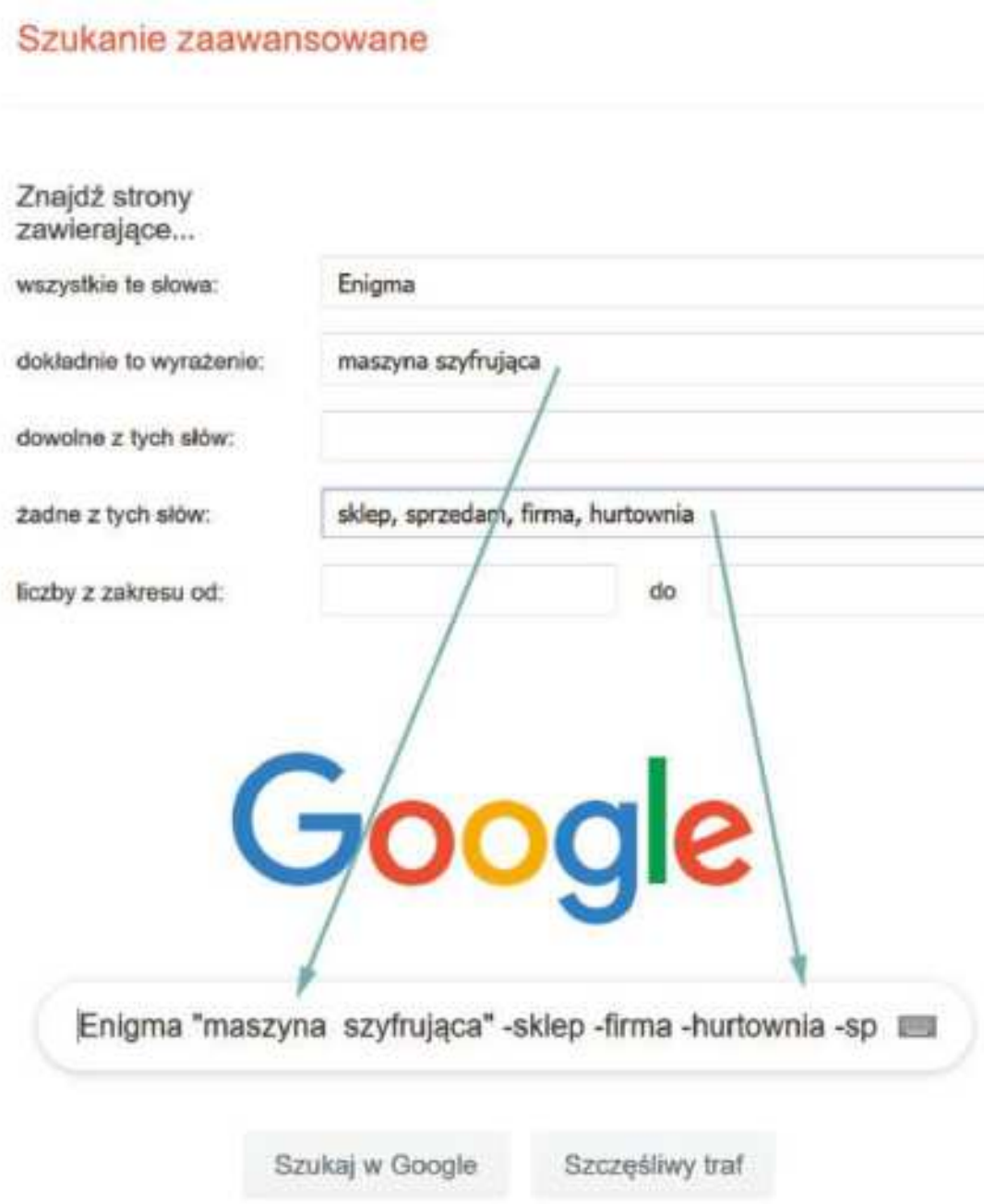
Rys. 21.7. Zaawansowane opcje wyszukiwania w *google.pl*

Opcja szukania zaawansowanego daje jeszcze kilka możliwości precyzowania parametrów wyszukiwania. Sprawdź samodzielnie ich działanie (rys. 21.8.).



Rys. 21.8. Opcje wyszukiwania zaawansowanego

Istnieje także inna droga do osiągnięcia podobnego efektu jak przy korzystaniu z opcji zaawansowanego szukania. Możesz użyć odpowiednich znaków już w polu wpisywania słów kluczowych (rys. 21.9.). Jeśli chcesz znaleźć strony zawierające pewne wyrażenie, możesz użyć cudzysłowu. W celu wyeliminowania stron zawierających określone słowa postaw przed nimi minus.



Rys. 21.9. Użycie znaków specjalnych w polu wpisywania słów kluczowych wyszukiwarki

Czy zdarzyło ci się, że musiałeś/musiałaś znaleźć informację o jakimś obiekcie, którego nazwy nie znałeś/znałaś? To trudne, ponieważ w takiej sytuacji nie można dobrać odpowiednich słów kluczowych. Wyszukiwarka Google kryje jeszcze jedną niespodziankę. Można wyszukiwać informacje, dysponując jedynie zdjęciem obiektu. Ta możliwość ma jeszcze inne zastosowania, np. można sprawdzić, na jakich stronach ukazało się dane zdjęcie, postać lub obiekt. Można więc wykryć kradzież twojej własności.

Załóżmy, że w swoim projekcie opisujesz polskie ptaki. Masz ich zdjęcia, ale niektórych nie zidentyfikowałeś/zidentyfikowałaś. Czy wyszukiwarka ci w tym pomoże (rys. 21.10.)?



Rys. 21.10. Wyszukiwanie obrazem w Google

21.4. Legalnie, czyli szukamy obiektów do publikacji

W czasie nauki w szkole masz okazję brać udział w różnych projektach. Często wymagają one tworzenia dokumentów, np. prezentacji na różne tematy. Zmusza to do poszukiwań tekstów, filmów, ilustracji, zdjęć i innych danych. Nie wszystkich możesz użyć w swojej publikacji ze względu na obowiązujące prawo autorskie. Wykorzystanie danego obiektu, np. zdjęcia, zależy od licencji, na zasadzie której zostało opublikowane.

Należy uwzględnić to już podczas wyszukiwania. Licencją pozwalającą na użycie dokumentu lub jego fragmentu we własnych opracowaniach i publikacjach (z poszanowaniem praw autorskich i uwarunkowań określonych przez autora) jest CC (**Creative Commons**). Zazwyczaj wystarcza podanie źródła i autora dzieła.

◂ Creative Commons

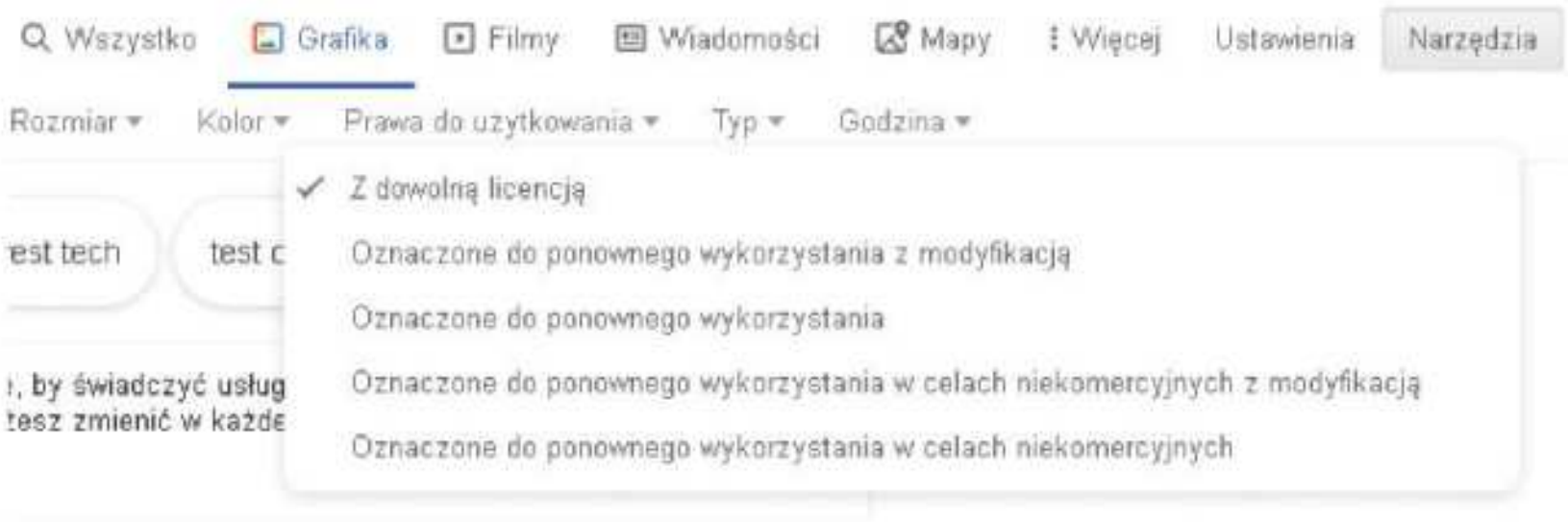
Uwaga!

Szczegółowe informacje na temat CC znajdziesz na stronie *creativecommons.pl*.

Jeśli znaczący dla projektu dokument (np. fotografia) został opublikowany z inną licencją, np. na portalu informacyjnym, konieczne jest uzyskanie zgody właściciela praw majątkowych.

Niektóre wyszukiwarki pozwalają na szybkie dotarcie do treści na licencji CC, np. Bing w połączeniu z OneDrive lub Office 365.

Możesz także użyć zaawansowanego wyszukiwania w Google (rys. 21.11.).



Rys. 21.11. Opcje wyszukiwania zaawansowanego Google dotyczące licencji szukanych materiałów

21.5. Dbamy o wiarygodność, czyli jak ocenić wartość źródła

Samo wyszukiwanie nie miałoby sensu, gdyby ze znalezionych informacji nie było pożytku. Jeśli robi się to w celu poszerzenia wiedzy lub użycia we własnych projektach, konieczne jest sprawdzenie wiarygodności źródła. Twoje opracowanie lub wiedza będzie bowiem tyle warta, ile źródła, z których pozyskałeś/pozyskałaś informacje. Pamiętaj o tym i nigdy nie opieraj swojej wiedzy na niesprawdzonych informacjach.

Postępuj według kilku zasad:

1. Wiarygodność informacji lub danych potwierdzaj, porównując kilka źródeł.
2. Nie ulegaj manipulacjom i teoriom spiskowym, np. o płaskiej Ziemi.
3. Weź pod uwagę, że większość informacji podawanych przez blogerów, vlogerów itp. ma niewiele wspólnego z rzetelną informacją. Najczęściej są wyrażeniem prywatnej opinii na dany temat.
4. Poznaj i sprawdź wiarygodność autora informacji.
5. Podpis autora lub zespołu redakcyjnego jest jednym z elementów uwiarygodnienia informacji.
6. Nigdy nie rozpowszechniaj niesprawdzonych informacji.
7. Wykorzystując znalezione informacje we własnych opracowaniach, podawaj ich źródła.

ZADANIA DO WYKONANIA

1. W wyszukiwaniu użyj słowa „Adelajda". Zanotuj, ile otrzymałeś/otrzymałaś odpowiedzi w opcji Wszystkie i jak długo trwało wyszukiwanie informacji. Zauważ, że większość znalezionych stron dotyczy albo imienia, albo miasta w Australii. Użyj dodatkowych słów kluczowych, by wyszukane dane dotyczyły miasta. Porównaj liczbę znalezionych stron z pierwszym wyszukiwaniem. Wytłumacz, dlaczego liczby znacznie się różnią.
2. W polu adresowym przeglądarki wpisz nazwę stolicy twojego województwa i wciśnij **Enter**. Na podstawie wyników wyszukiwania zidentyfikuj domyślną wyszukiwarkę. Zmień ją na inną i porównaj wyniki wyszukiwania.
3. Użyj wyszukiwarki domyślnej, by znaleźć informację na temat stolicy twojego województwa lub powiatu. Sprawdź, jakie informacje znalazła wyszukiwarka w poszczególnych grupach, np. grafikę, filmy itp. Zidentyfikuj obiekty na znalezionych zdjęciach.
4. Użyj zdjęcia z własnej lub szkolnej kolekcji do wyszukania informacji wyszukiwarką Google. Oceń trafność i skuteczność działania tej opcji. Jeśli operacja się nie powiodła, czyli wyszukiwarka błędnie zinterpretowała zdjęcie, podejmij próbę wytłumaczenia tego zjawiska, odpowiadając na pytania:

   Czy obiekt na zdjęciu jest charakterystyczny?
   Czy zdjęcie przedstawia wiele różnych obiektów?
   Czy zdjęcie przedstawia obiekty lub otoczenie występujące w wielu miejscach, np. łąka, las, rzeka, ulica nocą itp.?
5. Załóż, że musisz znaleźć ilustrację muzyczną do szkolnego przedstawienia na podstawie *Balladyny*. Ponieważ macie zamiar pokazywać tę sztukę poza szkołą, muzyka musi być oznaczona licencją CC. Znajdź odpowiednią ilustrację muzyczną dla dowolnej sceny zaczerpniętej z lektury, pamiętając o uszanowaniu praw autorskich kompozytora (licencja CC).

PODSUMOWANIE LEKCJI

- Wyszukiwarka nie przeszukuje internetu za każdym razem, gdy jej użyjesz. W swojej bazie danych zapisanej na serwerze przez **roboty indeksujące** (programy wyszukujące i segregujące informacje) przechowuje ona potrzebne ci informacje, takie jak krótki opis (metaopis) i adres strony, na której wystąpiła poszukiwana informacja. ◂ s. 135
- Im trafniej dobierzesz słowa kluczowe, czyli takie, które opisują szukany obiekt, tym trafniejsze otrzymasz odpowiedzi. ◂ s. 136
- Istnieje wiele wyszukiwarek, np. *google.pl* lub *bing.com*. Istnieją także wyszukiwarki gromadzące informacje o dokumentach z jednej dziedziny, np. ***europeana.eu***, w której znajdziesz zbiory kultury europejskiej.
- W przeglądarce można ustawić **wyszukiwarkę domyślną**. ◂ s. 136
- Dzięki opcjom wyszukiwania zaawansowanego można wyeliminować strony o niechcianej treści lub rodzaju. ◂ s. 137
- Niektóre wyszukiwarki pozwalają na wyszukiwanie obrazem.
- Darmowe zasoby, np. grafika, zdjęcia lub dźwięki, są udostępniane na zasadzie licencji **CC Creative Commons**. ◂ s. 143
- Nie wierz we wszystko, co znajdziesz w sieci. Sprawdzaj wiarygodność informacji. ◂ s. 143

22. Wirtualne serwery, czyli instalujemy Apache

NA TEJ LEKCJI:

- poznasz definicję serwera lokalnego;
- dowiesz się, do czego służy pakiet XAMPP;
- nauczysz się instalować i uruchamiać pakiet XAMPP.

Jeśli chcemy rozpocząć przygodę z tworzeniem stron internetowych za pomocą języków programowania interpretowanych po stronie serwera, z wykorzystaniem baz danych czy systemów zarządzania treścią (CMS), potrzebny nam będzie serwer lokalny.

Serwer lokalny, popularnie zwany localhostem (czyt. lokalhostem), wykonuje pracę oprogramowania sieciowego na komputerach niepodłączonych fizycznie do sieci. Użytkownik może wykonać na nim program lub projekt, przetestować go, a następnie wersję finalną wgrać na serwerze podłączonym do sieci. ◂ serwer lokalny

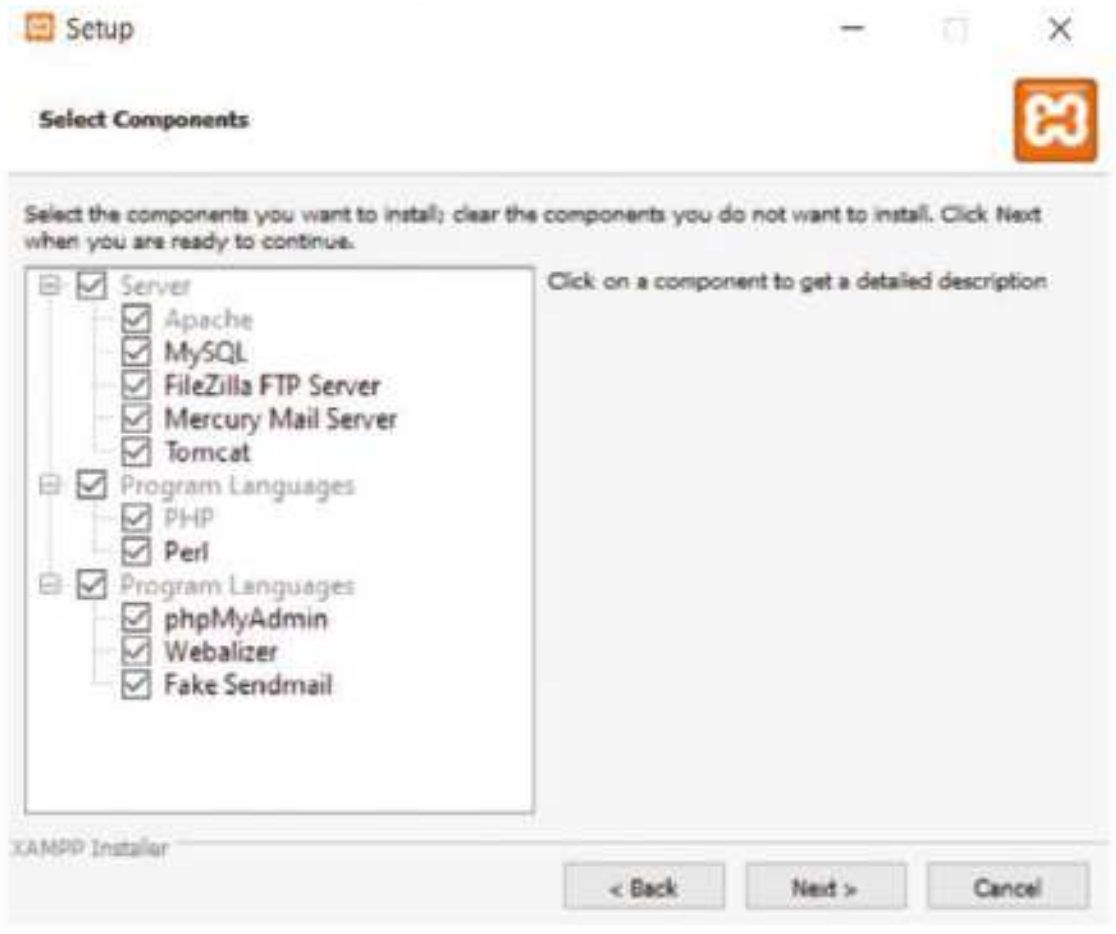
22.1. XAMPP – Apache, MySQL, PHP, Perl

XAMPP ▸ Zaczynając pracę z lokalnym serwerem, należałoby rozszyfrować jego nazwę. Co to jest XAMPP? XAMPP jest najpopularniejszym wieloformatowym, zintegrowanym serwerem WWW, który oprócz serwera Apache (czyt. apaczi) i bazy danych MySQL zawiera również interpretery skryptów napisanych w językach PHP i Perl. Nazwa jest akronimem: X – z języka angielskiego *cross*, w naszym wypadku *cross-platform*, czyli wieloplatformowe, A – od Apache, serwera WWW, M – od bazy danych MySQL, a także dwie litery P od dwóch interpreterów – języka PHP i języka Perl.

22.2. Instalacja XAMPP

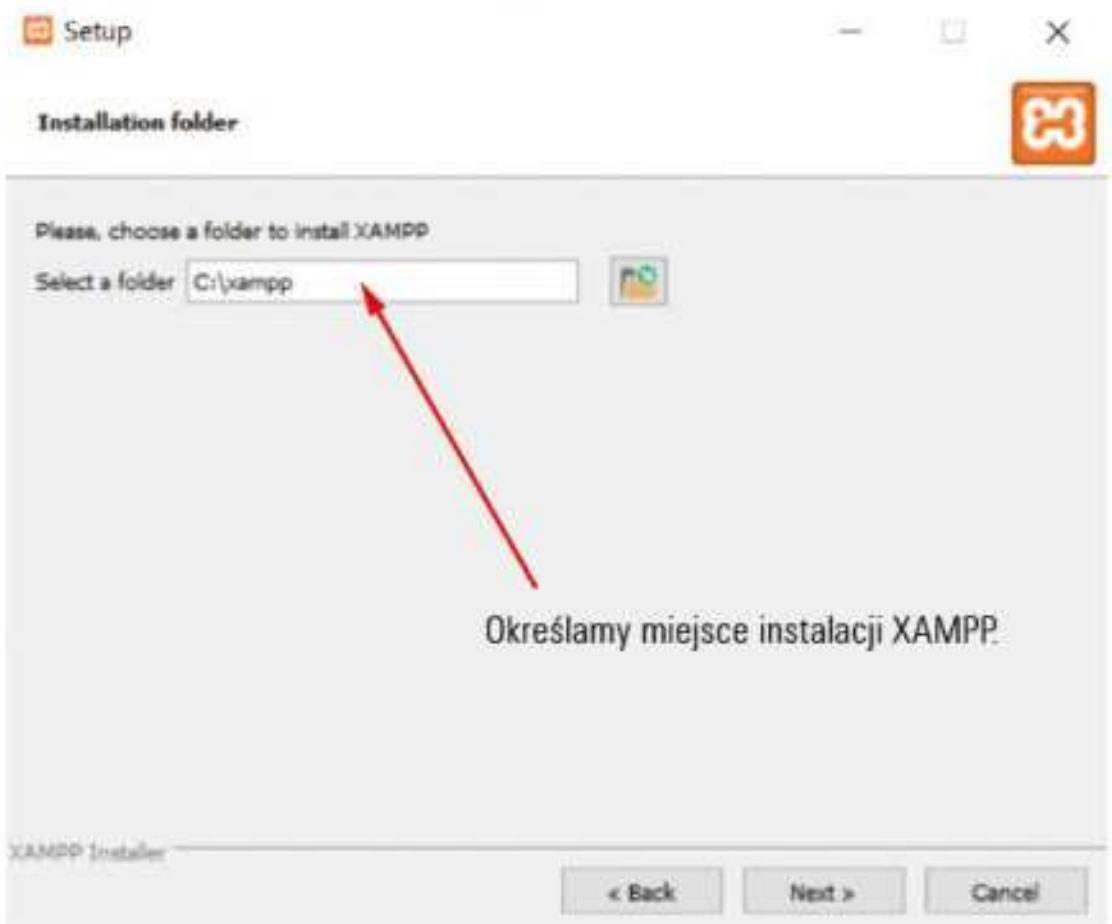
Instalacja XAMPP jest bardzo prosta. Cała paczka została tak skonstruowana, aby nie wymagała od użytkownika zbyt wielu konfiguracji.
Najnowszą wersję pobieramy ze strony organizacji, która ją udostępnia: *www.apachefriends.org*.
Pakiet jest wydawany na licencji GNU General Public License (czyt. dżeneral pablik lajsens) jako darmowy serwer WWW do obsługi dynamicznych stron.
Obecnie XAMPP jest dostępny dla trzech platform: Windowsa, Linuxa i OS X.
My zainstalujemy ten serwer dla systemu Windows. Wersja, którą będziemy instalować, to 7.3.11. (PHP 7.3.11.).

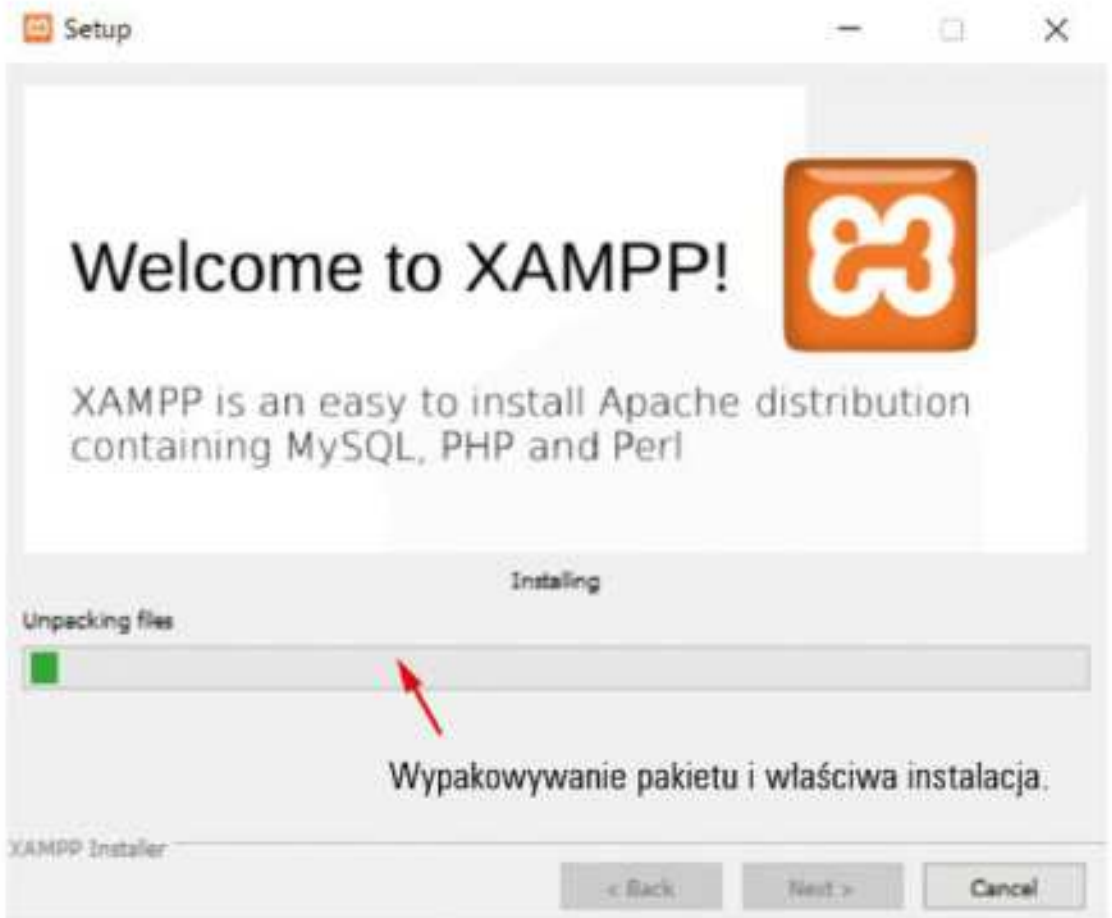
- Krok 1. Pobieramy paczkę ze strony *www.apachefriends.org*. Zapisujemy ją w dogodnej dla nas lokalizacji na naszym komputerze.
- Krok 2. Uruchamiamy program instalacyjny z miejsca, w którym zapisaliśmy naszą paczkę. W pierwszym oknie pojawi się możliwość wyboru komponentów, które możemy zainstalować (rys. 22.1.).

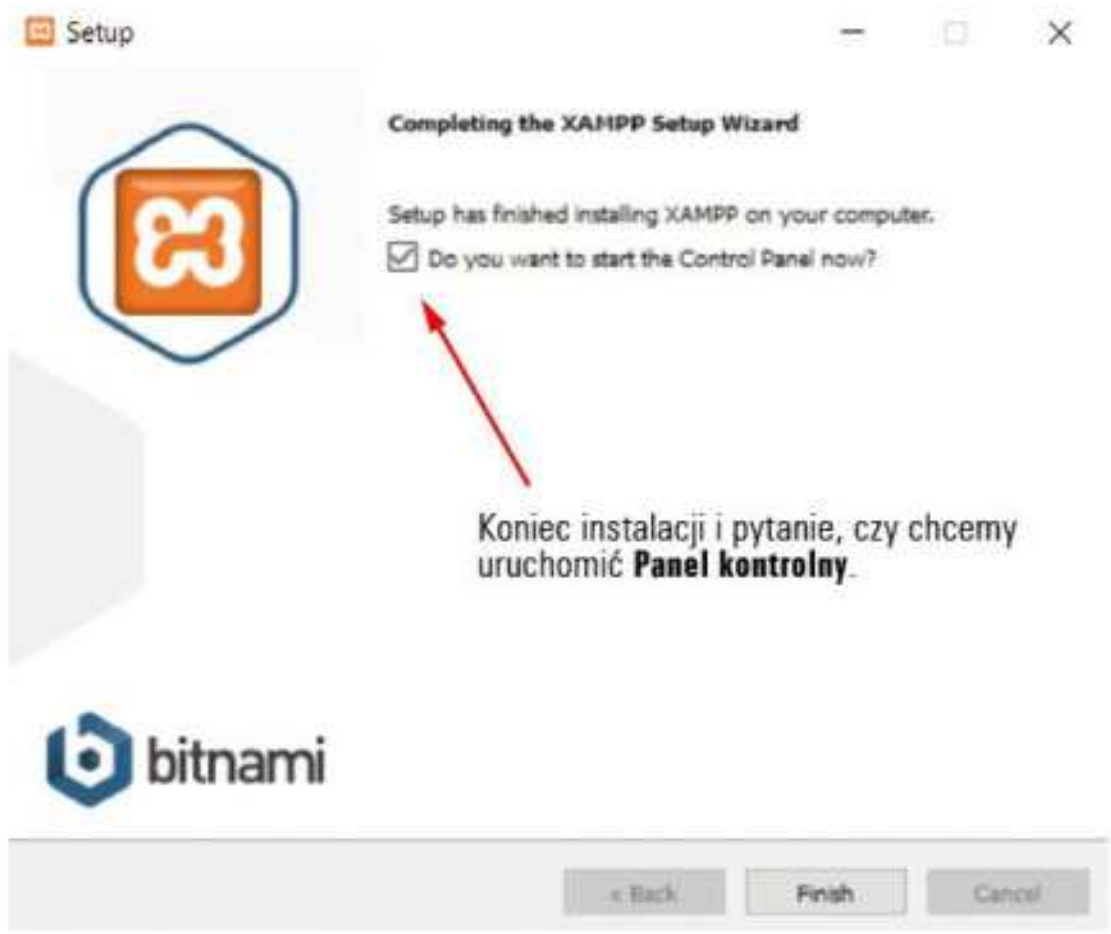


Rys. 22.1. Wybór komponentów do zainstalowania

- Krok 3. Instalacja pakietu (rys. 22.2.) Określamy folder docelowy, gdzie będzie zainstalowany XAMPP. Wówczas rozpocznie się proces rozpakowywania i właściwej instalacji. Po jego zakończeniu program instalacyjny zapyta nas, czy chcemy uruchomić **Panel kontrolny** (ang. **Control Panel**).

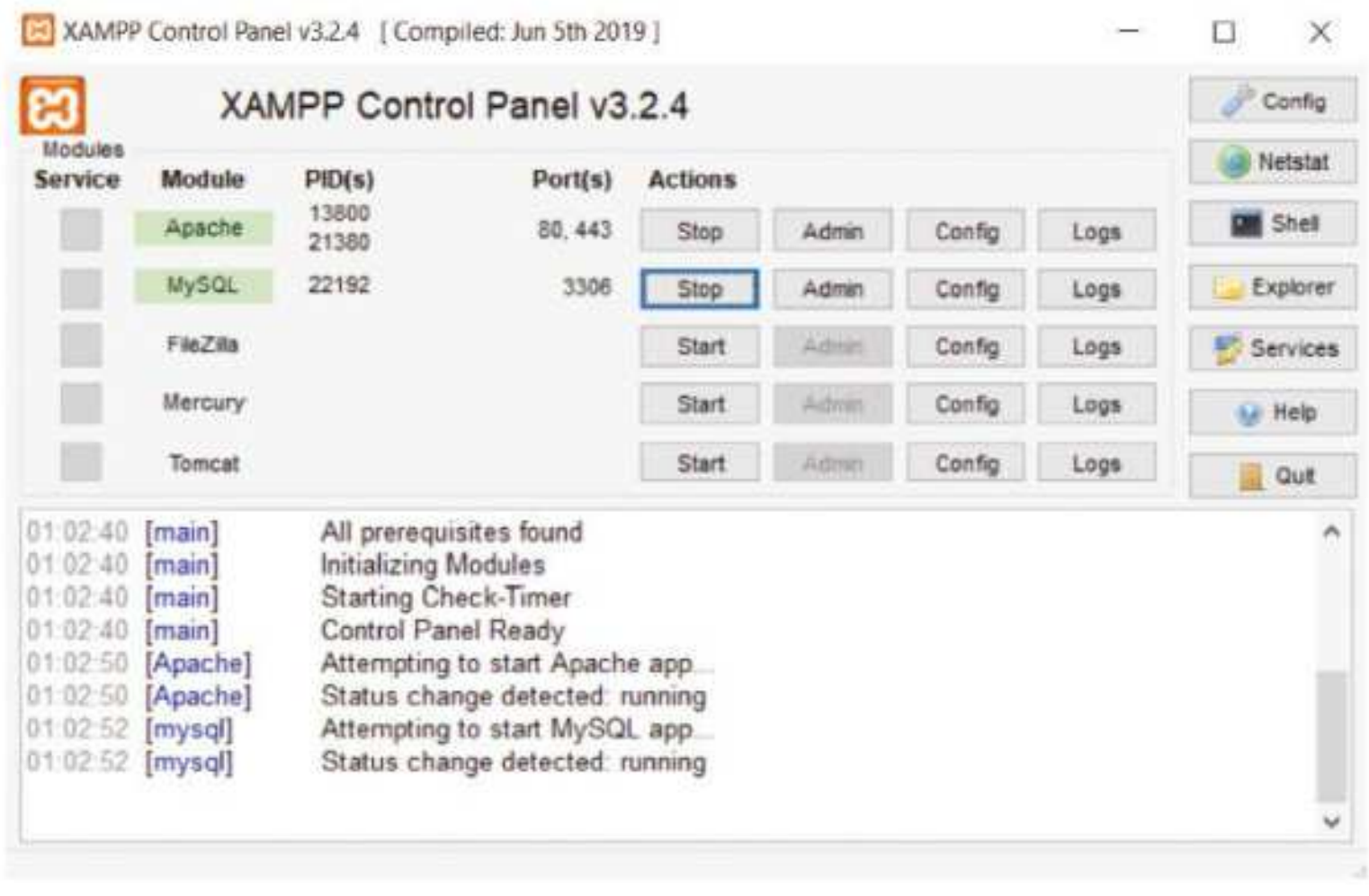






Rys. 22.2. Proces instalacji pakietu XAMPP

- Krok 4. Uruchamiamy panel oraz wciskamy **Start** przy Apache i MySQL, aby sprawdzić, czy instalacja przebiegła poprawnie. Podświetlenie napisów na zielono oznacza, że tak jest i że możemy przystąpić do pracy (rys. 22.3.).



Rys. 22.3. Panel kontrolny XAMPP

- Krok 5. W pasku adresu naszej przeglądarki wpisujemy *http://loclahost*, aby uruchomić serwer.



Rys. 22.4. Strona startowa XAMPP

ZADANIA DO WYKONANIA

1. Wyszukaj w internecie informacje o najnowszej wersji PHP dla systemu operacyjnego twojego komputera.
2. Wyszukaj w internecie informacje o najnowszej wersji Apache (lub innego serwera WWW) dla systemu operacyjnego twojego serwera
3. Wyszukaj w internecie informacje o najnowszej wersji MySQL (lub innej bazy danych) dla systemu operacyjnego twojego serwera.

PODSUMOWANIE LEKCJI

- Aby tworzyć strony WWW oparte na systemach zarządzania treścią lub wykorzystujące języki programowania interpretowane po stronie serwera, potrzebny jest serwer lokalny. ◂ s. 145
- Przykładem serwera lokalnego, który umożliwia instalację stron opartych na systemach zarządzania treścią i pracy z bazą danych, jest Apache, który wchodzi w skład pakietu XAMPP. ◂ s. 146

23. Język zapytań, czyli poznajemy język SQL

NA TEJ LEKCJI:

- poznasz definicję języka SQL;
- dowiesz się, jaką składnię ma język SQL;
- poznasz typy danych stosowanych w języku SQL.

23.1. Strukturalny język zapytań SQL

SQL ▸ Strukturalny język zapytań (**SQL**), używany do tworzenia i modyfikowania baz danych oraz umieszczania i pobierania danych z baz, został opracowany w latach 70. XX wieku w firmie IBM. Stał się standardem w komunikacji z serwerami relacyjnych baz danych. Pierwszą firmą, która włączyła SQL do swoich komercyjnych produktów, była Oracle.

Język SQL jest językiem deklaratywnym, w którym użytkownik opisuje warunki, jakie musi spełnić końcowe rozwiązanie, a nie szczegółowe kroki, które do niego doprowadzą.

W SQL decyzję o tym, w jaki sposób są pobierane i przechowywane dane, pozostawia się systemowi zarządzania bazą danych – DBMS.

23.2. Składnia SQL

składnia SQL ▸ Wyrażenia SQL – kwerendy – pozwalają nam na wykonywanie operacji na bazach i zawartych w nich danych.

Praca z językiem SQL polega na zadawaniu zapytań bazie danych. Można je zaliczyć do czterech podzbiorów. Są to:

- SQL DML – język manipulacji danymi,
- SQL DDL – język definicji danych,
- SQL DCL – język kontroli nad danymi,
- SQL DQL – język definiowania zapytań.

DML służy do wykonywania operacji na danych – do ich umieszczania w bazie, kasowania, przeglądania oraz dokonywania zmian. Najważniejsze polecenia z tego zbioru to:

- **INSERT** – umieszczenie danych w bazie,
- **UPDATE** – zmiana danych,
- **DELETE** – usunięcie danych z bazy.

Dane tekstowe muszą być zawsze ujęte w znaki pojedynczego cudzysłowu (').

DDL pozwala operować na strukturach, w których dane są przechowywane, czyli np. dodawać, zmieniać i kasować tabele lub bazy. Najważniejsze polecenia tej grupy to:

- **CREATE** (np. **CREATE TABLE**, **CREATE DATABASE**, ...) – utworzenie struktury (bazy, tabeli, indeksu itp.),
- **DROP** (np. **DROP TABLE**, **DROP DATABASE**, ...) – usunięcie struktury,
- **ALTER** (np. **ALTER TABLE ADD COLUMN**, ...) – zmiana struktury (dodanie kolumny do tabeli, zmiana typu danych w kolumnie tabeli).

DCL ma zastosowanie do nadawania uprawnień do obiektów bazodanowych. Najważniejsze polecenia w tej grupie to:

- **GRANT** – nadawanie uprawnień do pojedynczych obiektów lub globalnie konkretnemu użytkownikowi (np. **GRANT ALL PRIVILEGES ON EMPLOYEE TO PIOTR WITH GRANT OPTION** – przyznanie wszystkich praw do tabeli EMPLOYEE użytkownikowi PIOTR z opcją pozwalającą mu nadawać prawa do tej tabeli),
- **REVOKE** – odbieranie wskazanych uprawnień konkretnemu użytkownikowi (np. **REVOKE ALL PRIVILEGES ON EMPLOYEE FROM PIOTR** – odebranie użytkownikowi PIOTR wszystkich praw do tabeli EMPLOYEE),
- **DENY** – zabranianie wykonywania operacji.

DQL to język formułowania zapytań do bazy danych. W zakres tego języka wchodzi jedno polecenie – **SELECT**. Często **SELECT** traktuje się jako część języka DML, ale to podejście nie wydaje się właściwe, ponieważ DML z definicji służy do manipulowania danymi – do ich tworzenia, usuwania i uaktualniania. Na pograniczu obu języków znajduje się polecenie **SELECT INTO** (czyt. selekt intu), które dodatkowo modyfikuje (przepisuje, tworzy) dane.

Instrukcje w SQL powinno się zapisywać wielkimi literami, ale nie stanowi to wymogu.

Aby zbudować poprawną instrukcję w języku SQL, musimy ściśle przestrzegać pewnych reguł:

- każda instrukcja zaczyna się od słowa kluczowego (polecenia) określającego operację, która ma być wykonana,
- po słowie kluczowym powinny się znaleźć klauzule, czyli dopowiedzenia słowa kluczowego; musi być zachowana właściwa kolejność klauzul,
- każda instrukcja musi się kończyć średnikiem.

Słowo kluczowe jest zastrzeżone i ma przypisane specjalne znaczenie – nie można używać go jako identyfikatora.

Podczas tworzenia tabeli za pomocą języka SQL definiujemy typy atrybutów, jakie mogą przyjąć dane w konkretnej kolumnie. Zbiór wartości, które są dopuszczalne dla atrybutu, nazywamy domenami. W jednej tabeli atrybuty mogą mieć rożne domeny lub mogą mieć wspólną domenę. Domeną może być np. tekst, liczba, data itp.

Literały (stałe dosłowne) to ciąg znaków pomiędzy znakami cudzysłowu, które są przytaczane dosłownie.

Do przechowywania danych w języku SQL służą tablice.

Obiekty bazy danych (baza, tabela, kolumna) muszą mieć niepowtarzalny identyfikator zgodny z językiem SQL. Nazwy identyfikatora nie mogą zawierać spacji.

23.3. Typy danych w języku SQL

W języku SQL typy danych służą do określania, jakiego rodzaju będą kolumny tabeli. W różnych systemach baz danych będziemy mieli do czynienia z różnymi typami danych – wynika to z implementacji (czyli przystosowania do pracy) języka SQL. Zapoznaj się z tabelą 23.1.

Numeryczne	Data/Czas	Tekstowe
BOOL (z zakresu true i false)	DATE (przechowywanie daty)	CHAR (łańcuchy tekstowe)
INT (liczby całkowite)	TIME (przechowywanie czasu)	VARCHAR (łańcuchy tekstowe)
FLOAT, DOUBLE (liczby zmiennoprzecinkowe)		

Tab. 23.1. Niektóre typy danych wykorzystywanych w MySQL

ZADANIA DO WYKONANIA

1. Jakie są podstawowe cechy języka SQL?
2. Jakiego typu działania są realizowane za pomocą języka DDL?
3. Jakiego typu działania są realizowane za pomocą języka DML?

PODSUMOWANIE LEKCJI

s. 150 ▸

- SQL jest strukturalnym językiem zapytań powstałym w latach 70. XX wieku w firmie IBM. Używa się go do tworzenia i modyfikowania baz danych oraz umieszczania i pobierania danych z baz.

s. 150 ▸

- Praca z językiem SQL polega na zadawaniu zapytań bazie danych. Można je zaliczyć do czterech podzbiorów. Są to:
  - SQL DML – język manipulacji danymi,
  - SQL DDL – język definicji danych,
  - SQL DCL – język kontroli nad danymi,
  - SQL DQL – język definiowania zapytań.

24. Zadajemy pytania, czyli kwerendy w języku SQL

NA TEJ LEKCJI:

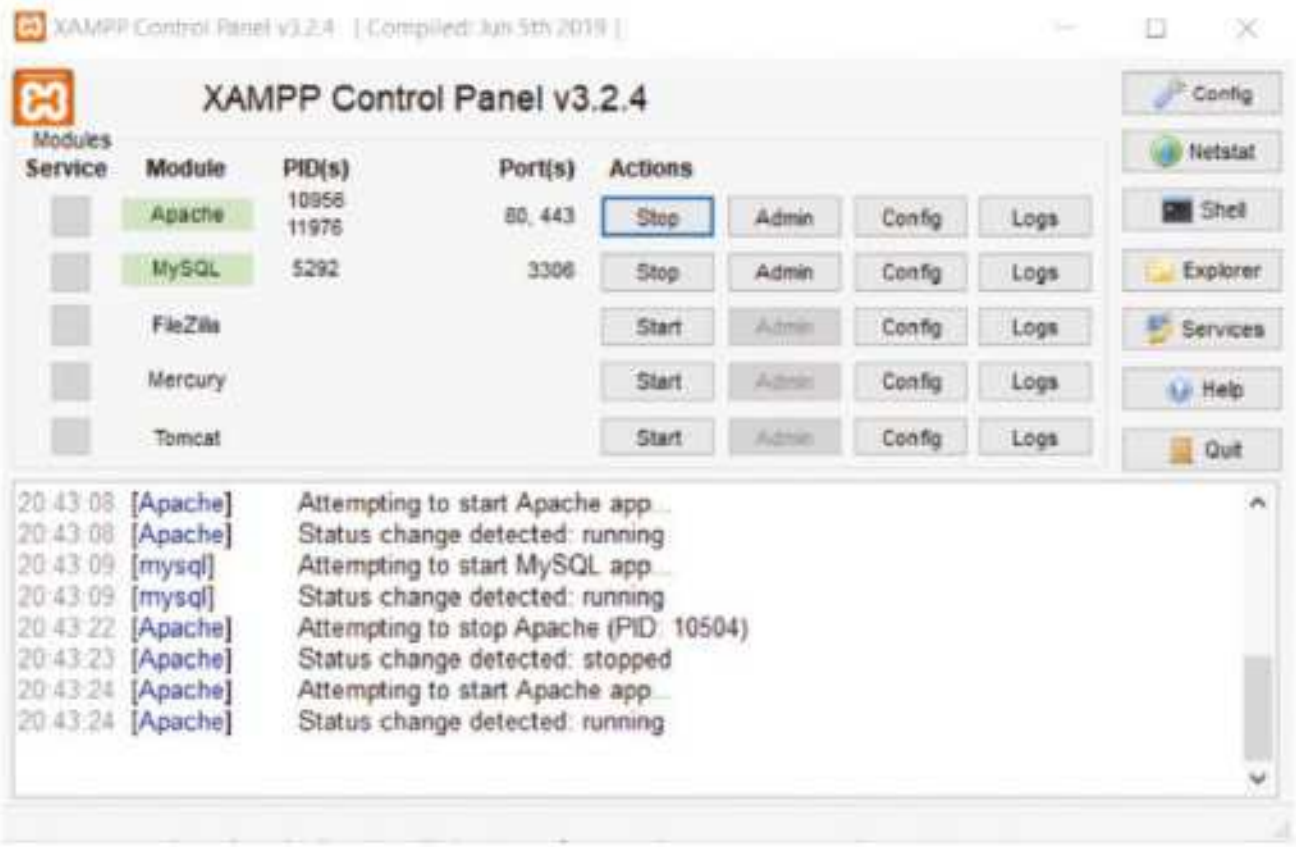
- poznasz definicję MySQL;
- dowiesz się, jak korzystać z phpMyAdmin w pakiecie XAMPP;
- nauczysz się tworzyć bazę danych z wykorzystaniem phpMyAdmin;
- dowiesz się, jak tworzyć tabele w bazie danych z wykorzystaniem języka SQL;
- poznasz zasady tworzenia zapytań w języku SQL.

Aby poznać tajniki języka SQL, musimy mieć środowisko pracy. Wiemy już, jak zainstalować pakiet XAMPP, który zawiera narzędzie do pracy z językiem SQL – **phpMyAdmin**.

24.1. Pierwsze kroki z MySQL, czyli tworzenie nowej bazy

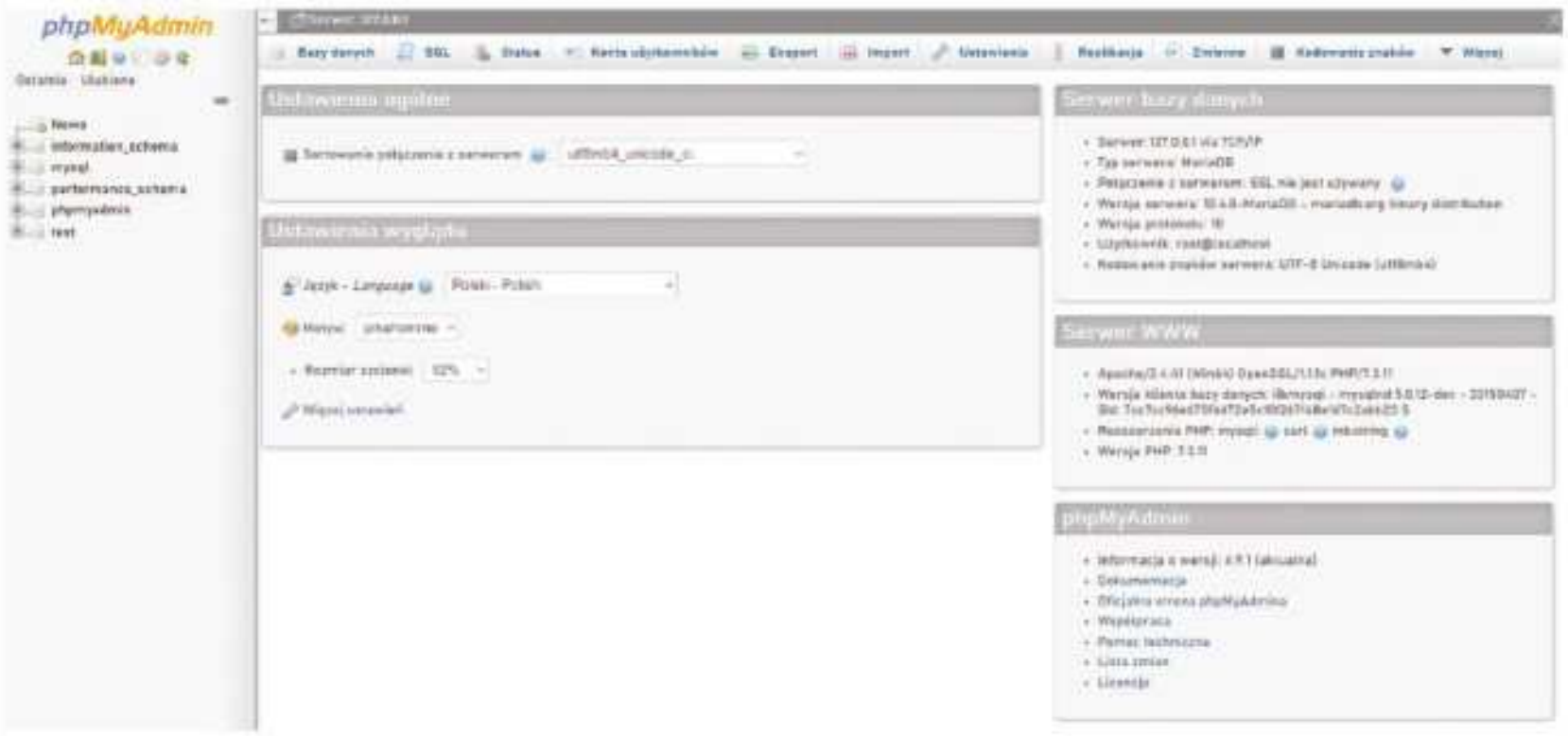
◂ MySQL

MySQL jest systemem zarządzania relacyjnymi bazami danych, wykorzystującym język SQL. Jest to rodzaj oprogramowania komputerowego, którego kod źródłowy jest wydawany na podstawie licencji otwartego oprogramowania (właściciel praw autorskich przyznaje użytkownikom prawa do badania, zmiany i rozpowszechniania oprogramowania w ramach tej licencji). Aby rozpocząć pracę z językiem SQL, należy uruchomić narzędzie phpMyAdmin, które zostało zainstalowane z pakietem XAMPP. W tym celu uruchamiamy najpierw **Panel kontrolny** (ang. **Control Panel**) XAMPP, a następnie moduły Apache i MySQL (rys. 24.1.).



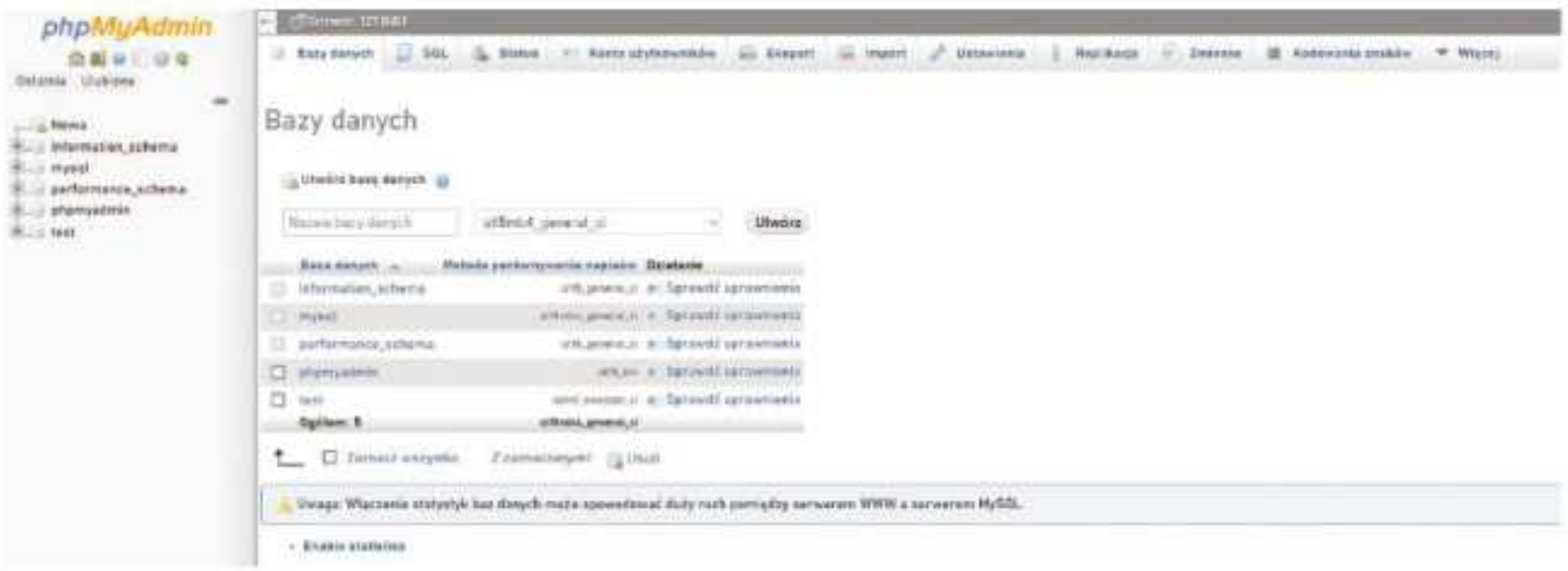
Rys. 24.1. Uruchomienie **Panelu kontrolnego** XAMPP oraz zainicjowanie pracy modułów Apache i MySQL

Następnie w przeglądarce internetowej w pasku adresu wpisujemy: *http://localhost/phpmyadmin*, aby uruchomić narzędzie phpMyAdmin (rys. 24.2.).



Rys. 24.2. Uruchomione narzędzie phpMyAdmin

Przechodzimy do zakładki **Bazy danych**, aby stworzyć naszą pierwszą bazę (rys. 24.3.).



Rys. 24.3. Tworzenie nowej bazy danych

Następnie uzupełniamy formularz, aby stworzyć bazę. Po lewej stronie formularza wpisujemy nazwę **baza1**, a po prawej określamy kodowanie oraz metodę porównywania napisów – ustawiamy ją na **utf8_polish_ci** (tę drugą cechę zawsze możemy zmienić). Klikamy przycisk **Utwórz** (rys. 24.4.).



Rys. 24.4. Wstawianie nazwy i metody porównywania napisów nowej bazy danych

Po odświeżeniu przeglądarki pojawi się formularz tworzenia pierwszej tabeli. Wypełniamy go, wpisując w pole **Nazwa**: **imię**, a w pole **Liczba kolumn**: **3.** Następnie klikamy **Wykonaj** (rys. 24.5.).



Rys. 24.5. Tworzenie tabeli

Pojawi się okno, w którym możemy zdefiniować nasze kolumny.

W pierwszym wierszu wpisujemy kolejno:

- Nazwa: id,
- Typ: INT,
- Długość/Wartość: 11,
- Index: PRIMARY (ponieważ nasze id będzie kluczem głównym).

W drugim wierszu:

- Nazwa: imie,
- Typ: VARCHAR,
- Długość/Wartość: 70.

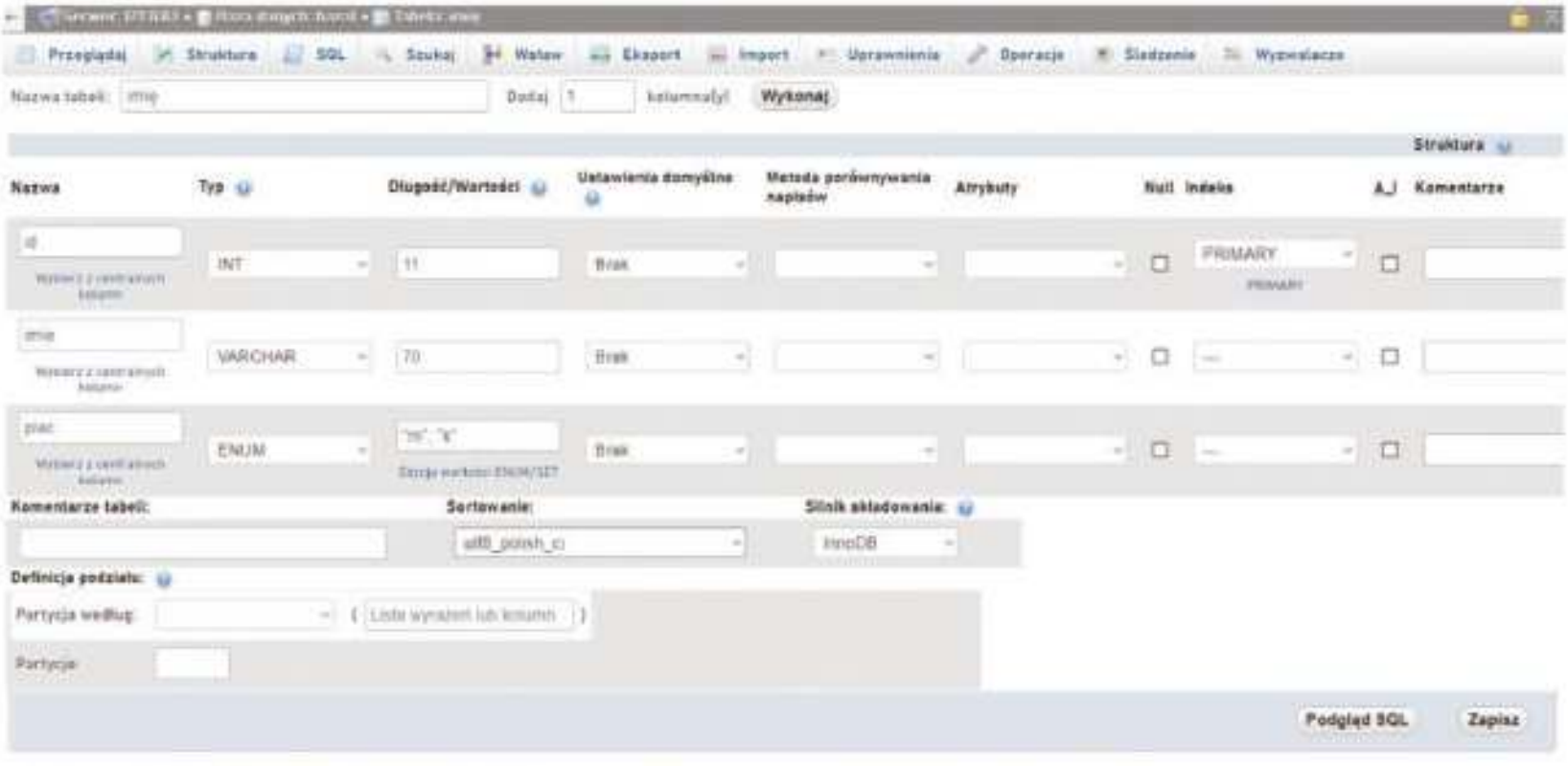
W trzecim wierszu:

- Nazwa: plec,
- Typ: ENUM,
- Długość/Wartość: „m", „k".

Dodatkowo w polu **Sortowanie** ustawiamy **utf8_polish_ci**.

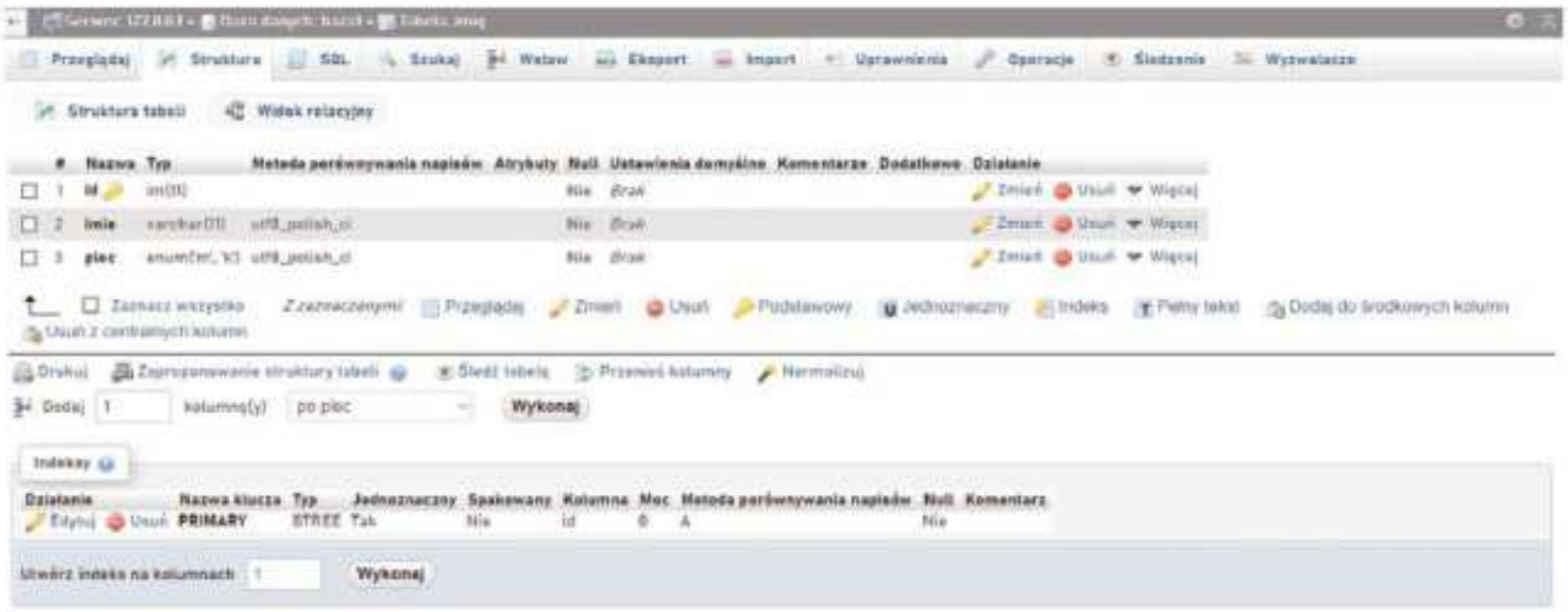
Klikamy **Zapisz**.

W polach **Nazwa** nie używamy polskich znaków diakrytycznych (rys. 24.6.).



Rys. 24.6. Definiowanie kolumn tabeli

Kiedy utworzymy tabelę, zostaniemy przeniesieni do jej podglądu (rys. 24.7.).



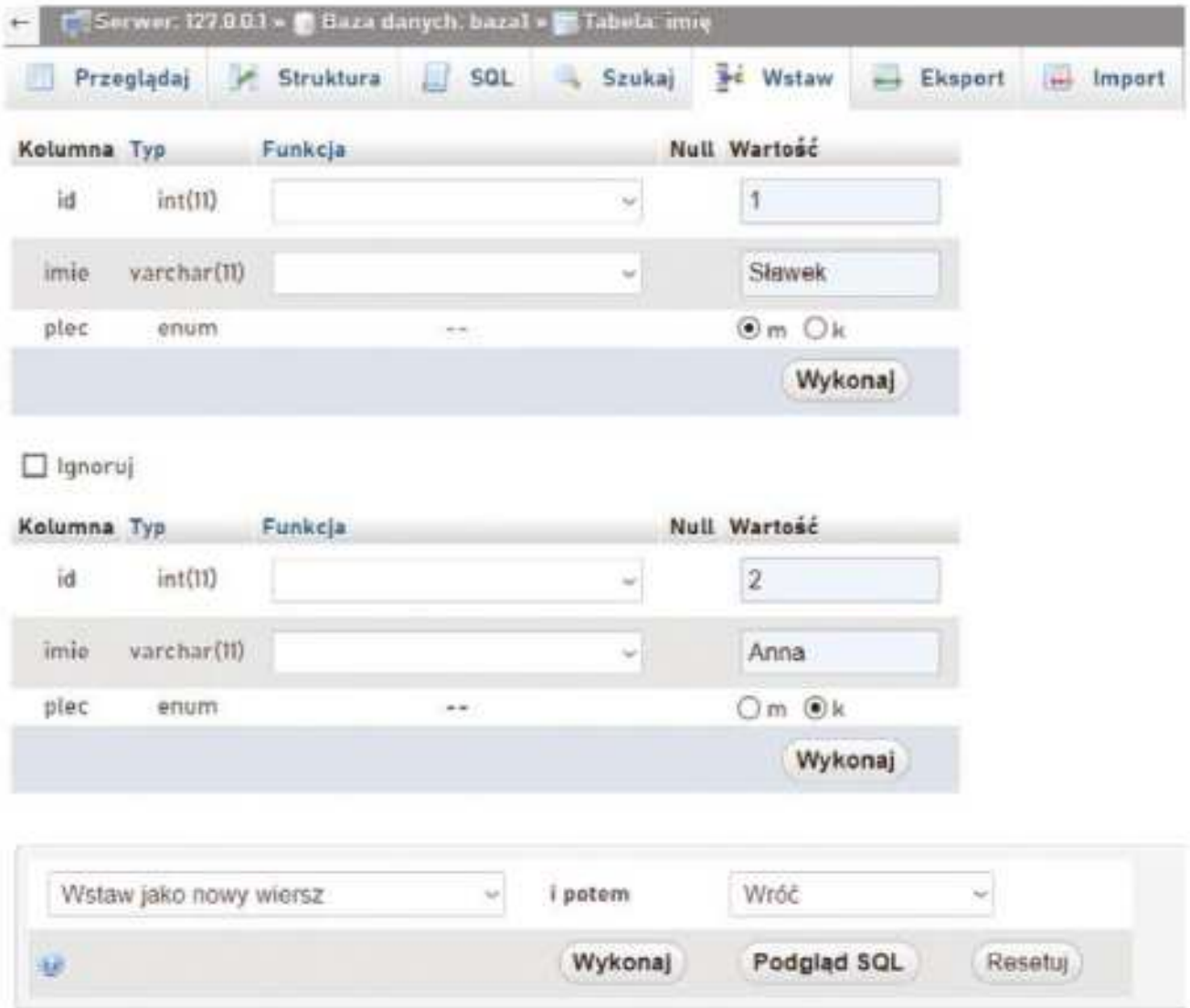
Rys. 24.7. Utworzona tabela w MySQL

Tworząc tabelę, użyliśmy kilku typów danych. Są one podobne do tych, których używa się w MS Access. W poniższej tabeli zawarto wyjaśnienie, jakie typy danych zostały użyte (tab. 24.1.).

INT	4-bitowa wartość całkowita. Zakres wartości ze znakiem od −2 147 483 648 do 2 147 483 647 lub bez znaku od 0 do 4 294 967 295.
ENUM	Kolumna łańcuchowa, która może zawierać jedną z wartości wymienionych na liście wartości w definicji tej kolumny.
VARCHAR	Pole znakowe o zmiennej długości z zakresu od 1 do 255 bajtów. Zajmowany jest jedynie taki obszar pamięci, jakiego wymaga wartość wstawionego pola.
PRIMARY KEY	Określa klucz główny tabeli.
AUTO_INCREMENT	Kolumna automatycznie dodaje +1 do poprzedniej wartości. Nie podajemy w zapytaniu SQL Insert wartości, zostawiając to pole puste. Baza danych automatycznie je uzupełni.
NOT NULL	Wartość nie może być pusta. Należy tutaj uważać na znaczenie słowa NULL. NULL oznacza brak wartości i nie jest jednoznaczna z wartością 0 lub białym znakiem.

Tab. 24.1. Typy danych użytych w tabeli **imię**

Skoro mamy już stworzoną tabelę w bazie danych, należałoby uzupełnić ją rekordami. Do tego posłuży nam zakładka **Wstaw** w phpMyAdmin (rys. 24.8.).



Rys. 24.8. Wstawianie rekordów do tabeli **imię**

24.2. Tworzenie tabel za pomocą języka SQL

W phpMyAdmin możemy również tworzyć tabele z wykorzystaniem języka SQL. W tym celu klikamy zakładkę **SQL** i tam wpisujemy kod. Do tworzenia tabel służy polecenie **CREATE TABLE**, które ma postać:

◂ CREATE TABLE

```
CREATE TABLE nazwa_tabeli
(
nazwa_kolumny_1 typ_kolumny_1 [atrybuty],
nazwa_kolumny_2 typ_kolumny_2 [atrybuty],
...
nazwa_kolumny_n typ_kolumny_n [atrybuty],
)
```

Podczas tworzenia tabel powinniśmy pamiętać o tym, aby:

- każda z tabel miała swoją unikatową nazwę,
- każda z kolumn w tabeli również miała swoją unikatową nazwę (nie stosujemy tu polskich znaków diakrytycznych),
- każda kolumna w tabeli miała zdefiniowany typ,

- każda kolumna typu znakowego miała podaną maksymalną długość,
- każda kolumna po stworzeniu była pusta.

Przykład

Stwórzmy bazę danych **księgarnia**, a w niej tabelę **klient** z kolumnami jak w tabeli 24.2.

Nazwa	Typ	Rozmiar	Wymagana
id_klienta	INT	11	tak
nazwisko	VARCHAR	60	tak
imie	VARCHAR	40	tak
kod_pocztowy	VARCHAR	6	
miejscowosc	VARCHAR	50	
ulica	VARCHAR	50	
nr_domu	VARCHAR	4	
PESEL	VARCHAR	11	tak
telefon	VARCHAR	12	
adres_e_mail	VARCHAR	70	

Tab. 24.2. Kolumny tabeli **klient**

Kod w języku SQL będzie wyglądał następująco:

```
CREATE TABLE 'klient' (
  'id_klienta' int(11) NOT NULL AUTO_INCREMENT,
  'nazwisko' varchar(60) NOT NULL,
   'imie' varchar(40) NOT NULL,
   'kod_pocztowy' varchar(6),
   'miejscowosc' varchar(50),
   'ulica' varchar(50),
   'nr_domu' varchar(4),
   'PESEL' varchar(11) NOT NULL,
   'telefon' varchar(12),
   'adres_e_mail' varchar(70),
  PRIMARY KEY (id_klienta)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

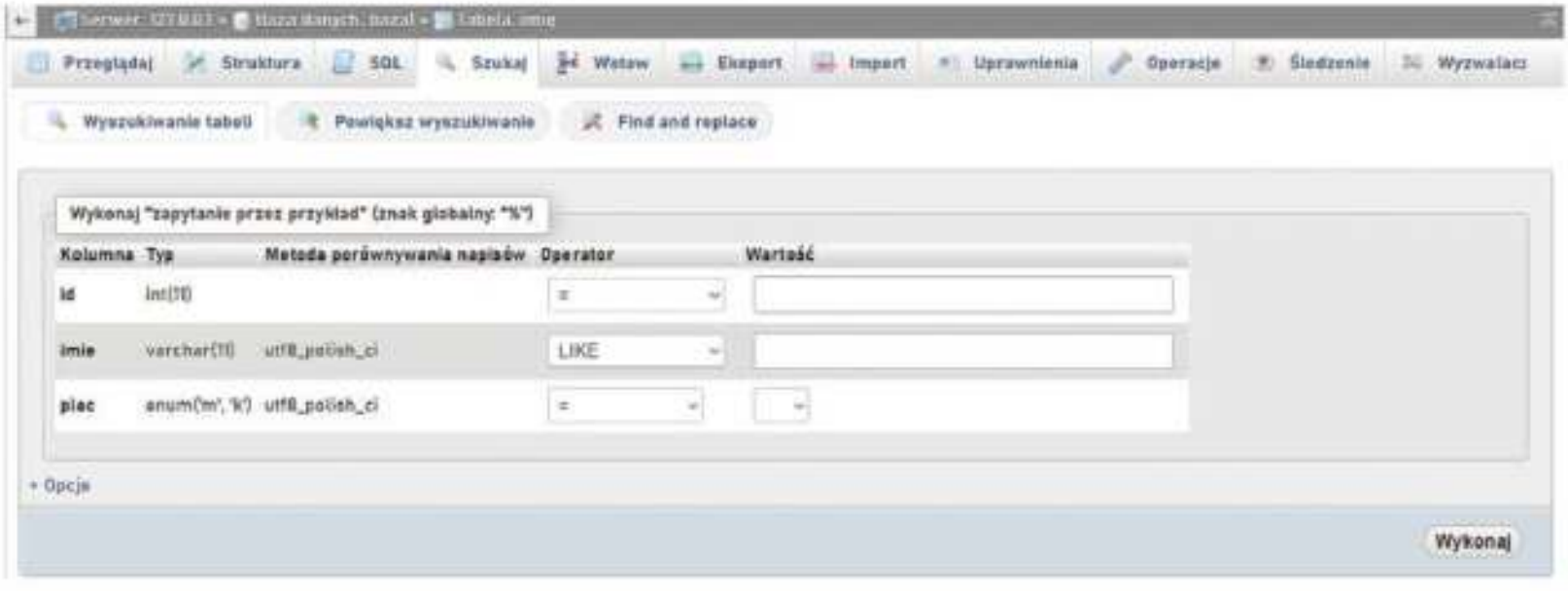
24.3. Wyszukiwanie rekordów w MySQL, czyli pierwsze zapytania SQL

Do wyszukiwania rekordów wykorzystamy narzędzia, które oferuje nam phpMyAdmin. W tabeli 24.3. przedstawiono listę dostępnych operatorów, których możemy użyć podczas wyszukiwania rekordów.

LIKE	Wykorzystywany do wyszukiwania określonego wzorca. W połączeniu z LIKE można używać dwóch symboli wieloznacznych:
LIKE %...%	% – znak procentu reprezentuje zero, jeden lub wiele dowolnych znaków. _ – znak podkreślenia reprezentuje dokładnie jeden dowolny znak.
NOT LIKE	Wykorzystywany do wyszukiwania rekordów niepasujących do wzorca. Można używać symboli wieloznacznych jak wyżej.
=	Wykorzystywany do wyszukiwania rekordów z określoną wartością w kolumnie.
!=	Wykorzystywany do wyszukiwania rekordów, które nie mają wskazanej wartości w kolumnie.
REGEXP **REGEXP ^...$**	Wykorzystywany do wyszukiwania danego wzorca określonego za pomocą wyrażeń regularnych. Jeśli wzorzec zostanie odnaleziony, funkcja zwraca wartość 1, a w przeciwnym razie 0. W przypadku gdy wyrażenie lub wzorzec ma wartość NULL, funkcja zwróci wartość NULL.
NOT REGEXP	Wykorzystywany do znajdowania rekordów niepasujących do danego wzorca określonego za pomocą wyrażeń regularnych.
IN (...)	Funkcja wykorzystywana do wyszukiwania rekordów, które mają wartość kolumny dopasowaną do podanych argumentów.
NOT IN (...)	Funkcja wykorzystywana do wyszukiwania rekordów, których wartość kolumny jest inna niż wprowadzone argumenty.
BETWEEN	Wykorzystywany do wyszukiwania rekordów, których wartość kolumny zawiera się pomiędzy wprowadzonymi argumentami. Zwracany zbiór jest zbiorem zamkniętym.
NOT BETWEEN	Wykorzystywany do wyszukiwania rekordów, których wartość kolumny nie zawiera się pomiędzy wprowadzonymi argumentami. Zwracany zbiór jest zbiorem otwartym.
IS NULL	Wykorzystywany do zwrócenia zbioru rekordów, których wartość kolumny jest równa wartości NULL.
IS NOT NULL	Wykorzystywany do zwrócenia zbioru rekordów, których wartość kolumny nie jest równa wartości NULL.

Tab. 24.3. Operatory wykorzystywane podczas wyszukiwania

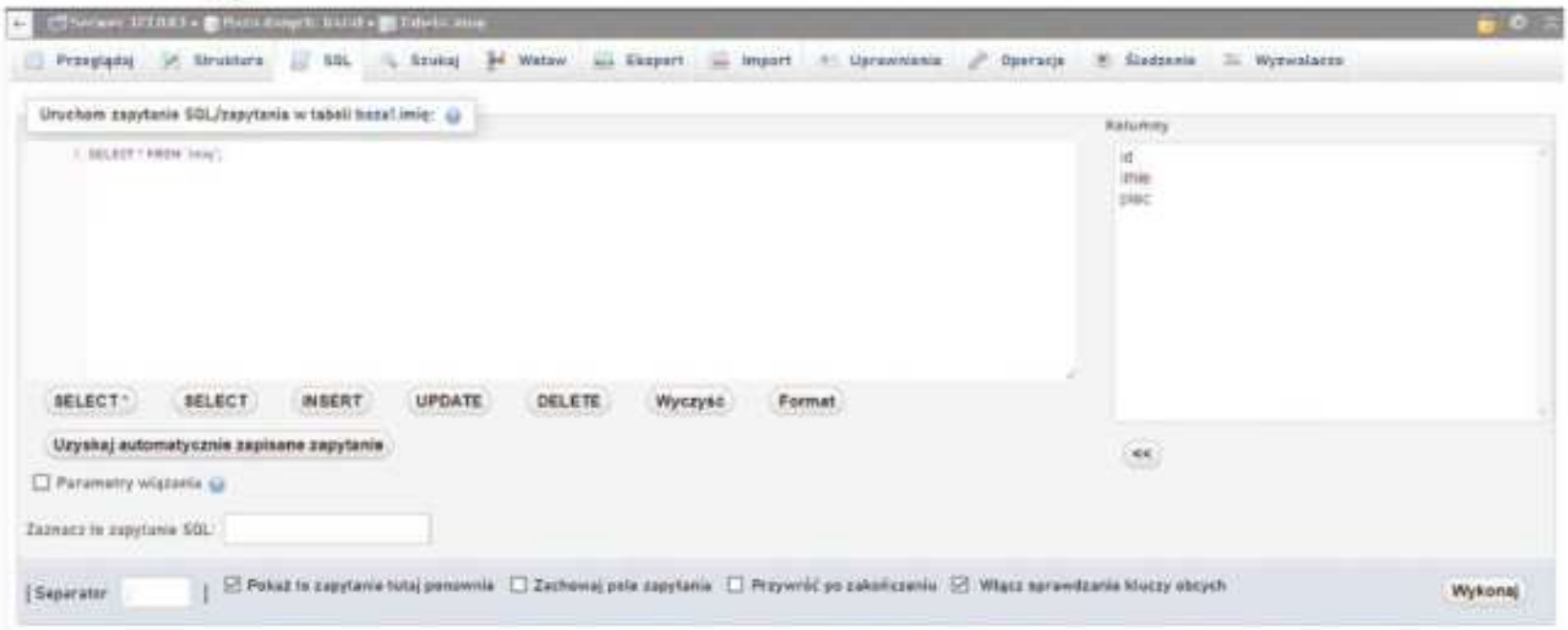
Najprostszym zapytaniem SQL, jakie możemy wykonać na stworzonej tabeli w zakładce **Szukaj**, jest zapytanie wyszukania (rys. 24.9).



Rys. 24.9. Wykonanie zapytania wyszukiwanie za pomocą zakładki **Szukaj**

zapytania w języku SQL

Możemy również wprowadzić zapytanie w zakładce SQL (rys. 24.10.):



Rys. 24.10. Zapytanie wyszukiwania z wykorzystaniem języka SQL

Zapytanie `SELECT * FROM 'imię';` zwróci nam wszystkie rekordy z tabeli posortowane rosnąco według klucza głównego.

Podczas pracy z językiem SQL bardzo pomocna będzie znajomość języka angielskiego. Aby posortować elementy malejąco, powinniśmy użyć zapytania:

```
SELECT * FROM 'imię' ORDER BY 'id' DESC;
```

`DESC` oznacza tu sortowanie malejące.

Możemy również wyświetlić dwa pierwsze dodane rekordy dzięki poleceniu `LIMIT` (rys. 24.11.).



Rys. 24.11. Zapytanie wyszukujące dwa pierwsze rekordy

ZADANIA DO WYKONANIA

1. Do utworzonej w rozdziale 24.3. bazy **księgarnia** za pomocą języka SQL dodaj tabelę **książki**, która będzie się składała następujących kolumn:

Nazwa	Typ	Rozmiar	Wymagane
id_ksiazki	INT	11	TAK
tytul	VARCHAR	100	TAK
autor	VARCHAR	80	TAK
cena	VARCHAR	5	
wydawnictwo	VARCHAR	50	
rok_wydania	VARCHAR	4	

2. Do utworzonej w rozdziale 24.3. bazy **księgarnia** za pomocą kreatora tabel dodaj tabelę **klienci**, która będzie się składała następujących kolumn:

Nazwa	Typ	Rozmiar	Wymagane
id_klienta	INT	100	TAK
nazwisko	VARCHAR	50	TAK
imie	VARCHAR	50	TAK
kod_pocztowy	VARCHAR	6	TAK
miejscowosc	VARCHAR	50	TAK
ulica	VARCHAR	50	TAK
nr_domu	VARCHAR	4	TAK
nr_mieszkania	VARCHAR	3	TAK
nr_telefonu	VARCHAR	12	TAK
adres_email	VARCHAR	50	TAK

3. Uzupełnij tabelę **książki** w bazie **księgarnia** czterema rekordami, a następnie wyświetl wszystkie rekordy malejąco.
4. Uzupełnij tabelę **klienci** w bazie **księgarnia** czterema rekordami, a następnie wyświetl wszystkie rekordy rosnąco z wykorzystaniem języka SQL.

PODSUMOWANIE LEKCJI

- MySQL jest systemem zarządzania relacyjnymi bazami danych, wykorzystującym język SQL. Jest to rodzaj oprogramowania komputerowego, którego kod źródłowy jest wydawany na podstawie licencji otwartego oprogramowania. ◂ s. 153
- W narzędziu, jakim jest phpMyAdmin, tabelę możemy stworzyć za pomocą kreatora tworzenia tabel, a także wpisując kod języka SQL. ◂ s. 153
- Zapytanie wyszukujące możemy tworzyć, wykorzystując kod języka SQL, oraz za pomocą zakładki Szukaj w phpMyAdmin. ◂ s. 160

25. Więcej pytań, czyli jeszcze o kwerendach w języku SQL

NA TEJ LEKCJI:

- dowiesz się, do czego jest wykorzystywane polecenie SELECT w języku SQL;
- nauczysz się konstruować zapytania z różnymi parametrami wyszukiwania;
- dowiesz się, jak łączy się kilka tabel w bazie, aby wyszukać w nich potrzebne informacje;
- poznasz metody wyszukiwania danych niepołączonych ze sobą.

Kwerendy pozwalają w różny sposób oglądać, zmieniać i analizować dane. Można ich również używać jako źródeł rekordów dla formularzy, raportów i stron dostępu do danych.

25.1. Wyszukiwanie danych w bazie

Najczęściej używanym poleceniem w języku SQL jest `SELECT`, które służy do wyszukiwania w bazie tych wierszy w tabelach, które będą spełniały podane przez nas kryteria. Pierwsze zapytania z tym poleceniem poznaliśmy w rozdziale 24. W różnych silnikach baz danych polecenie `SELECT` będzie miało inną składnię. Najbardziej znane oraz powszechnie wykorzystywane silniki bazy danych to Microsoft SQL, ORACLE czy darmowy MySQL. My będziemy posługiwać się silnikiem MySQL.

Polecenie `SELECT` możemy wykorzystywać do wykonywania zapytań na różny sposób. Jego składnia jest następująca (uwaga: w różnych silnikach baz danych polecenia mogą mieć inną składnię):

```
SELECT [opcje] pozycje
[INTO plik]
FROM nazwa_tabel
[WHERE] warunek
[GROUP BY rodzaj_grupowania]
[HAVING wartość_funkcji]
[ORDER BY porządek_sortowania]
[LIMIT limit]
[PROCEDURE nazwa_procedury(argumenty)]
;
```

W tym rozdziale będziemy pracować na bazie danych **księgarnia**, która składa się z następujących tabel (tab. 25.1.–25.5.).

Tabela **klienci**:

Nazwa	Typ	Rozmiar	Wymagana
id_klienta	INT	11	TAK
nazwisko	VARCHAR	50	TAK
adres	VARCHAR	100	TAK
miejscowosc	VARCHAR	30	

Tab. 25.1. Tabela **klienci**

Tabela **zamówienia**:

Nazwa	Typ	Rozmiar	Wymagana
id_zamowienia	INT	11	TAK
id_klienta	INT	11	TAK
Wartosc	FLOAT	6.2	
Data	DATE	30	

Tab. 25.2. Tabela **zamówienia**

Tabela **książki**:

Nazwa	Typ	Rozmiar	Wymagana
isbn	VARCHAR	13	TAK
autor	VARCHAR	50	TAK
tytul	VARCHAR	100	TAK
cena	FLOAT	4.2	

Tab. 25.3. Tabela **książki**

Tabela **pozycje_zamówione**:

Nazwa	Typ	Rozmiar	Wymagana
id_zamowienia	INT	11	TAK
isbn	VARCHAR	13	TAK
ilosc	VARCHAR	100	TAK

Tab. 25.4. Tabela **pozycje_zamówione**

Tabela **recenzje**:

Nazwa	Typ	Rozmiar	Wymagana
isbn	VARCHAR	13	TAK
autor_recenzji	VARCHAR	100	TAK
recenzja	TEXT	255	TAK

Tab. 25.5 Tabela **recenzje**

Poszczególne tabele zostały wypełnione następującymi danymi (tab. 25.6. –25.10.):

Tabela **klienci**:

id_klienta	nazwisko	adres	miejscowość
1	Jan Nowak	Kołobrzeska 12	Gdańsk
2	Anna Michalak	Bydgoska 7a	Elbląg
3	Sławomir Kowalski	Wojska Polskiego 39	Bydgoszcz
4	Paweł Kowalewski	Żołnierska 1	Mrągowo
5	Zuzanna Aleksandrowicz	Mazurska 112	Olsztyn

Tab. 25.6. Dane do tabeli **klienci**

Tabela **książki**:

isbn	autor	tytuł	cena
83-7361-925-9	Tomasz Francuz	Zagadnienia maturalne z informatyki	29.00
978-83-283-25	Ryan Stephans	SQL w 24 godziny	69.00
987-83-267-21	Krzysztof Czarkowski	E14. Tworzenie baz danych i administrowanie bazami	35.00
987-83-283-08	Robin Nixon	PHP, MySQL i JavaScript	99.00

Tab. 25.7. Dane do tabeli **książki**

Tabela **pozycje_zamówione**:

id_zamowienia	isbn	ilość
1	987-83-283-08	2
2	978-83-283-25	1
3	978-83-283-25	1
4	987-83-283-08	1
5	83-7361-925-9	3

Tab. 25.8. Dane do tabeli **pozycje_zamówione**

Tabela **recenzje**:

isbn	autor recenzji	recenzja
978-83-283-25	Jan Nowak	Pozycja ta bardzo pomogła w zrozumieniu zasad posługiwania się językiem SQL.

Tab. 25.9. Dane do tabeli **recenzje**

Tabela **zamówienia**:

id_zamowienia	id_klienta	wartość	data
1	5	198.00	2018-12-12
2	3	69.00	2018-12-03
3	4	168.00	2019-01-22
4	5	87.00	2018-12-07

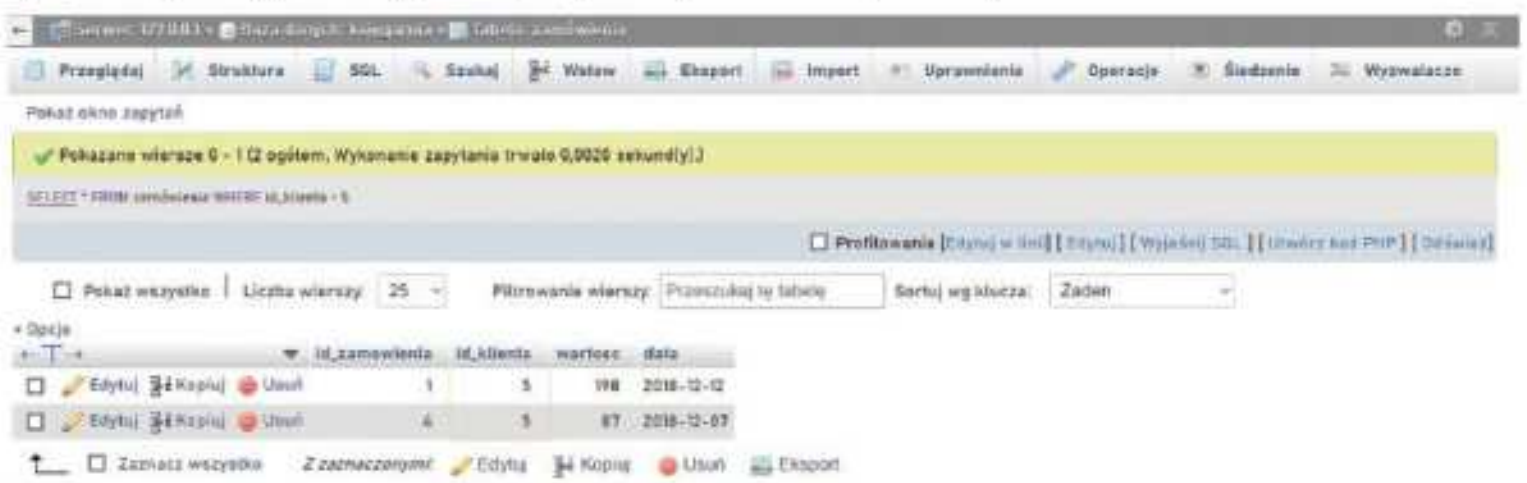
Tab. 25.10. Dane do tabeli **zamówienia**

◂ klauzula WHERE

W rozdziale 24. poznaliśmy wyszukiwanie wszystkich wierszy w bazie, które układaliśmy w kolejności malejącej lub rosnącej. W celu ograniczenia liczby wyświetleń wierszy należy określić kryteria wyszukiwania. Do tego celu wykorzystujemy klauzulę `WHERE`. Zapytanie z wykorzystaniem tej klauzuli ma następującą postać:

```
SELECT * FROM zamówienia WHERE id_klienta = 5
```

Wyszukujemy w tabeli **zamówienia** wszystkie zamówienia klienta o id równym 5. Wykonując to zapytanie, uzyskamy efekt taki jak na rysunku 25.1.



Rys. 25.1. Wynik wyszukiwania z użyciem klauzuli **WHERE**

W poleceniu `SELECT` można określić większą liczbę warunków wyszukiwania przy użyciu operatorów logicznych `AND` i `OR`, na przykład:

```
SELECT * FROM zamówienia WHERE id_klienta = 3 or id_
klienta = 4;
```

Efekt zastosowania tego zapytania będzie następujący (rys. 25.2.):



Rys. 25.2. Wynik wyszukiwania z użyciem klauzuli **WHERE** i operatorem logicznym **OR**

25.2. Łączenie kilku tabel

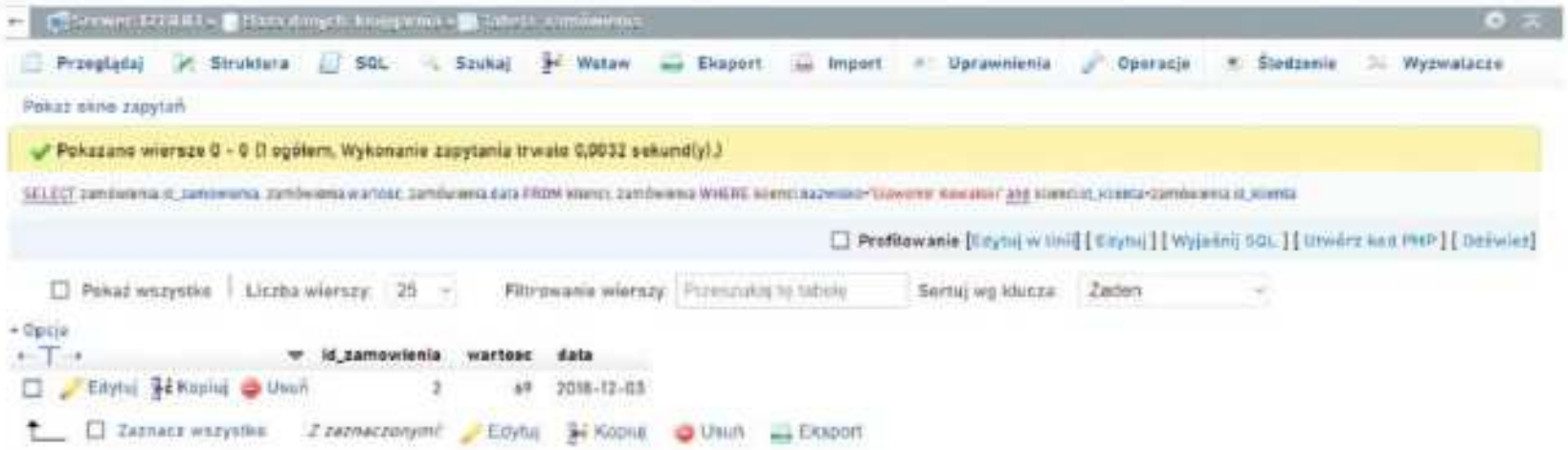
Często musimy przeanalizować i wyszukać konkretne informacje z większej liczby tabel. Przyjmijmy, że musimy się dowiedzieć, którzy klienci złożyli zamówienia w konkretnym miesiącu. Dane te są umieszczone w dwóch tabelach: **klienci** i **zamówienia**. Poza tym będzie nam potrzebna jeszcze informacja, które książki zostały zamówione, więc będziemy musieli zajrzeć jeszcze do tabeli **pozycje_zamówione**.

łączenie ▸ Aby uzyskać te informacje za pomocą języka SQL, należy wykonać operację nazywaną łączeniem. Dzięki niej scalimy dwie tabele (lub więcej) i będziemy mogli odnaleźć interesujące nas informacje. Chcemy na przykład się dowiedzieć, jakie książki zamówił Sławomir Kowalski. Najpierw musimy go odnaleźć w tabeli **klienci**, gdzie sprawdzimy jego **id_klienta**, a następnie w tabeli **zamówienia** odszukamy wiersze z id, które zostało temu klientowi przyporządkowane.

Kod SQL wyszukiwania zamówień Sławomira Kowalskiego będzie następujący:

```
SELECT zamówienia.id _ zamowienia, zamówienia.wartosc,
zamówienia.data
FROM klienci, zamówienia
WHERE klienci.nazwisko='Sławomir Kowalski'
and klienci.id _ klienta=zamówienia.id _ klienta;
```

Po wykonaniu zapytania otrzymamy następujący wynik (rys. 25.3.):



Rys. 25.3. Wynik wyszukiwania informacji w dwóch tabelach

Znalezienie informacji wymagało od nas przeszukania dwóch tabel, dlatego ich nazwy znalazły się w tekście zapytania. W tym celu dokonano również połączenia tych tabel. Przecinek oddzielający nazwy tych tabel jest równoważny użyciu słów kluczowych `INNER JOIN` lub `CROSS JOIN`. W wyniku tej operacji otrzymano jedną dużą tabelę, która zawiera każdy wiersz tabeli **klienci** połączony z każdym wierszem tabeli **zamówienia**, bez względu na to, czy dany klient złożył zamówienie, czy nie. Podczas łączenia tabel najczęściej poszukiwane są wiersze, które do siebie pasują. W omawianym przypadku są to rekordy, które przedstawiają zamówienia złożone przez wskazanego klienta wraz z rekordami zawierającymi jego dane. Zostały one odnalezione dzięki warunkowi połączenia w klauzuli `WHERE` – jest ona specjalnym typem wyrażenia warunkowego, które określa, jakie pola ustanawiają związek między tabelami. W omawianym przez nas przykładzie warunek miał postać:

```
klienci.id _ klienta=zamówienia.id _ klienta
```

W kodzie SQL, którym wyszukiwaliśmy zamówienia Sławomira Kowalskiego, pojawiły się zapisy (dwie składowe oddzielone znakiem kropki):

```
zamówienia.id _ zamowienia, zamówienia.wartosc, zamówienia.data,
klienci.nazwisko, klienci.id _ klienta, zamówienia.id _ klienta
```

Takim zapisem wyszukujemy konkretny rekord w interesującej nas tabeli: `zamówienia.id _ klienta`, czyli w tabeli zamówienia wyszukujemy informacje dotyczące id klienta.

Możemy też potrzebować wyszukać informacje w więcej niż dwóch tabelach.

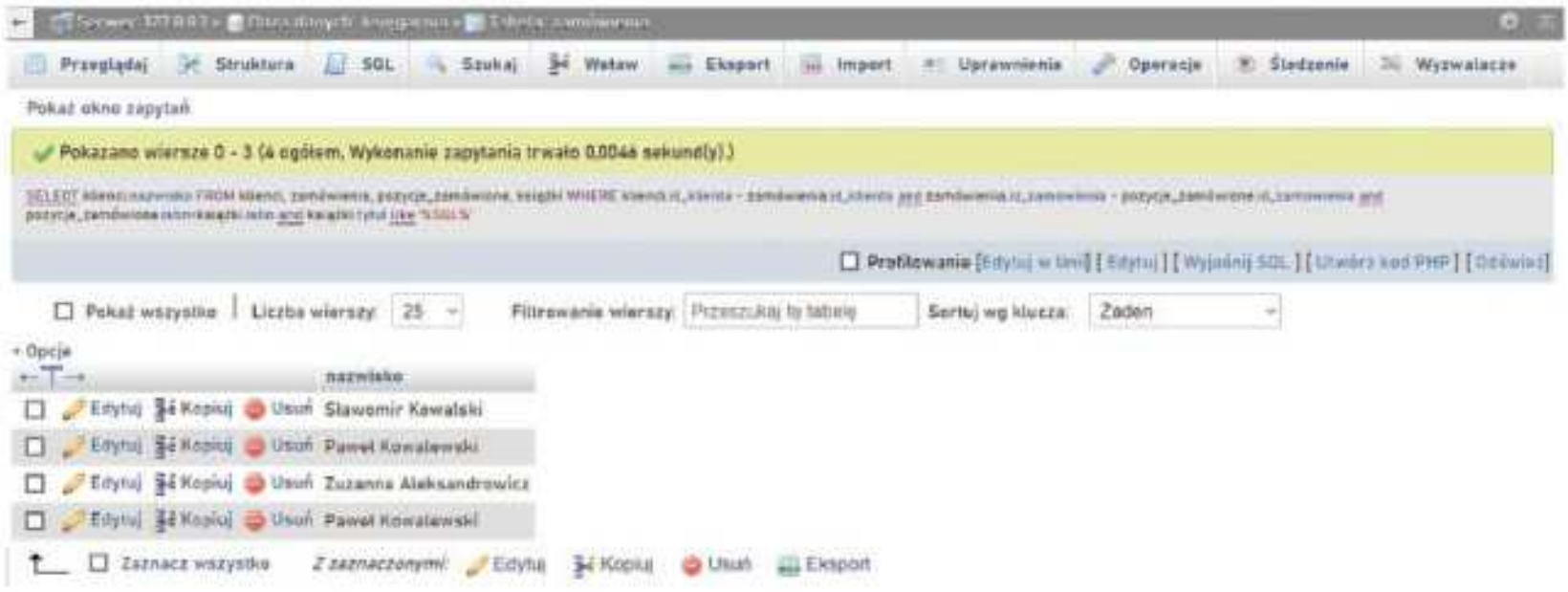
Przykład

Mamy w bazie danych odszukać wszystkich klientów, którzy zamówili książki o tematyce dotyczącej SQL.

Nasze zapytanie będzie wyglądać następująco:

```
SELECT klienci.nazwisko
FROM klienci, zamówienia, pozycje _ zamówione, książki
WHERE klienci.id _ klienta = zamówienia.id _ klienta
and zamówienia.id _ zamowienia = pozycje _ zamówione.
id _ zamowienia
and pozycje _ zamówione.isbn=książki.isbn
and książki.tytul like '%SQL%';
```

Po wykonaniu zapytania otrzymamy taki wynik (rys. 25.4.):



Rys. 25.4. Wynik wyszukiwania informacji w wielu tabelach

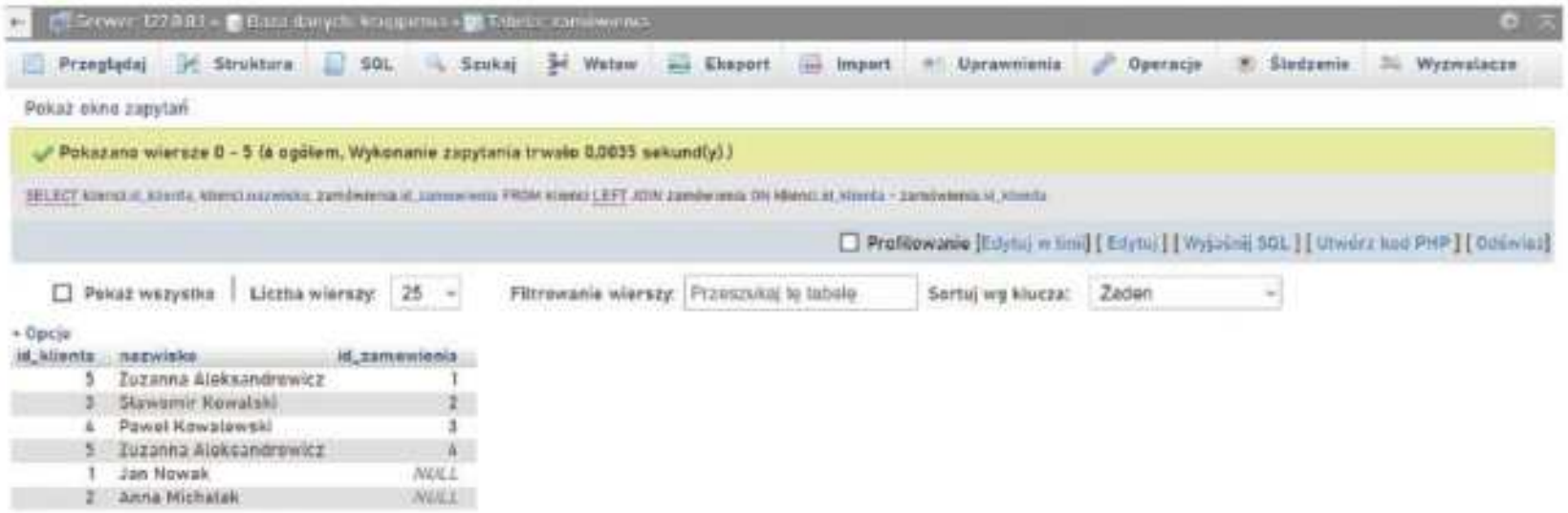
25.3. Wyszukiwanie danych niepołączonych ze sobą

Zapytania w poprzednich przykładach służyły do wyszukiwania informacji pochodzących z różnych tabel, które w jakiś sposób były ze sobą połączone. Kolejnym typem wyszukiwania będzie tzw. `LEFT JOIN`. Spowoduje ono wyszukanie wierszy spełniających zadany przez nas warunek połączenia między tabelami, jednak jeżeli w drugiej tabeli nie będzie odpowiedniego wiersza, to do tabeli wynikowej zostanie dodany wiersz z wartościami `NULL` w kolumnach pobieranych z tabeli wskazanej po prawej stronie polecenia `LEFT JOIN`.

Przykład

```
SELECT klienci.id_klienta, klienci.nazwisko,
zamówienia.id_zamowienia
FROM klienci LEFT JOIN zamówienia
ON klienci.id_klienta = zamówienia.id_klienta;
```

Po wykonaniu zapytania otrzymamy następujący wynik (rys. 25.5.):



Rys. 25.5. Wynik wyszukiwania danych niepołączonych ze sobą

ZADANIA DO WYKONANIA

1. Utwórz bazę danych **księgarnia** zgodnie z wytycznymi z rozdziału 25.1.
2. Napisz kod w języku SQL, dzięki któremu wyszukasz zamówienia dokonane w 2018 roku.
3. Napisz kod w języku SQL, dzięki któremu wyszukasz zamówienia książek dokonane przez Pawła Kowalewskiego.
4. Napisz kod w języku SQL, dzięki któremu wyszukasz informacje o klientach, którzy w ogóle nie dokonali żadnego zamówienia.

PODSUMOWANIE LEKCJI

s. 165 • Polecenie `WHERE` wykorzystujemy przy tworzeniu zapytania z określonymi kryteriami wyszukiwania.

s. 166 • Aby uzyskać dane z kilku tabel za pomocą języka SQL, należy wykonać operację nazywaną łączeniem. Dzięki niej scalimy dwie tabele (lub więcej) i będziemy mogli odnaleźć interesujące nas informacje.

s. 169 • Polecenie `LEFT JOIN` spowoduje wyszukanie wierszy spełniających zadany przez nas warunek połączenia między tabelami, jednak jeżeli w drugiej tabeli nie będzie odpowiedniego wiersza, to do tabeli wynikowej zostanie dodany wiersz z wartościami `NULL` w kolumnach pobieranych z tabeli wskazanej po prawej stronie polecenia `LEFT JOIN`.

26. Podzapytania, czyli instrukcje modyfikujące dane w języku SQL

NA TEJ LEKCJI:

- poznasz definicję podzapytań;
- nauczysz się korzystać z podzapytań podstawnych;
- poznasz podzapytania skorelowane, wierszowe i o charakterze tabeli tymczasowej;
- dowiesz się, dlaczego w bazach danych tabelom nadajemy aliasy.

◂ podzapytania

Podzapytania to szczególny typ zapytań, który występuje w języku SQL. Są one zagnieżdżone w innym zapytaniu i łatwiejsze w konstrukcji niż używanie złączeń i tabel tymczasowych, przydatnych podczas przetwarzania danych – zwłaszcza podczas przekształcania, w którym pośrednie wyniki są przejściowe.

26.1. Podzapytania podstawowe

W tym rozdziale również będziemy korzystać z bazy danych **księgarnia** z rozdziału 25.

Najczęściej podzapytania tworzymy, wykorzystując wyniki z jednego zapytania jako danych porównawczych dla drugiego zapytania.

Przykład

Wyszukajmy w bazie **księgarnia** zamówienie, które miało największą wartość.

```
SELECT id_klienta, wartosc
FROM zamówienia
WHERE wartosc = (SELECT MAX(wartosc) FROM zamówienia);
```

Po wykonaniu tego zapytania otrzymamy następujący wynik (rys. 26.1.):



Rys. 26.1. Wynik zapytania o zamówienie o najwyższej wartości

W odpowiedzi na nasze zapytanie otrzymaliśmy informację o zamówieniu za największą kwotę. Nasze podzapytanie zwraca jedną wartość, która stanowi wartość porównawczą dla zapytania zewnętrznego.

operatory podzapytań

Stosując podzapytania, mamy do dyspozycji cztery podstawowe operatory:

1. `ANY`, który zwraca wartość true (jeśli argument jest prawdziwy), jeżeli porównanie będzie miało również wartość true dla dowolnego wiersza z podzapytania, np.:

```
SELECT kolumna1
FROM tabela1
WHERE kolumna1 >
ANY (SELECT kolumna1 FROM tabela2);
```

2. `IN`, który jest równoważnikiem konstrukcji `=ANY`:

```
SELECT kolumna1
FROM tabela1
WHERE kolumna1 >
IN (SELECT kolumna1 FROM tabela2);
```

3. `SOME` jest aliasem (inną nazwą) dla `ANY`, czasami bardziej zrozumiałym dla użytkownika:

```
SELECT kolumna1
FROM tabela1
WHERE kolumna1 >
SOME (SELECT kolumna1 FROM tabela2);
```

4. `ALL` zwraca wartość true, jeżeli porównanie będzie miało wartość true dla wszystkich wierszy podzapytania:

```
SELECT kolumna1
FROM tabela1
WHERE kolumna1 >
ALL (SELECT kolumna1 FROM tabela2);
```

Każdy z wymienionych operatorów występuje tylko i wyłącznie z operatorem równości. Wyjątek stanowi operator `IN`, ponieważ operator równości jest w nim niejako zagnieżdżony.

26.2. Podzapytania skorelowane

podzapytanie skorelowane

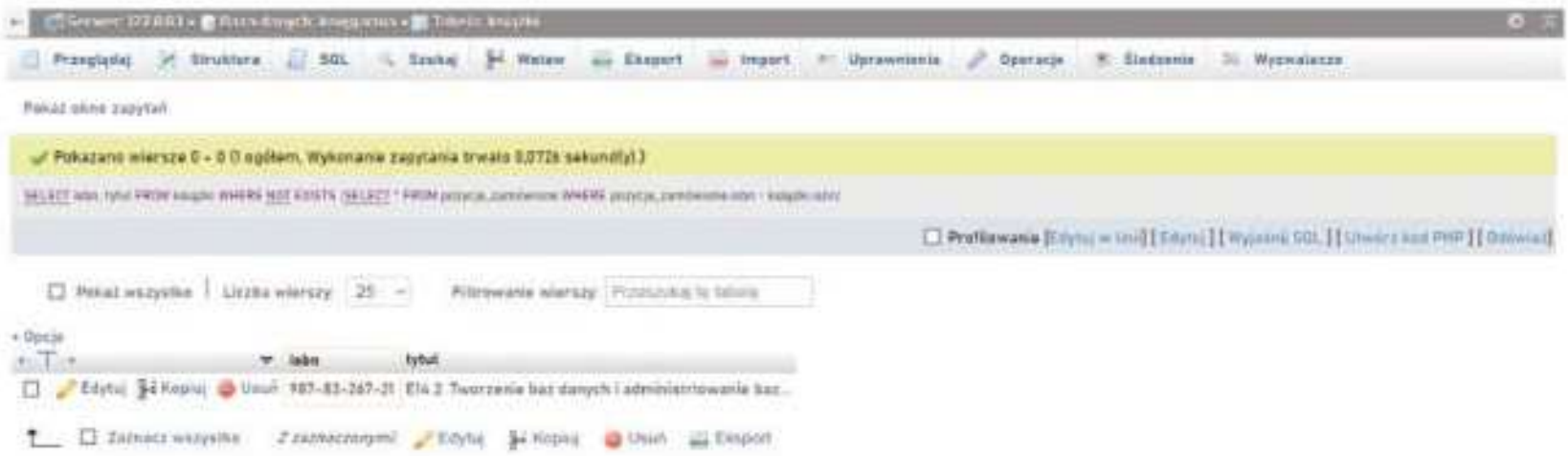
W podzapytaniu podstawowym najpierw jest wykonywane podzapytanie, a potem jego wynik jest używany jako warunek wyboru wierszy przez zapytanie nadrzędne. Natomiast podzapytanie skorelowane, a właściwie jego zapytanie nadrzędne, sprawdza wiersz po wierszu i porównuje jego zawartość (przez warunek porównania określony w klauzuli `WHERE`) z wynikiem podzapytania. Jest ono jest wyliczane od nowa dla każdego kolejnego wiersza na podstawie wybranej wartości występującej w tym właśnie wierszu.

Przykład

Stwórzmy podzapytanie, które wyszuka nam wszystkie książki, które nie były nigdy zamówione.

```
SELECT isbn, tytul
FROM książki
WHERE NOT EXISTS (SELECT * FROM pozycje_zamówione WHERE
pozycje_zamówione.isbn = książki.isbn);
```

Po wykonaniu tego zapytania otrzymamy następujący wynik (rys. 26.2.):



Rys. 26.2. Wynik zapytania o niezamówione pozycje

Zwróćmy uwagę na to, że w wewnętrznym zapytaniu na liście `FROM` została wyszczególniona tabela **pozycje_zamówione**, ale odwołujemy się w nim również do kolumny **książki.isbn**, czyli nasze zapytanie wewnętrzne odwołuje się do danych z zapytania zewnętrznego.

◂ EXISTS

W tym podzapytaniu wykorzystaliśmy piąty operator, który jest stosowany w podzapytaniach – `EXISTS`. Zwraca on wartość true, jeśli w podzapytaniu znajduje się co najmniej jeden pasujący wiersz i odwrotnie, `NO EXISTS` zwróci nam wartość true, jeżeli żaden z wierszy nie będzie pasował do argumentu zapytania. Operator `EXISTS` używany jest tylko w podzapytaniach skorelowanych.

26.3. Podzapytania wierszowe i podzapytania o charakterze tabeli tymczasowej

◂ podzapytania wierszowe

Do tej pory rozważaliśmy podzapytania odnoszące się do kolumn, to znaczy podzapytania, które zwracają pojedynczą wartość kolumny. W wielu przypadkach była to jedynie wartość true lub false (jeśli argument był fałszywy). Podzapytanie wierszowe jest takim podzapytaniem, które zwraca jeden wiersz, i tym samym może zwrócić więcej niż jedną wartość kolumny.

Konstrukcja zapytania wierszowego będzie wyglądała następująco:

```
SELECT kolumna1, kolumna2, kolumna3
FROM tabela1
WHERE (kolumna1, kolumna2, kolumna3) IN (SELECT kolumna1,
kolumna2, kolumna3 FROM tabela2);
```

Z tego typu zapytań korzysta się, aby odnaleźć w jednej tabeli wiersze, które istnieją również w drugiej.

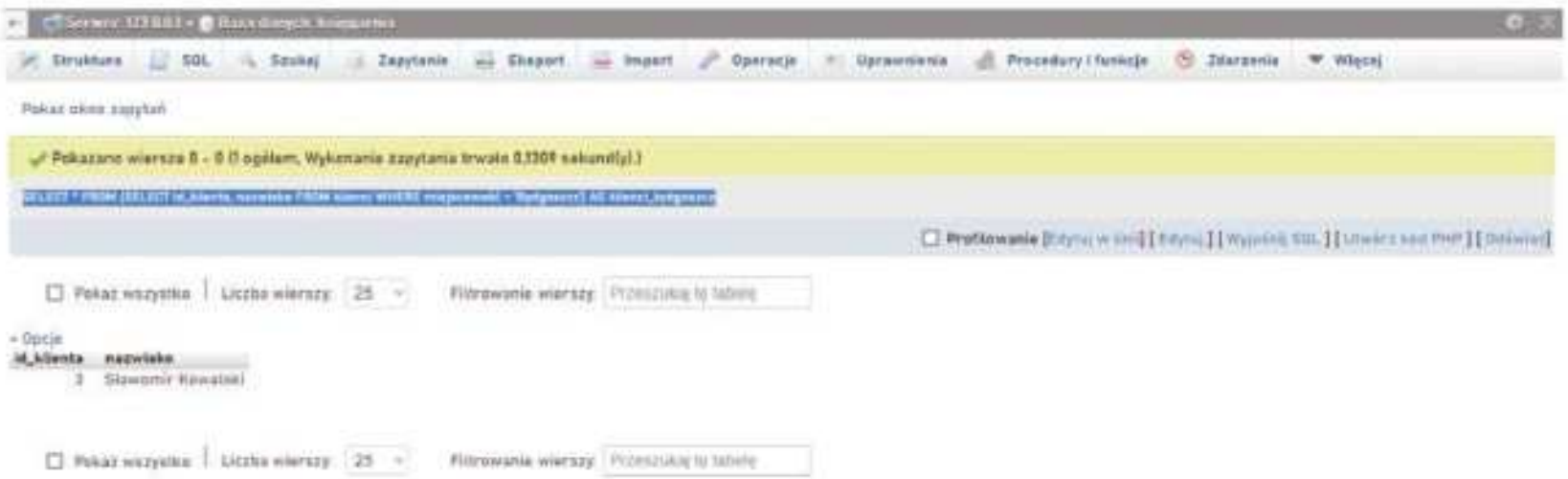
podzapytania o charakterze tabeli tymczasowej

Podzapytanie możemy umieścić również w klauzuli `FROM` zapytania zewnętrznego. Dzięki temu jest możliwe wykonywanie zapytań na wynikach podzapytania, które jest traktowane jak tabela tymczasowa.

Przykład

```
SELECT * FROM (SELECT id_klienta,
nazwisko FROM klienci WHERE miejscowość = 'Bydgoszcz')
AS klienci_bydgoszcz
```

Po wykonaniu tego zapytania otrzymamy następujący wynik (rys. 26.3.):



Rys. 26.3. Wynik zastosowania podzapytania o charakterze tabeli tymczasowej

W tym przykładzie podzapytanie zostało umieszczone w klauzuli `FROM`. Zaraz za nawiasami zamykającymi nadaliśmy jego wynikom alias.

aliasy

W bazach danych, jak w przykładzie powyżej, czasami niezbędne jest użycie aliasu, czyli innej nazwy tabeli. Jest to często wygodniejsze lub nawet konieczne. Aliasy nadaje się na początku zapytania i używa się ich w pozostałej części kodu.

Przykład

```
SELECT nl.nazwisko
FROM klienci AS nl, zamówienia AS zam,
pozycje_zamówione AS poz, książki AS ks
WHERE nl.id_klienta = zam.id_klienta
AND zam.id_zamowienia = poz.id_zamowienia
AND poz.isbn = ks.isbn
AND ks.tytul LIKE '%SQL%';
```

Po wykonaniu tego zapytania otrzymamy następujący wynik (rys. 26.4.):



Rys. 26.4. Wynik zapytania wykorzystującego alias

Zapytanie miało nam wyszukać nazwiska klientów, którzy zamówili książki dotyczące SQL. Do wskazania tabel, które miały być przeszukane, została użyta klauzula `AS` w celu zadeklarowania dla nich aliasów.

ZADANIA DO WYKONANIA

1. Korzystając z podzapytań, wyszukaj w bazie danych **księgarnia** zamówienie, które miało najniższą wartość.
2. Używając podzapytania skorelowanego, wyszukaj w bazie danych **księgarnia** klienta, który nigdy nie dokonał zamówienia.
3. Używając podzapytania o charakterze tabeli tymczasowej, wyszukaj klienta z Gdańska.

PODSUMOWANIE LEKCJI

- Podzapytania są zagnieżdżone w innym zapytaniu i łatwiejsze w konstrukcji niż używanie połączeń i tabel tymczasowych. s. 169
- Stosując podzapytania podstawowe, korzystamy z czterech operatorów: `ANY`, `IN`, `SOME` i `ALL`. s. 170
- Podzapytanie skorelowane wykonuje się wielokrotnie – tyle razy, ile rekordów przegląda zapytanie zewnętrzne. s. 170
- Operatory `EXISTS` i `NO EXISTS` są stosowane tylko podczas używania podzapytań skorelowanych. s. 171
- Podzapytanie wierszowe zwraca jeden wiersz i tym samym może zwrócić więcej niż jedną wartość kolumny. s. 171
- Podzapytanie o charakterze tabeli tymczasowej umieszczamy w klauzuli `FROM` zapytania zewnętrznego. Dzięki temu możemy wykonać zapytanie na wynikach podzapytania, które jest traktowane jak tabela tymczasowa. s. 172
- Alias nadaje się, aby ułatwić sobie pracę z tabelami w bazie danych. W niektórych sytuacjach, np. podczas stosowania podzapytań o charakterze tabeli tymczasowej, stosowanie aliasów jest koniecznością. s. 172

27. Modyfikacje, czyli zarządzanie bazą danych

NA TEJ LEKCJI:

- dowiesz się, w jaki sposób modyfikować rekordy w bazie danych;
- poznasz polecenia, za pomocą których można wprowadzać zmiany w strukturze bazy danych;
- nauczysz się usuwać rekordy, tabele i całe bazy danych.

W pracy z bazami danych często zachodzi potrzeba wprowadzania zmian w strukturze bazy lub w jej zawartości. Aktualizując je, musimy na przykład wprowadzić nowe tabele lub hurtowo zmienić pewne wartości rekordów. Praca z bazą to również usuwanie nieprzydatnych rekordów czy tabel.

Aktualizacja rekordów czy zmiana struktur kolumn to nie są zbyt trudne czynności, ale podczas ich wykonywania musimy zachować ostrożność, bo wprowadzane zmiany często są nieodwracalne.

27.1. Zmiany rekordów w bazie danych

UPDATE ▸ Praca z bazą danych to nie tylko wyszukiwanie danych, lecz także wprowadzanie zmian. Pracując dalej na naszej bazie **księgarnia**, nauczymy się, w jaki sposób to robić.

Pierwszym krokiem będzie zmiana ceny książek, które mamy zapisane w naszej bazie. Do tego celu posłużymy się poleceniem `UPDATE`.

Składnia tego polecenia wygląda następująco:

```
UPDATE tabela1
SET kolumna1 = wyrazenie1, kolumna2 = wyrazenie2, …
[WHERE warunek]
[ORDER BY kryteria_przyporzadkowania]
[LIMIT ilosc]
```

Działanie tego polecenia powoduje zmianę wartości, które są zapisane w kolumnach we wskazanej tabeli, na nowe. Zakres modyfikacji, jakiej chcemy dokonać, może być ograniczony do konkretnego zbioru wierszy za pomocą klauzuli `WHERE`, a ich liczbę możemy limitować poprzez klauzulę `LIMIT`.

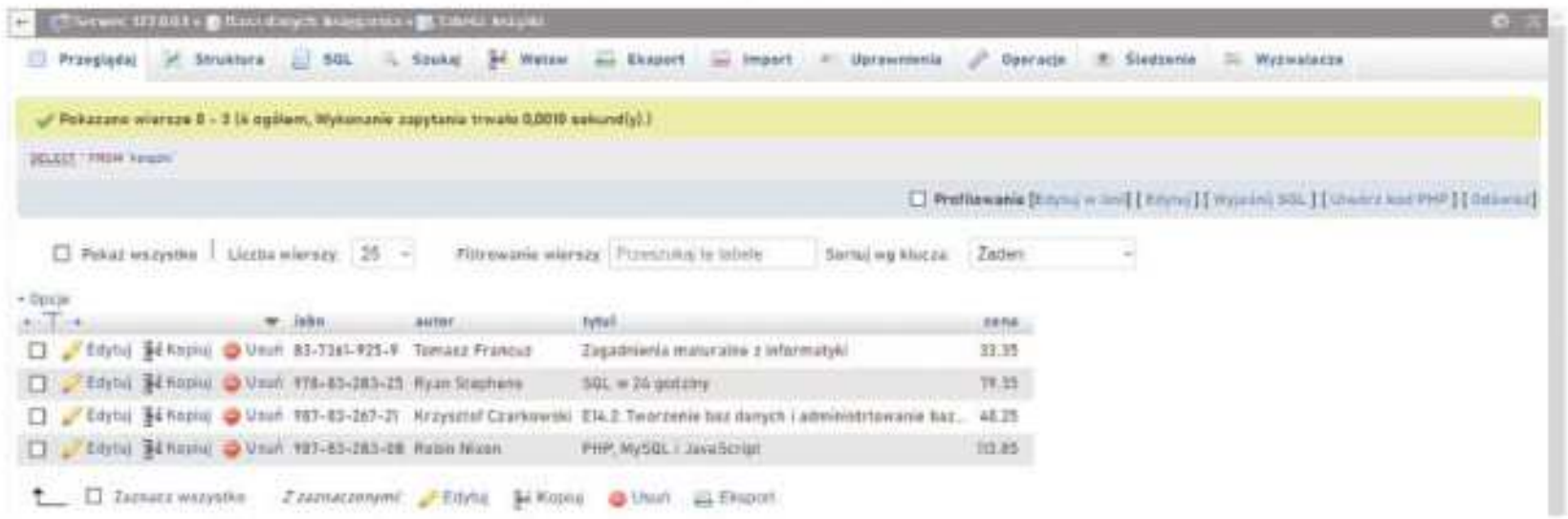
Klauzuli `ORDER BY` używamy zazwyczaj z klauzulą `LIMIT`. Jeżeli chcemy zmienić tylko pięć pierwszych wierszy, to za pomocą `ORDER BY` musimy je najpierw uporządkować.

Przykład

Podnosimy ceny książek o 15%. Użyjemy polecenia `UPDATE` bez klauzuli `WHERE`.

```
UPDATE książki
SET cena=cena*1.15;
```

Efekt zastosowania tego zapytania będzie następujący (rys. 27.1.):



Rys. 27.1. Wynik wykonania polecenia **UPDATE**

Jeżeli chcemy zmienić tylko jeden element, na przykład zaktualizować adres naszego klienta, zmieniając nazwę ulicy, przy której mieszka, to zapiszemy to w następujący sposób:

```
UPDATE klienci
SET adres = 'Jagiellońska 210'
WHERE id_klienta = 1;
```

Efekt zastosowania tego zapytania będzie następujący (rys. 27.2.):



Rys. 27.2. Zmiana jednego elementu w tabeli za pomocą polecenia **UPDATE**

Jak widzimy, w adresie klienta Jana Nowaka zmieniła się ulica (na Jagiellońską 210).

27.2. Zmiany w strukturze tabel

ALERT TABLE ▸ Czasami zachodzi potrzeba zmiany całej struktury tabeli. W tym celu skorzystamy z polecenia `ALERT TABLE`, które ma następująca składnię:

```
ALERT TABLE tabela1 zmiana [, zmiana, ...]
```

Standardy ASCII SQL umożliwiają nam dokonanie tylko jednej zmiany na raz, na szczęście MySQL nie ma takich ograniczeń. Każda klauzula zmiany może prowadzić do innego rodzaju modyfikacji naszej tabeli.

Zapoznaj się z tabelą 27.1. Przedstawiono w niej najważniejsze rodzaje zmian, których możemy dokonać za pomocą polecenia `ALERT TABLE`:

Składnia	Opis zmiany
ADD [COLUMN] (opis_kolumny) [FIRST | AFTER kolumna]	Dodaje nową kolumnę na określonej pozycji; jeżeli nie podamy pozycji, to kolumna zostanie wstawiona na końcu tabeli. Parametr `opis_kolumny` wymaga podania jej nazwy i typu.
ADD [COLUMN] (opis_kolumny, opis_kolumny, ...)	Dodaje jedną lub więcej kolumn na końcu tabeli.
ALERT [COLUMN] kolumna {SET DEFAULT wartość | DROP DEFAULT}	Przypisuje lub usuwa domyślną wartość kolumny.
CHANGE [COLUMN] kolumna opis_nowej_kolumny	Zmienia kolumnę w sposób zależny od podanej wartości parametru `opis_nowej_kolumny`. Pozwala również na zmianę nazwy kolumny.
MODIFY [COLUMN] opis_kolumny	Działa podobnie jak `CHANGE`, ale nie pozwala na zmianę nazwy kolumny.
DROP [COLUMN] kolumna	Usuwa kolumnę.
DROP PRIMARY KEY	Usuwa klucz podstawowy, ale nie usuwa kolumny.
DROP FOREIGN KEY klucz	Usuwa klucz obcy, ale nie usuwa kolumny.
RENAME [AS] nowa_nazwa_tabeli	Zmienia nazwę wskazanej tabeli.
ORDER BY nazwa_kolumny	Tworzy ponownie tabelę z wierszami ułożonymi we wskazanym porządku.
opcje_tabel	Umożliwia zresetowanie opcji tabel – używana składnia jest identyczna jak w przypadku `CREATE TABLE`.

Tab. 27.1. Niektóre rodzaje zmian, których można dokonać poleceniem **ALERT TABLE**

Przykład

Zmieniamy długość znaków w tabeli **klienci** w kolumnie nazwisko, ponieważ w trakcie dopisywania nowych klientów okazało się, że 50 znaków to za mało. Jeżeli podczas tworzenia tabeli w którejś z kolumn zdefiniujemy zbyt małą liczbę znaków możliwych do wprowadzenia, to program będzie sam skracał zapis.

```
ALTER TABLE klienci
MODIFY nazwisko varchar(70) NOT NULL;
```

Efektem zastosowania tego zapytania będzie zwiększenie liczby znaków w kolumnie **nazwisko** do 70.

Przykład

Dodajmy nową kolumnę **podatek** do tabeli zamówienia:

```
ALTER TABLE zamówienia
ADD podatek float (6.2) AFTER wartosc;
```

Efekt zastosowania tego zapytania będzie następujący (rys. 27.3.):



Rys. 27.3. Dodanie nowej kolumny do tabeli

Jeśli chcemy usunąć dodaną kolumnę **podatek**, to zrobimy to, używając poniższego kodu SQL:

```
ALERT TABLE zamówienia
DROP podatek;
```

Efektem wykonania tego kodu będzie usunięcie kolumny **podatek** z tabeli zamówienia.

27.3. Usuwanie rekordów, tabel i całych baz

Do usuwania rekordów z bazy danych wykorzystamy polecenie `DELETE`, którego składnia ma postać: ◂ DELETE

```
DELETE [LOW PRIORITY] [QUICK] [IGNORE] FROM tabela1
[WHERE warunek]
[ORDER BY kolumny_sortowania]
[LIMIT ilosc]
```

Jednak najprostszym sposobem, aby usunąć wszystkie dane z tabeli, jest wpisanie polecenia:

```
DELETE FROM tabela;
```

Aby usunąć tylko część wierszy, zastosujemy klauzulę `WHERE`. Zazwyczaj korzystamy z niej w momencie aktualizowania naszej bazy.
Klauzulę `LIMIT` zastosujemy wtedy, gdy będziemy chcieli określić liczbę wierszy do usunięcia. Oczywiście w takim przypadku będziemy musieli zastosować również klauzulę `ORDER BY`.

Przykład

Z bazy **księgarnia** usuniemy klienta o numerze id=5

```
DELETE FROM klienci
WHERE id_klienta = 5;
```

DROP ▸ Gdy zajdzie potrzeba usunięcia całej tabeli, to skorzystamy z polecenia `DROP TABLE`. Składnia tego polecenia jest następująca:

```
DROP TABLE tabela1;
```

Użycie tego polecenia wykasuje nam całą tabelę wraz ze wszystkimi danymi, jakie ona zawierała. Zanim go użyjemy, upewnijmy się, czy jest to konieczne.

Usunięcie całej bazy danych też nie jest zbyt skomplikowane w języku SQL. Do tego celu użyjemy polecenia:

```
DROP DATABASE baza;
```

Użycie go spowoduje usunięcie bazy danych z całą zawartością. Tak jak poprzednio, przed wykonaniem tej czynności powinniśmy się upewnić, czy na pewno o tę konkretną bazę nam chodzi. Użycie tego polecenia jest nieodwracalne.

ZADANIA DO WYKONANIA

1. Korzystając z polecenia `UPDATE`, w bazie danych **księgarnia** zwiększ wartość **książki**, której id = 3, o 25%.
2. Korzystając z polecenia `UPDATE`, w bazie danych **księgarnia** zmień adres (ulicę i miejscowość) klienta, którego id=3, na Starowiślna 13, Kraków.
3. Korzystając z polecenia `ALERT TABLE` , dodaj do tabeli **klienci** w bazie danych **księgarnia** kolumnę **telefon**, która będzie zawierała 12 znaków.

PODSUMOWANIE LEKCJI

s. 174 ▸ • Korzystając z polecenia `UPDATE`, jesteśmy w stanie zmienić w bazie danych informacje dotyczące zarówno pojedynczych rekordów, jak i całych kolumn.

s. 176 ▸ • Dzięki poleceniu `ALERT TABLE` jesteśmy w stanie zmienić strukturę całej tabeli. Możemy dodać lub usunąć jedną lub więcej kolumn.

s. 177 ▸ • Aby usunąć pojedyncze rekordy, stosujemy składnię `DELETE FROM` tabela `WHERE` warunek.

s. 178 ▸ • Do usunięcia danych i definicji tabeli lub całych baz danych stosujemy polecenie `DROP`.

28. Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe

NA TEJ LEKCJI:

- dowiesz się, jak dzielimy zagrożenia związane z atakiem na bazy danych;
- poznasz zasady bezpieczeństwa zarządzania bazami danych;
- nauczysz się tworzyć kopie baz danych;
- dowiesz się, czym jest przyrostowa kopia bezpieczeństwa bazy danych;
- poznasz metody odzyskiwania danych z kopii bezpieczeństwa bazy danych.

Podstawowym zadaniem administratora bazy danych jest jej zabezpieczenie. Spotykamy się bowiem z różnymi rodzajami zagrożeń, które mogą wniknąć z awarii sprzętu, błędów w oprogramowaniu, umyślnego (lub nie) działania człowieka, czy też z bardzo prozaicznych powodów (np. zanik napięcia elektrycznego).

28.1. Bezpieczeństwo baz danych

Administrator zarządzający bazami danych ma obowiązek odpowiednio zabezpieczyć informacje, które są w tych bazach przechowywane. Ataki, z jakimi możemy się spotkać, dzielimy na dwa rodzaje:

- **pasywne** – jest to podglądanie lub prowadzenie nasłuchu sieciowego – podglądanie wpisywanego hasła, zainstalowanie programu typu keylogger, prowadzenie nasłuchu w ruchu sieci radiowych, przechwytywanie e-maili itp.,
- **aktywne** – polegające na modyfikacji danych, ingerencji w obrębie sieci, podszyciu się pod użytkownika lub urządzenie zarządzające bazą, posłużeniu się danymi przechwyconymi w sposób pasywny i w ten sposób wprowadzeniu, zablokowaniu lub wykradzeniu informacji z bazy danych, zagłuszaniu lub zniszczeniu medium transmisyjnego.

Najmniejszym powodem do obaw o bezpieczeństwo bazy danych powinna być utrata zasilania elektrycznego, ponieważ serwery przechowujące bazy danych i zarządzające nimi zawsze powinny być podpięte do urządzeń podtrzymujących takie zasilanie na wypadek awarii.

zasady bezpieczeństwa zarządzania bazą danych

Każdy administrator bazy danych powinien zachować zasady bezpieczeństwa, które wymagają kontroli dostępu do Systemu Zarządzania Bazami Danych i do samych danych. Zasady te to:

- **dostępność** – każdy z użytkowników bazy powinien mieć dostęp tylko do takich jej obszarów, które są dla niego przeznaczone,
- **poufność** – użytkownik nie może podglądać danych przechowywanych przez innych użytkowników bazy oraz tych, do których nie powinien mieć w ogóle dostępu,
- **spójność** – każdy użytkownik ma nałożone ograniczenia dotyczące modyfikacji danych, do których ma dostęp.

Aby ułatwić zarządzanie bazami danych, administrator definiuje role, czyli uprawnienia, ograniczenia lub ich brak dla każdej z grup lub indywidualnych użytkowników.

28.2. Tworzenie kopii bezpieczeństwa

spójność bazy danych

Przed każdą ingerencją administratorską należy wykonać backup. W szczególności przed czyszczeniem bazy czy naprawianiem danych. Po wykonaniu tej czynności powinno się wykonać kolejny backup.
Do sprawdzenia poprawności tabel w bazie danych służy polecenie `CHECK TABLE`, które ma postać:

```
CHECK TABLE tabela1, tabela2, ...;
```

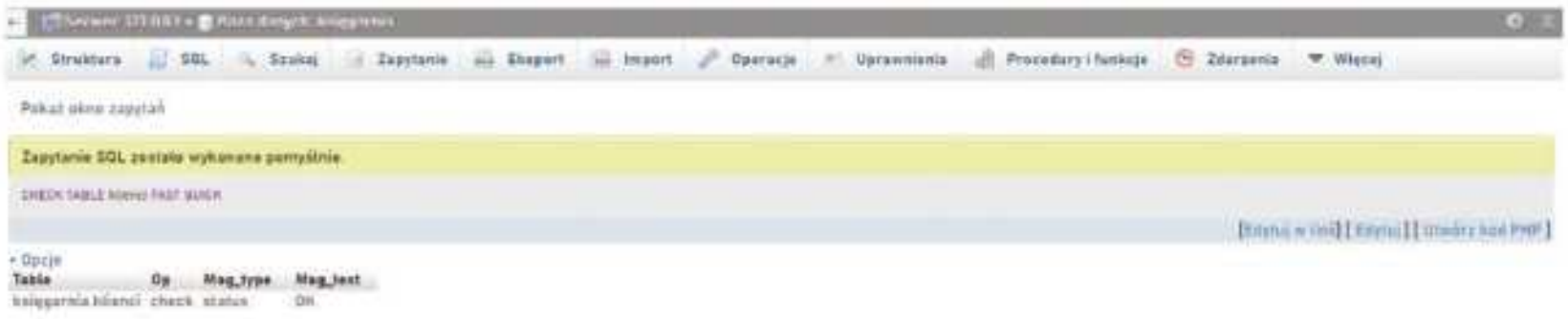
Polecenie to może mieć następujące opcje:
– `CHANGED` – dotyczy tylko tabel, które zmieniły się od ostatniej kontroli lub nie zostały poprawnie zamknięte,
– `EXTENDED` – sprawdzane są wszystkie referencje, gdy musimy mieć pewność, że dane w tabeli są spójne,
– `FAST` – dotyczy tylko tabel, które nie zostały poprawnie zamknięte,
– `MEDIUM` – dotyczy poprawności wierszy oraz niektórych referencji,
– `QUICK` – opcja ta sprawdza poprawność tabel bez weryfikacji, czy wiersze mają właściwe referencje.

Podane opcje mogą być połączone w jednym poleceniu.

Przykład

```
CHECK TABLE klienci FAST QUCIK;
```

Efekt zastosowania polecenia sprawdzenia poprawności tabeli przedstawiono na rysunku 28.1.



Rys. 28.1. Zastosowanie polecenia CHECK TABLE

naprawa uszkodzonej tabeli

Jeżeli podczas sprawdzania spójności bazy danych okaże się, że w tabelach są jakieś błędy, przechodzimy do ich naprawiania. Do tego służy polecenie `REPAIR TABLE`, które ma postać:

```
REPAIR TABLE tabela1;
```

Natomiast do naprawy tylko pliku indeksu można użyć polecenia:

```
REPAIR TABLE tabela1 QUICK;
```

Każda osoba administrująca bazami danych powinna mieć wypracowany harmonogram tworzenia kopii zapasowych. W zależności od potrzeby powinny to być kopie pełne (całkowita kopia danych z miejsca źródłowego), przyrostowy (zapisuje się jedynie te dane, które uległy zmianie od czasu wykonania poprzedniego backupu) lub różnicowy (to typ pośredni między backupem pełnym, a przyrostowym; zapisuje się zmiany, które zaszły od ostatniego pełnego backupu).

Utworzenie pełnej kopii bezpieczeństwa możliwe jest poprzez program narzędziowy **mysqldump**. Program ten tworzy plik **sql**, który zawiera komendy źródłowej bazy danych.
Przed rozpoczęciem tworzenia pełnej kopii bezpieczeństwa musimy zablokować bazę poleceniem `FLUSH TABLES`. Składnia tego polecenia ma postać:

- wykonanie kopii zapasowej:

```
mysqldump -u użytkownik -p[hasło] [baza_danych] > plik.sql
```

- przywrócenie bazy danych z kopii zapasowej:

```
mysqldump -u użytkownik -p[hasło] [baza_danych] < plik.sql
```

Przykłady

1. Utworzenie kopii bezpieczeństwa wszystkich baz danych na serwerze:

```
mysqldump -u użytkownik -p hasło databases > kopiabazy.sql
```

2. Utworzenie kopii bezpieczeństwa bazy **księgarnia**:

```
mysqldump -u użytkownik -p hasło księgarnia > baza_ksiegarnia.sql
```

3. Wczytanie kopii zapasowej bazy danych z pliku baza_ksiegarnia.sql:

```
mysqldump -u użytkownik -p hasło księgarnia < …/katalog/baza_ksiegarnia.sql
```

28.3. Przyrostowa kopia bezpieczeństwa bazy danych

przyrostowa kopia bezpieczeństwa

Przyrostowa kopia bezpieczeństwa jest związana bezpośrednio z dziennikiem transakcji, w którym są zapisywane wszystkie operacje dotyczące modyfikacji bazy od momentu utworzenia ostatniej kopii zapasowej. Dzienniki zapisów są w tym przypadku traktowane jako kopie przyrostowe bazy danych. Żeby MySQL mógł zapisywać te operacje, musimy uruchomić serwer z opcją `--log-bin` lub ustawić tę opcję w pliku konfiguracyjnym.

Nowy dziennik transakcji będzie tworzony zawsze, gdy zostanie uruchomiony serwer, zostanie wykonane polecenie `FLUSH LOG` oraz gdy aktualny plik dziennika uzyska maksymalny rozmiar.

Pliki dziennika możemy usunąć poleceniem `RESET MASTER;` lub `PURGE MASTER LOGS;`.

Do zlokalizowania plików dziennika transakcji posłużymy się poleceniem `SHOW BINLOG EVENTS;`.

Do odtworzenia stanu bazy danych sprzed awarii będzie nam jednak potrzebna pełna kopia bazy wraz z dziennikami transakcji.

28.4. Odzyskiwanie danych z kopii bezpieczeństwa

Proces odzyskiwania danych z kopii bezpieczeństwa odbywa się w dwóch krokach:

Krok 1. Odzyskujemy bazę danych z pełnej kopii bezpieczeństwa (do której utworzenia użyliśmy programu mysqldump) poleceniem:

```
mysql -u [użytkownik] - p [hasło] < kopia.sql
```

Krok 2. Odzyskujemy wszystkie dane, które zostały zmodyfikowane po utworzeniu pełnej kopii bezpieczeństwa, stosując polecenie:

```
mysqlbinlog -u [użytkownik] -p [hasło] binlog.2 | mysql
```

ZADANIA DO WYKONANIA

1. Opisz różnice między atakami aktywnymi i pasywnymi oraz zachodzące między nimi relacje.
2. Wyszukaj w internecie informacje na temat programów, które mogą być użyte do przeprowadzenia ataków na bazy danych.
3. Wyszukaj w internecie informacje dotyczące tego, jakie warunki zachowania polityki bezpieczeństwa powinny być spełnione podczas administrowania bazami danych.
4. Podaj, kiedy należy wykonać pełną kopię bezpieczeństwa bazy danych, i przedstaw, jak to przebiega.
5. Wyjaśnij, na czym polega sprawdzenie spójności bazy danych.

PODSUMOWANIE LEKCJI

s. 179
- Każdy administrator bazami danych powinien zachować zasady bezpieczeństwa, które wymagają kontroli dostępu do Systemu Zarządzania Bazami Danych i do samych danych.

s. 180
- Spójność bazy danych to sprawdzenie, czy baza nie zawiera błędów. Należy to zrobić przed stworzeniem pełnej kopii bezpieczeństwa bazy.

s. 181
- Uszkodzoną tabelę naprawiamy, używając polecenia `REPAIR TABLE`. Robimy to zawsze po sprawdzeniu spójności bazy danych, a przed wykonaniem pełnej kopii bezpieczeństwa.

s. 181
- Przyrostową kopię bezpieczeństwa bazy danych wykonujemy, aby zapisać wszystkie informacje dotyczące operacji i modyfikacji w bazie danych.

PODSUMOWANIE PRZED SPRAWDZIANEM

15. Jak gromadzić informacje, czyli komputerowe bazy danych

- Elektroniczną **bazą danych** nazywa się uporządkowany zbiór informacji z określonym przez program komputerowy sposobem zapisu, wyszukiwania i dostępu do poszczególnych danych. Dane zapisywane są na nośnikach elektronicznych (np. dyskach), a dostęp do nich następuje za pomocą aplikacji komputerowej. s. 96
- Podstawowymi elementami bazy danych są **tabele**, a ich wiersze to **rekordy**. s. 96
- Powiązania między rekordami to **relacje**. s. 96

16. Tabele i formularze, czyli jak utworzyć bazę danych

- Przed przystąpieniem do tworzenia bazy danych należy sporządzić jej **projekt** i ustalić podstawowe parametry, np. liczbę tabel i kolumn w tabelach oraz rodzaj danych wprowadzanych do pól. s. 98
- W razie potrzeby należy skorzystać z opcji **pomocy** lub z publikacji na temat danego systemu baz. s. 98
- W bazie danych wszystkie informacje są zapisane w **tabelach**. s. 98
- Systemy baz danych, np. MS Access, LibreOffice Base lub OpenOffice.org Base, wspomagają tworzenie elementów bazy poprzez **kreatory**. Warto z nich korzystać. s. 100
- Tabele można wypełniać danymi bezpośrednio lub za pośrednictwem **formularzy**. s. 101

17. Relacje i pytania, czyli jak uczynić bazę użyteczną

- Każdy **obiekt bazy** po edycji powinien być zapisany i zamknięty. s. 103
- **Relacja** to połączenie logiczne pomiędzy dwiema tabelami, określające, które pola są wspólne dla tych tabel. s. 103
- W bazie Access są możliwe trzy rodzaje relacji – **jeden-do-wielu**, **wiele-do-wielu** i **jeden-do-jednego**. s. 104
- Do tworzenia relacji w bazie Access służy narzędzie **Relacje**. s. 104
- **Typy danych** w Access określają rodzaj informacji, jakie są przechowywane w bazie, np. czy to będzie tekst, czy liczba, i określają wielkość pamięci potrzebnej do ich zapamiętania. s. 107

18. Kto pyta, nie błądzi, czyli jak korzystać z baz danych Access

- **Kwerendy** stosuje się dla ułatwienia wyświetlania i zmiany danych oraz ich dodawania i usuwania z bazy. s. 107
- W kwerendach i tabelach można stosować **filtrowanie** lub **sortowanie**. **Filtrowanie** jest jednym z kryteriów zapytania. Może mieć postać formuły ułożonej z użyciem operatorów, np. <, >, =, lub wyrażeń, z którymi będą porównywane dane z pól tabel. s. 107
- **Raport** jest obiektem bazy danych przeznaczonym do prezentowania odczytanych z niej informacji. Raporty mogą być powiązane z tabelami lub kwerendami. s. 111

19. Formularze, czyli ułatwiamy sobie wprowadzanie informacji do bazy danych

- Formularz jest jednym z obiektów bazy danych, który umożliwia wprowadzanie danych oraz wyświetlanie zawartych w niej informacji. Dzięki nim możemy określić sposoby interakcji użytkowników z danymi zawartymi w bazie danych. s. 115

s. 117 ▸ • Formularz z wieloma elementami umożliwia przedstawienie formatki z wieloma rekordami jednocześnie (zwykły formularz wyświetlał tylko jeden rekord). Typ **Wiele elementów** to formularz typu tabelarycznego.

s. 118 ▸ • Arkusz danych umożliwia utworzenie formularza pokazującego wiele rekordów w arkuszu danych, po jednym rekordzie na wiersz – jest on odmianą formularza z wieloma danymi.

s. 118 ▸ • Formularz dzielony umożliwia nam utworzenie formularza, w którym w górnej sekcji pokazany jest arkusz danych, a w dolnej sekcji znajduje się formularz do wprowadzania informacji dotyczących rekordu wybranego w arkuszu danych.

s. 119 ▸ • Kreator formularza umożliwia stworzenie prostych formularzy krok po kroku.

s. 120 ▸ • Chcąc stworzyć indywidualny formularz od podstaw, całkowicie dostosowany w układzie i wyglądzie do naszych potrzeb, skorzystamy z możliwości tworzenia formularza w widoku projektu.

s. 123 ▸ • Formanty są powiązane z określonymi polami tabeli lub kwerendy (np. pole tekstowe, przyciski opcji) i umożliwiają wyświetlenie żądanych informacji oraz niezbędnych elementów, za pomocą których będzie realizowany proces przetwarzania danych w bazie.

20. Czy to możliwe, czyli baza danych w arkuszu Excel

s. 132 ▸ • W programie Excel zgromadzone dane można **sortować** zgodnie z jednym kryterium – lub wieloma.

s. 128 ▸ • Do wpisywania danych można użyć narzędzia **Formularz**. Należy go przedtem dołączyć do wstążki lub paska narzędzi **Szybki dostęp**.

s. 133 ▸ • Każdą tabelę w bazie Access można **wyeksportować** do formatu Excel. Zostanie ona zapisana jako samodzielny arkusz.

21. Wiedza w sieci, czyli internet mądrych ludzi

s. 135 ▸ • Wyszukiwarka nie przeszukuje internetu za każdym razem, gdy jej użyjesz. W swojej bazie danych zapisanej na serwerze przez **roboty indeksujące** (programy wyszukujące i segregujące informacje) przechowuje ona potrzebne ci informacje, takie jak krótki opis (metaopis) i adres strony, na której wystąpiła poszukiwana informacja.

s. 136 ▸ • Im trafniej dobierzesz słowa kluczowe, czyli takie, które opisują szukany obiekt, tym trafniejsze otrzymasz odpowiedzi.

• Istnieje wiele wyszukiwarek, np. *google.pl* lub *bing.com*. Istnieją także wyszukiwarki gromadzące informacje o dokumentach z jednej dziedziny, np. *europeana.eu*, w której znajdziesz zbiory kultury europejskiej.

s. 136 ▸ • W przeglądarce można ustawić **wyszukiwarkę domyślną**.

s. 137 ▸ • Dzięki opcjom wyszukiwania zaawansowanego można wyeliminować strony o niechcianej treści lub rodzaju.

• Niektóre wyszukiwarki pozwalają na wyszukiwanie obrazem.

s. 143 ▸ • Darmowe zasoby, np. grafika, zdjęcia lub dźwięki, są udostępniane na zasadzie licencji **CC Creative Commons**.

s. 143 ▸ • Nie wierz we wszystko, co znajdziesz w sieci. Sprawdzaj wiarygodność informacji.

22. Wirtualne serwery, czyli instalujemy Apache

- Aby tworzyć strony WWW oparte na systemach zarządzania treścią lub wykorzystujące języki programowania interpretowane po stronie serwera, potrzebny jest serwer lokalny. ◂ s. 145
- Przykładem serwera lokalnego, który umożliwia nam instalację stron opartych na systemach zarządzania treścią i pracy z bazą danych, jest Apache, który wchodzi w skład pakietu XAMPP. ◂ s. 146

23. Język zapytań, czyli poznajemy język SQL

- SQL jest strukturalnym językiem zapytań powstałym w latach 70. XX wieku w firmie IBM. Używa się go do tworzenia i modyfikowania baz danych oraz umieszczania i pobierania danych z baz. ◂ s. 150
- Praca z językiem SQL polega na zadawaniu zapytań bazie danych. Można je zaliczyć do czterech podzbiorów. Są to: ◂ s. 150
  - SQL DML – język manipulacji danymi,
  - SQL DDL – język definicji danych,
  - SQL DCL – język kontroli nad danymi,
  - SQL DQL – język definiowania zapytań.

24. Zadajemy pytania, czyli kwerendy w języku SQL

- MySQL jest systemem zarządzania relacyjnymi bazami danych, wykorzystującym język SQL, którego kod źródłowy jest wydawany na podstawie licencji otwartego oprogramowania. ◂ s. 153
- W narzędziu, jakim jest phpMyAdmin, tabelę możemy stworzyć za pomocą kreatora tworzenia tabel, a także wpisując kod języka SQL. ◂ s. 153
- Zapytanie wyszukujące możemy tworzyć, wykorzystując kod języka SQL, oraz za pomocą zakładki **Szukaj** w phpMyAdmin. ◂ s. 160

25. Więcej pytań, czyli jeszcze o kwerendach w języku SQL

- Polecenie `WHERE` wykorzystujemy przy tworzeniu zapytania z określonymi kryteriami wyszukiwania. ◂ s. 165
- Aby uzyskać dane z kilku tabel za pomocą języka SQL, należy wykonać operację nazywaną łączeniem. Dzięki niej scalimy dwie tabele (lub więcej) i będziemy mogli odnaleźć interesujące nas informacje. ◂ s. 166
- `LEFT JOIN` pozwala nam wyszukać niepołączone ze sobą dane. ◂ s. 169

26. Podzapytania, czyli instrukcje modyfikujące dane w języku SQL

- Podzapytania są zagnieżdżone w innym zapytaniu i łatwiejsze w konstrukcji niż używanie połączeń i tabel tymczasowych. ◂ s. 169
- Stosując podzapytania podstawowe, korzystamy z czterech operatorów: `ANY`, `IN`, `SOME` i `ALL`. ◂ s. 170
- Podzapytanie skorelowane wykonuje się wielokrotnie – tyle razy, ile rekordów przegląda zapytanie zewnętrzne. ◂ s. 170
- Operatory `EXISTS` i `NO EXISTS` są stosowane tylko podczas używania podzapytań skorelowanych. ◂ s. 171
- Podzapytanie wierszowe zwraca jeden wiersz i tym samym może zwrócić więcej niż jedną wartość kolumny. ◂ s. 171

s. 172 ▸ • Podzapytanie o charakterze tabeli tymczasowej umieszczamy w klauzuli `FROM` zapytania zewnętrznego. Dzięki temu możemy wykonać zapytanie na wynikach podzapytania, które jest traktowane jak tabela tymczasowa.

s. 172 ▸ • Alias nadaje się, aby ułatwić sobie pracę z tabelami w bazie danych. W niektórych sytuacjach, na przykład podczas stosowania podzapytań o charakterze tabeli tymczasowej, stosowanie aliasów jest koniecznością.

27. Modyfikacje, czyli zarządzanie bazą danych

s. 174 ▸ • Korzystając z polecenia `UPDATE`, jesteśmy w stanie zmienić w bazie danych informacje dotyczące zarówno pojedynczych rekordów, jak i całych kolumn.

s. 176 ▸ • Dzięki poleceniu `ALERT TABLE` jesteśmy w stanie zmienić strukturę całej tabeli. Możemy dodać lub usunąć jedną lub więcej kolumn.

s. 177 ▸ • Aby usunąć pojedyncze rekordy, stosujemy składnię: `DELETE FROM` tabela `WHERE` warunek.

s. 178 ▸ • Do usunięcia danych i definicji tabeli lub całych baz danych stosujemy polecenie: `DROP`.

28. Bezpieczeństwo bazy, czyli tworzymy kopie zapasowe

s. 179 ▸ • Każdy administrator bazami danych powinien zachować zasady bezpieczeństwa, które wymagają kontroli dostępu do Systemu Zarządzania Bazami Danych i do samych danych.

s. 180 ▸ • Spójność bazy danych to sprawdzenie, czy baza nie zawiera błędów. Należy to zrobić przed stworzeniem pełnej kopii bezpieczeństwa bazy.

s. 181 ▸ • Uszkodzoną tabelę naprawiamy, używając polecenia `REPAIR TABLE`. Robimy to zawsze po sprawdzeniu spójności bazy danych, a przed wykonaniem pełnej kopii bezpieczeństwa.

s. 181 ▸ • Przyrostową kopię bezpieczeństwa bazy danych wykonujemy, aby zapisać wszystkie informacje dotyczące operacji i modyfikacji w bazie danych.



V. Programowanie stron internetowych

- HTML, czyli przeglądarka interpretuje język programowania stron
- Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony
- Grafika, czyli kolejna składowa stron internetowych w HTML
- Składnia stylów, czyli jak CSS pomaga w programowaniu strony
- Pliki stylów, czyli CSS w akcji
- Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej
- Widoczna w internecie, czyli jak opublikować stronę
- CMS, czyli system zarządzania treścią strony internetowej
- Panel i skórki, czyli tworzymy stronę w CMS
- Szybko i łatwo, czyli programy do tworzenia stron internetowych
- Pierwsze skrypty, czyli poznajemy język PHP
- Stałe i zmienne, czyli typy danych w języku PHP
- Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP
- Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP
- Funkcje w języku PHP, czyli jak wzbogacić naszą stronę
- Tablice, czyli jak odzyskać dane w języku PHP
- Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP

29. HTML, czyli przeglądarka interpretuje język programowania stron

NA TEJ LEKCJI:

- dowiesz się, czym są HTML i CSS;
- poznasz podstawowe znaczniki i budowę dokumentu HTML;
- zrozumiesz, jakie znaczenie dla wyglądu strony ma CSS.

HTML i CSS. Każdy coś o nich słyszał, ale czy wiesz, czym tak naprawdę są? Czy to języki programowania, a jeśli tak, to jak uruchamia się programy? A może jest inaczej, niż ci się wydaje? Ten rozdział rozwieje twoje wątpliwości.

29.1. Co interpretuje przeglądarka, czyli czym jest dokument HTML

Nasze rozważania zacznijmy od definicji.

HTML ▸ **HTML** (ang. *hypertext markup language*, czyt. hajpertekst maraup lengłidż) to **hipertekstowy język znaczników**. Nie jest językiem programowania, ponieważ nie układa się w nim programów komputerowych. Służy do opisu stron internetowych za pomocą znaczników i tekstu. Przeglądarka interpretuje język znaczników i na tej podstawie wyświetla strony.

Pierwsza wątpliwość rozwiana. HTML nie jest językiem programowania, chociaż często w języku potocznym tak się o nim mówi. W uproszczeniu można powiedzieć, że opisuje on za pomocą tekstu wygląd i zawartość stron internetowych. Często mówi się także, że ten tekst to program stron. Stąd nieporozumienia związane z interpretacją nazwy.

znaczniki ▸ Skoro jest to język znaczników, warto wiedzieć, czym one są. W tekście łatwo je odnaleźć, bo wpisuje się je pomiędzy znaki `<  >`. Rozróżniamy dwa ich rodzaje: **znaczniki otwarcia** `<  >` i **znaczniki zamknięcia** `</  >`. Jeśli postanowisz użyć któregoś z nich, np. w odniesieniu do tekstu, to przed nim umieścisz odpowiedni znacznik w `<  >`, a na jego końcu w `</  >`. Dzięki znacznikom można określać sposób wyświetlania obiektów na stronie.

Przykład

Tekst *kilka słów* na stronie ma się pojawić pogrubiony. Należy więc go umieścić pomiędzy znacznikami: `<b>kilka słów</b>`. Inne znaczniki poznasz w dalszej części podręcznika.

W nazwie HTML występuje też słowo **hipertekstowy**. Dlaczego? Zapewne wiesz, czym są linki (odsyłacze) na stronach internetowych. To, że działają, jest

wynikiem działania hipertekstu. Dzięki temu sieć stron internetowych może funkcjonować, a użytkownik – odnaleźć żądaną stronę.

Jeśli opis strony w HTML jest prawidłowy, to wynik jego działania można zobaczyć w przeglądarce. To ona dokonuje interpretacji tekstu HTML i zgodnie z nim wyświetla stronę w swoim oknie.

W informatyce następuje ciągły postęp. Nic dziwnego, że i HTML się rozwija. Najnowsze informacje na temat szczegółowych zagadnień związanych z HTML znajdziesz na stronie **World Wide Web Consortium** (czyt. łorld łajd łeb konsortjom) lub **Web Hypertext Application Technology Working Group** (czyt. łeb hajpertekst aplikejszyn teknolodżi łorkin grup). Są to organizacje odpowiedzialne za rozwój języka znaczników.

W naszych rozważaniach będziemy używać składni języka **HTML5**.

Gdy program strony jest już ułożony, należy go zapisać w odpowiednim formacie (rozszerzenie .HTML). Jeśli umieścisz go na serwerze WWW pod pewnym adresem, powinien on otrzymać nazwę **index.html**. Przeglądarka od niego właśnie zacznie wczytywanie i wyświetlanie strony. Oczywiście inne pliki z kodem HTML składające się na stronę, np. wyświetlające podstronę, możesz nazwać dowolnie, zachowując rozszerzenie.

29.2. Kaskadowe arkusze stylów, czyli określamy wygląd strony

HTML opisuje głównie treść strony. Gdyby w nim opisywać także formę prezentacji np. kolory, tła itp., teksty kodu stałyby się bardzo obszerne. Z każdym nowym sposobem wyświetlania, animacji itp. trzeba by dodawać do HTML nowe znaczniki. Wymuszałoby to także częste zmiany w przeglądarkach, by ujednolicić wyświetlanie stron. Dlatego postanowiono, żeby oddzielić warstwę treści od formy, i opracowano CSS.

CSS (*Cascading Style Sheets*, czyt. kaskejdin stajl szits) to język opisujący reguły, według których przeglądarka wyświetla strony WWW. ◂ CSS

Arkusz stylów składa się z reguł. Każda z nich określa styl dla danych fragmentów dokumentu HTML. Tej samej reguły można używać dla wielu fragmentów. **Reguła** składa się z selektora oraz deklaracji i ma formę:

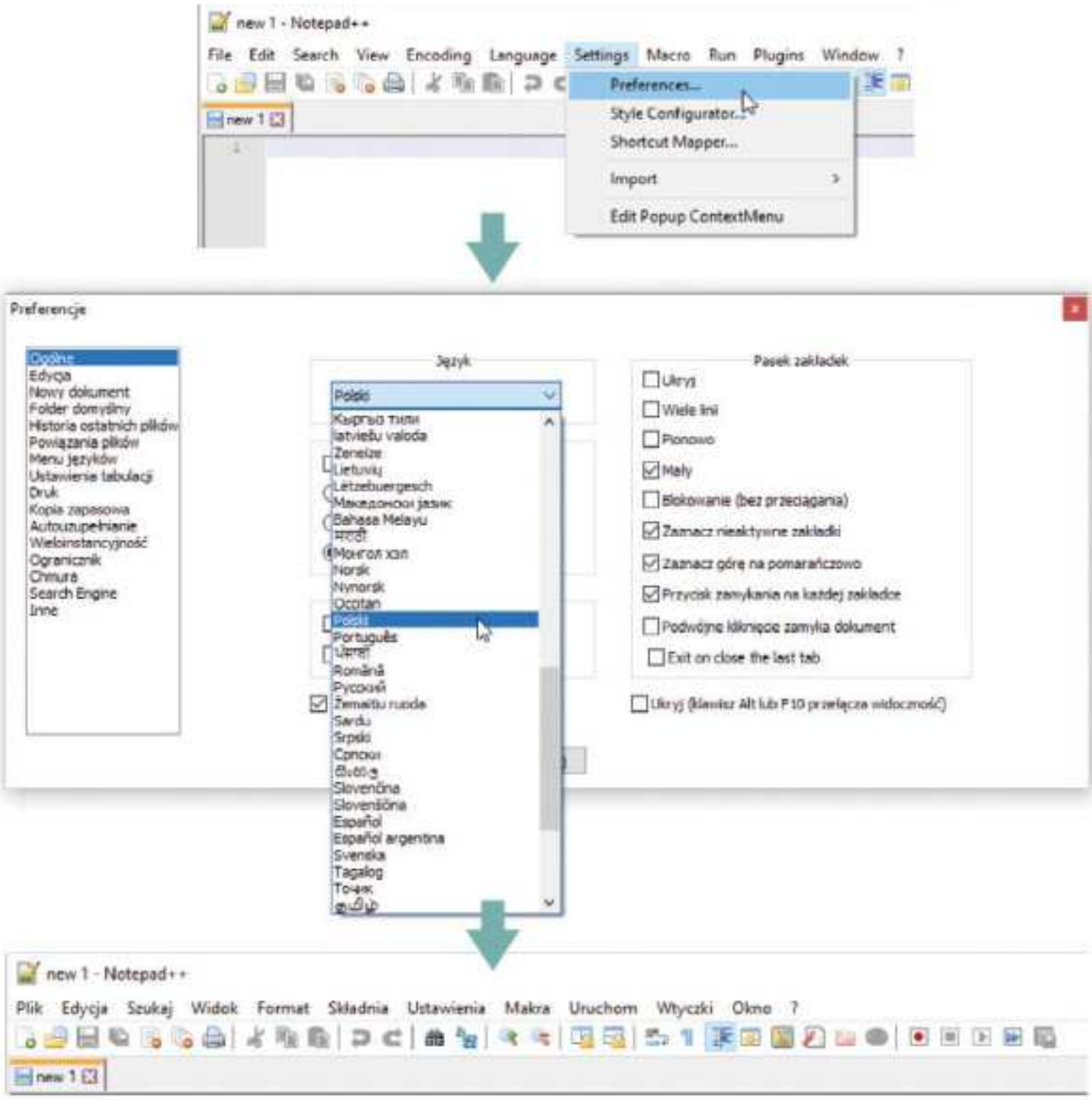
selektor {właściwość: wartość}

Pomocne są także biblioteki zwane **CSS Frameworks** (czyt. frejmłorks) ułatwiające definiowanie stylów dla tworzonych stron.

29.3. Jaki edytor, czyli wybieramy program do edycji HTML

Ponieważ HTML to tekst, do jego utworzenia jest wymagany edytor. Można użyć notatnika lub innego prostego programu. Warto jednak znaleźć edytor, który oprócz możliwości pisania ma dodatkowe funkcje, np. zaznaczanie kolorem znaczników,

liczb itp. W sieci znajdziecie ich wiele. W naszych przykładach będziemy używać darmowego **Notepad++** (czyt. nołtpad). Możesz go pobrać z internetu. Dla ułatwienia skonfiguruj go tak, by menu było opisane w języku polskim (rys. 29.1.).



Rys. 29.1. Polskie menu w edytorze Notepad++

29.4. Używamy znaczników, czyli pierwsze kroki w HTML

Zanim zabierzesz się do pracy, musisz poznać **strukturę dokumentu HTML** (rys. 29.2.). Każdy dokument HTML musi zawierać znaczniki i treści pomiędzy nimi.

struktura dokumentu HTML



Rys. 29.2. Struktura dokumentu HTML

Otwórz edytor i wpisz do niego treść dokumentu HTML (rys. 29.3.).

```
<!DOCTYPE html>
<html>
      <head>
          <meta charset="UTF-8">
      </head>
      <body>
          Ten tekst wyświetli przeglądarka
      </body>
</html>
```

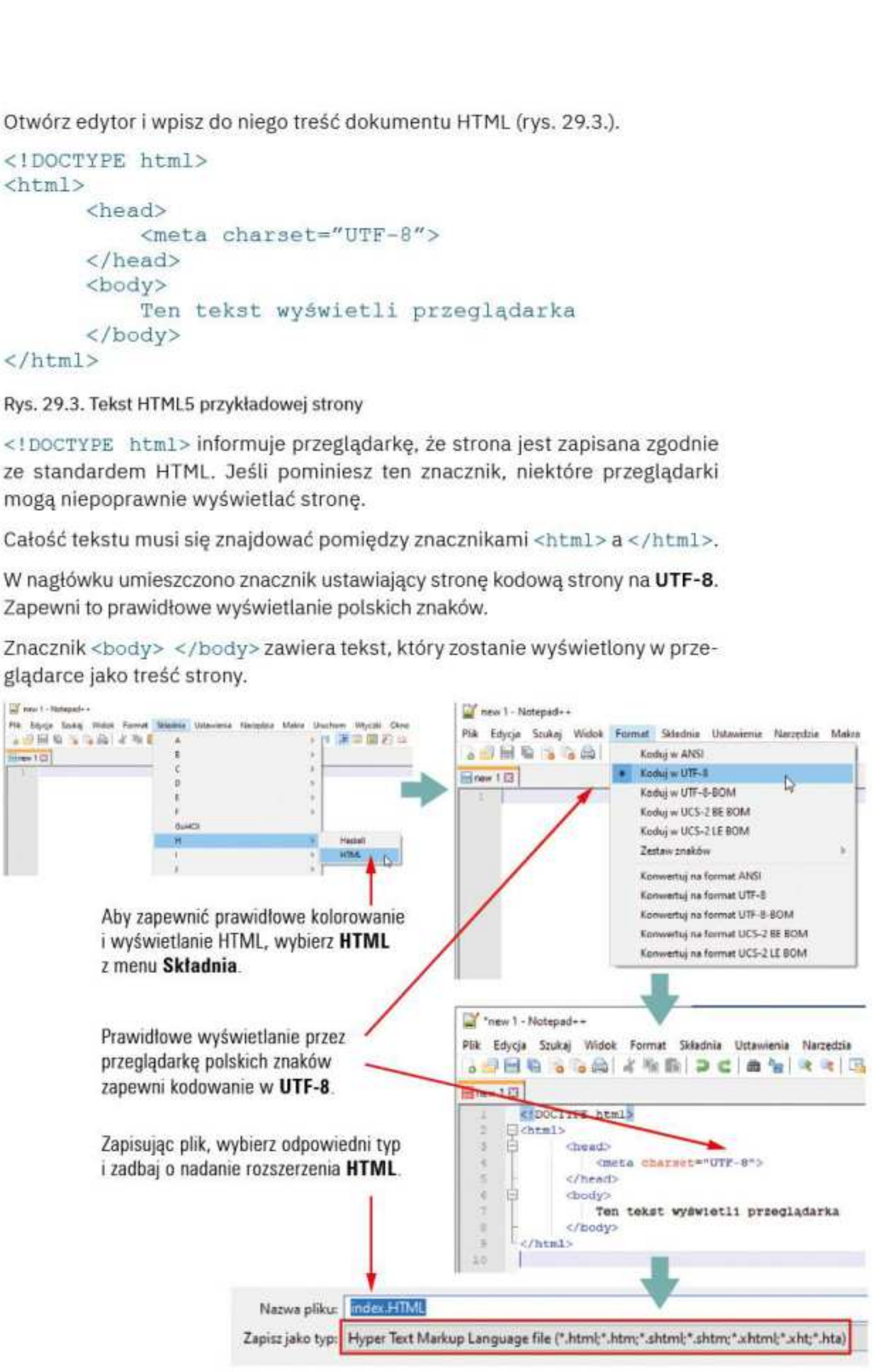
Rys. 29.3. Tekst HTML5 przykładowej strony

`<!DOCTYPE html>` informuje przeglądarkę, że strona jest zapisana zgodnie ze standardem HTML. Jeśli pominiesz ten znacznik, niektóre przeglądarki mogą niepoprawnie wyświetlać stronę.

Całość tekstu musi się znajdować pomiędzy znacznikami `<html>` a `</html>`.

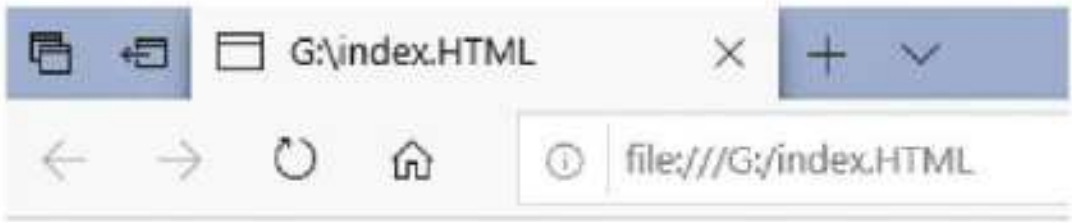
W nagłówku umieszczono znacznik ustawiający stronę kodową strony na **UTF-8**. Zapewni to prawidłowe wyświetlanie polskich znaków.

Znacznik `<body> </body>` zawiera tekst, który zostanie wyświetlony w przeglądarce jako treść strony.



Rys. 29.4. Edycja tekstu HTML5 w Notepad++

Plik z twoją stroną po podwójnym kliknięciu powinien otworzyć się w przeglądarce domyślnej dla systemu (rys. 29.5.).



Rys. 29.5. Efekt interpretacji pliku HTML z przykładu przez przeglądarkę

Pełną listę znaczników HTML oraz opis ich stosowania i efekty, jakie przynoszą na stronie, znajdziesz w internecie. Niektóre z nich poznasz w przykładach z następnych rozdziałów. Warto jednak zwrócić uwagę na najczęściej używane znaczniki, które zaczęły funkcjonować w wersji języka HTML5 (tab. 29.1.).

Znacznik	Efekt działania
`<section> </section>`	Tworzy nową sekcję treści wyświetlanej na stronie. Można ją porównać do podziału dokumentu w Wordzie za pomocą Ctrl+Enter.
`<header> </header>`	Tworzy nagłówek strony lub sekcji.
`<article> </article>`	Umieszcza na stronie niezmienną część pewnej treści, np. temat dyskusji przed komentarzami wprowadzanymi przez czytelników.
`<aside> </aside>`	Pozwala na wyświetlanie treści dodatkowych, np. jako sidebar (boczny, wysuwany pasek na stronie internetowej).
`<nav> </nav>`	Pozwala utworzyć główne menu strony do nawigacji po najważniejszych jej treściach.
`<footer> </footer>`	Tworzy stopkę strony, np. zawierającą informacje o właścicielu strony i jej twórcy.
`<main> </main>`	Wskazuje główną treść strony. Dzięki temu strona w telefonie wyświetli najpierw to, co jest na niej najważniejsze, bez konieczności sięgania do menu.
`<figure> </figure>`	Wyświetla ilustracje itp.
`<figcaption>`	Umieszczony wewnątrz `<figure> </figure>` wstawia na stronę podpis obiektu.
`<h1>, <h2>, <h3>, <h4>, <h5>, <h5>`	Znaczniki nagłówków znane z wcześniejszych wersji HTML. W HTML5 można jednak używać więcej niż jednego nagłówka na stronie.

Tab. 29.1. Najczęściej używane znaczniki dodane w wersji HTML5

ZADANIA DO WYKONANIA

1. Zainstaluj w swoim komputerze Notepad++ i skonfiguruj do pisania kodów strony w HTML.
2. Wyszukaj w sieci informacje na temat darmowych edytorów HTML. Zbierz wiarygodne opinie na ich temat i porównaj je z Notepad++. Sformułuj wnioski z porównania.
3. Za pomocą dowolnego edytora wykonaj tablicę, na której opiszesz znaczniki HTML5. Załóż, że zostanie ona powieszona w sali komputerowej i w zmniejszonej formie wklejona do zeszytu.
4. Znajdź w sieci opis podstawowych znaczników HTML5 lub skorzystaj z tablicy z zadania 3. i zredaguj tekst HTML, który wyświetli w oknie przeglądarki tekst o różnej wielkości.

PODSUMOWANIE LEKCJI

- **HTML** (ang. *hypertext markup language*) to hipertekstowy język znaczników. Nie jest językiem programowania, ponieważ nie układa się w nim programów komputerowych. ◂ s. 188
- Rozróżniamy dwa rodzaje znaczników: **znaczniki otwarcia** < > oraz **znaczniki zamknięcia** </ >. ◂ s. 188
- **CSS** (*Cascading Style Sheets*) to język opisujący reguły, według których przeglądarka wyświetla strony WWW. ◂ s. 189

30. Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony

NA TEJ LEKCJI:

- użyjesz list w dokumencie HTML i dowiesz się, w jakich przypadkach je stosować;
- ustanowisz odnośniki i poznasz różnice pomiędzy ich rodzajami.

Wiesz już, jak wygląda struktura dokumentu HTML. Twój pierwszy tekst wyświetlił napis w przeglądarce. Poznałeś/Poznałaś więc mechanizm prezentujący strony – hipertekst interpretowany przez przeglądarkę. W konstrukcji strony internetowej można użyć jeszcze kilku konstrukcji, np. tabeli, list.

30.1. Grupujemy treści na stronie, czyli listy w HTML

Listy mogą służyć do uporządkowania informacji na stronie lub tworzenia menu strony.

lista ▸ Użycie **list** w HTML porządkuje treści. Listę można wypunktować tak jak w zwykłym tekście dokumentu tekstowego. Należy w tym celu wymienić treści po kolei, np. składniki potrawy. Listy mogą służyć do uporządkowania informacji na stronie lub tworzenia menu strony.

Rozróżniamy trzy rodzaje list.

Lista uporządkowana jest zawarta w znaczniku `<ol> </ol>` (ang. *ordered list*, czyt. orderd list), w której numeruje się poszczególne treści (rys. 30.1.).

Lista uporządkowana

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
    </head>

    <body>
        <ol>
            <li>Pierwszy element listy uporządkowanej</li>
            <li>Drugi element listy uporządkowanej</li>
        </ol>
    </body>
</html>
```

1. Pierwszy element listy uporządkowanej
2. Drugi element listy uporządkowanej

Rys. 30.1. Przykład tworzenia listy uporządkowanej w HTML

Lista nieuporządkowana (ang. *unordered list*, czyt. anorderd list) jest zawarta w znaczniku `<ul> </ul>` . Jako wyznacznik kolejnej pozycji listy występuje tu punktor (rys. 30.2.).

Lista nieuporządkowana

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
    </head>

    <body>
        <ul>
            <li>Pierwszy element listy nieuporządkowanej</li>
            <li>Drugi element listy nieuporządkowanej</li>
        </ul>
    </body>
</html>
```

- Pierwszy element listy nieuporządkowanej
- Drugi element listy nieuporządkowanej

Rys. 30.2. Przykład tworzenia listy nieuporządkowanej w HTML

Powyższe znaczniki tworzą listę, ale do jej wypełnienia używa się znacznika `<li> </li>` (ang. *list item*, czyt. list ajtem).

Lista definicji (ang. *definition list*, czyt. definiszyn list) jest zawarta w znaczniku `<dl> </dl>`. Poszczególnym pozycjom z listy przypisuje się definicje. W działaniu ta lista przypomina słowniczek.

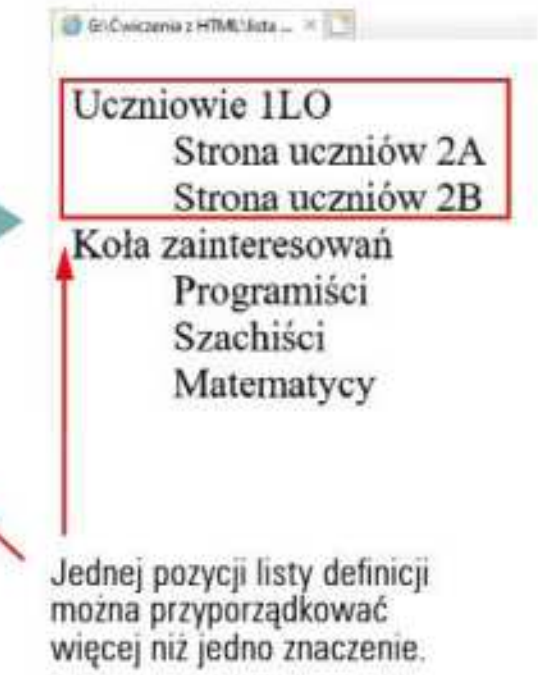
Elementy tej listy umieszcza się w znacznikach `<dt> </dt>`, które definiują nazwę pojęcia (ang. *definition term*, czyt. definiszyn term), a także w znacznikach `<dd> </dd>` definiujących opis pojęcia (ang. *definition description*, czyt. definiszyn deskrypszyn).

Lista definicji

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
    </head>

    <body>
      <dl>
        <dt>Uczniowie 1LO</dt>
        <dd>Strona uczniów 2A</dd>
        <dd>Strona uczniów 2B</dd>
        <dt>Koła zainteresowań</dt>
        <dd>Programiści</dd>
        <dd>Szachiści</dd>
        <dd>Matematycy</dd>
      </dl>
    </body>
</html>
```



Rys. 30.3. Przykład tworzenia listy definicji w HTML

Jeśli przejrzysz tabelę z poprzedniego rozdziału (tab. 29.1.), znajdziesz tam znaczniki, które wprowadzają nagłówki. Zastosujmy je do listy z rysunku 30.3. Zwróć uwagę na to, że znacznik `<h3> </h3>` (możesz użyć dowolnego z zestawu nagłówków) w przykładzie obejmuje tylko nazwy pojęć.

```
<!DOCTYPE html>
<html>

    <head>
        <meta charset="UTF-8">
    </head>

    <body>
      <dl>
        <h3>
        <dt>Uczniowie 1LO</dt>
        </h3>
        <dd>Strona uczniów 2A</dd>
        <dd>Strona uczniów 2B</dd>
        <h3>
        <dt>Koła zainteresowań</dt>
        </h3>
        <dd>Programiści</dd>
        <dd>Szachiści</dd>
        <dd>Matematycy</dd>
      </dl>
    </body>
</html>
```



Rys. 30.4. Przykład tworzenia listy definicji w HTML z dodanymi nagłówkami

30.2. Nie tylko w bazie są ważne, czyli jak tworzyć tabele w HTML

tabele w HTML ▸ **Tabele**, podobnie jak listy, tworzy się w celu uporządkowania danych wyświetlanych na stronach. Do ich utworzenia służy znacznik `<table> </table>`. Niestety, nie może on funkcjonować bez znaczników dodających w tabeli wiersze `<tr> </tr>` oraz komórki `<td> zawartość komórki </td>`.

Korzystając z wielu dostępnych atrybutów, można zbudować dowolne tabele. Prześledź działanie następującego kodu HTML (rys. 30.5.). W przykładowej tabeli umieszczono tekst, ale mogą się w niej znaleźć także innego typu elementy, np. grafika. Pełni ona funkcję typowej tabeli podobnej do tych, które są stosowane w dokumentach tekstowych. Tabele w HTML odgrywają jeszcze inną rolę. Pozwalają na rozmieszczanie poszczególnych bloków strony. Wtedy oczywiście nie używa się obramowań.

Atrybut **border** powoduje wyświetlenie obramowań pól tabeli.

Znacznik **<tr> </tr>** tworzy pierwszy wiersz tabeli. Użyte wewnątrz tego znacznika znaczniki **<th> </th>** tworzą pola tabeli, które będą jej nagłówkami. Atrybut **sope="col"** informuje o zakresie działania nagłówka (cała kolumna).

W kolejnym wierszu nie ma już nagłówków, więc jego komórki opisujemy znacznikiem **<td> </td>**. Atrybut **rowspan="3"** pozwala utworzyć połączenie kilku komórek w kolumnie. W tym przypadku trzech kolejnych w kolumnie drugiej i dwóch kolejnych w kolumnie piątej.

Atrybut **align="center"** powoduje wyświetlenie elementu tabeli pośrodku pola.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
    </head>
    <body>

<table border>

  <tr>
    <th scope="col">Uczeń</th>
    <th scope="col">Szkoła</th>
    <th scope="col">Klasa</th>
    <th scope="col">Wynik</th>
    <th scope="col">Olimpiada</th>
  </tr>

  <tr>
    <td>Kowalski Jan</td>
    <td rowspan="3" align="center">1LO</td>
    <td align="center">3B</td>
    <td>Laureat</td>
    <td rowspan="2">OWT</td>
  </tr>

  <tr>
    <td>Tablicowski Włodzimierz</td>
    <td align="center">3A</td>
    <td>Finalista</td>
  </tr>

  <tr>
    <td>Nowak Krystyna</td>
    <td align="center">4C</td>
    <td>Finalistka</td>
    <td>Olimpiada Informatyczna</td>
  </tr>

</table>

    </body>
</html>
```

Uczeń	Szkoła	Klasa	Wynik	Olimpiada
Kowalski Jan	1LO	3B	Laureat	OWT
Tablicowski Włodzimierz		3A	Finalista	
Nowak Krystyna		4C	Finalistka	Olimpiada Informatyczna

Rys. 30.5. Przykładowa tabela w HTML

30.3. Istota hipertekstu, czyli tworzymy odnośniki

Nie ma chyba internauty, który nie używa odnośników. Bez nich serfowanie po sieci byłoby po prostu niemożliwe. Czas poznać sposoby wprowadzania odnośników do dokumentu HTML, by nasza strona mogła się odnosić do podstron lub innych stron w sieci.

Odnośnik (ang. *hyperlink*, czyt. hajperlink) po kliknięciu powoduje wczytanie się strony o danym adresie (link zewnętrzny) albo fragmentu tej samej strony lub podstrony (link wewnętrzny). ◂ odnośnik

Załóżmy, że piszesz artykuł i chcesz go umieścić na stronie. Często w takich publikacjach autorzy odwołują się do innych źródeł, np. stron urzędów lub encyklopedii, słowników itp. Wówczas wygodnie jest oznaczyć (np. kolorem) pewien fragment tekstu, np. nazwę źródła informacji, i spowodować, by kliknięcie go doprowadziło do wczytania strony, do której się odnosimy. Domyślasz się, że należy użyć znacznika – w tym przypadku `<a> </a>`, który obejmuje fragment tekstu mający funkcjonować na stronie jako odnośnik. Konieczne będzie także zastosowanie atrybutu `href="adres odnośnika"` (rys. 30.5.).

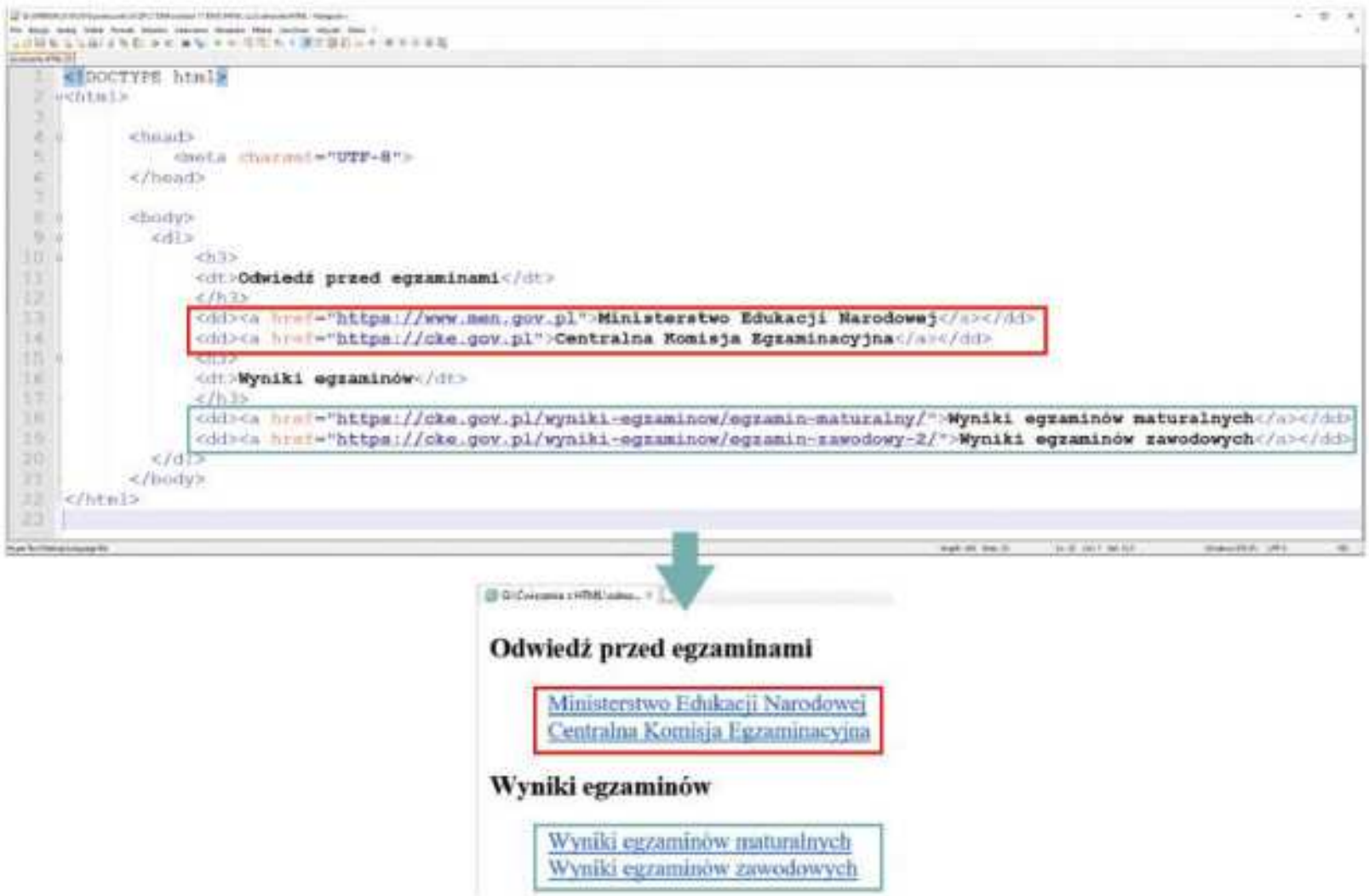
Format użycia znacznika `<a> </a>`:

```
<a href="odnośnik">tekst odnośnika</a>
```

Uwaga!

Adres odnośnika w linku powinien być zapisany w pełnej wersji, np. dla strony Ministerstwa Edukacji Narodowej: *https://www.gov.pl/web/edukacja*, a nie *www.gov.pl/web/edukacja*.

W poniższym przykładzie połączymy **listę definicji** z **odnośnikami zewnętrznymi**, tworząc menu (rys. 30.6.). Zwróć uwagę na kolejność występowania znaczników i miejsca ich zamknięcia.



Rys. 30.6. Przykład użycia odnośników w liście definicji w dokumencie HTML

Oprócz atrybutu znacznika `href` można użyć jeszcze innych. Jeśli chcesz, by strony, do których prowadzą odnośniki z rysunku 30.6., otwierały się na nowej karcie przeglądarki, użyj `target="_blank"`. Na przykład w dokumencie z rysunku 30.5. w linii 13 można by wprowadzić następujące zmiany:

```
d><a href="https://www.men.gov.pl" target="_blank">
Ministerstwo Edukacji Narodowej</a></dd>
```

Oczywiście analogiczne zmiany możesz przeprowadzić w pozostałych odnośnikach przykładu.

Ciekawe zastosowanie mają także odnośniki wewnętrzne, które mogą się odnosić do miejsca na danej stronie lub do podstrony. Dzięki nim można nawigować po obszernych stronach. Domyślasz się zapewne, że potrzebny będzie kolejny atrybut, ale to nie wszystko. Dany element strony należy oznakować identyfikatorem `id`.

Uwaga!

Atrybut id można nadać tylko jednemu elementowi danego dokumentu HTML.

Jeśli przez znacznik `<p> </p>` utworzymy paragraf naszego tekstu na stronie, to nadanie mu atrybutu `id` może wyglądać następująco:

```
<p id="rozdział1">tytuł rozdziału pierwszego</p>
```

Użycie w HTML `id` do nawigowania po stronie może mieć teraz postać:

```
<a href="#rozdział1">Rozdział pierwszy</a>
```

Kliknięcie w tekst odnośnika **Rozdział pierwszy** spowoduje „przewinięcie" strony do rozdziału 1.

Odnośnik wewnętrzny może również przenosić do podstron. Tym razem nie potrzebujesz nowego znacznika. Musisz jednak dokładnie poznać miejsce pliku HTML (ścieżkę dostępu), do którego chcesz się odnieść.

Główny plik HTML **index.html** strony internetowej umieszcza się na serwerze w folderze głównym konta. Pozostałe pliki, np. z ilustracjami, HTML podstron itp. dla porządku zapisuje się w odpowiednich folderach.

Załóżmy, że podstrona, której plik HTML nazwano **index2.html**, znajduje się w folderze **o_nas**. Wtedy znacznik `<a> </a>` odsyłający do tej podstrony będzie miał postać:

```
<a href="o_nas/index2.html">O nas</a>
```

Argumentem atrybutu `href` jest więc ścieżka dostępu do pliku wraz z jego pełną nazwą `o_nas/index2.html`.

ZADANIA DO WYKONANIA

1. Zaprojektuj prostą stronę internetową, która wyświetli listę uporządkowaną zawierającą spis twoich przedmiotów szkolnych ułożoną alfabetycznie. Poprzedź ją nagłówkiem. Dobierz odpowiedni znacznik nagłówka.
2. Znajdź w internecie kilka stron związanych z aktualnie przerabianymi tematami z wybranych przedmiotów z listy z zadania 1. Zmodyfikuj HTML z zadania 1., dodając odnośniki, które otworzą znalezione strony.
3. Zbuduj prostą stronę, na której umieścisz listę definicji zgodną ze spisem treści twojego podręcznika do informatyki. Uwzględnij kilka wybranych rozdziałów w każdym dziale. Użyj nagłówków dla odróżnienia ich tytułów.
4. Znajdź w internecie kilka stron związanych z wybranymi rozdziałami uwzględnionymi z listą definicji zadania 3. Zmodyfikuj HTML z zadania 3., dodając odnośniki, które otworzą każdą znalezioną stronę w nowej karcie przeglądarki.

5*. Utwórz trzy proste strony internetowe: jedną z listą nieuporządkowaną dwóch tytułów, a drugą i trzecią – z fragmentami lektur aktualnie omawianych na lekcjach języka polskiego. Utwórz folder **c:/lektury** i zapisz w nim pierwszy plik HTML **index.html** z listą. Utwórz jeszcze dwa foldery **c:/lektury/lektura1** i **c:/lektury/lektura2**. Zapisz w nich odpowiednio dwa pozostałe pliki HTML. W pliku **index.html** dodaj do listy odnośniki lokalne do stron z fragmentami lektur. Przetestuj działanie strony.

6. Zmodyfikuj tabelę z przykładu (rys. 30.5.) dla uczniów chodzących do dwóch różnych szkół i startujących w jednej olimpiadzie, np. olimpiadzie wiedzy technicznej.

PODSUMOWANIE LEKCJI

- **Lista** w HTML porządkuje treści. Istnieją trzy różne listy: **uporządkowana**, **nieuporządkowana** i **lista definicji**. ◂ s. 194
- **Tabele**, podobnie jak listy, pomagają porządkować dane. ◂ s. 196
- Użycie **znacznika nagłówków** pozwala wyróżnić tekst z nagłówkiem, np. listy.
- **Odnośniki** tworzy się za pomocą znacznika `<a> </a>`. Można także używać atrybutów: `href=` do podania adresu odnośnika i `target="blank"` do wyświetlenia odnośnika w nowej karcie przeglądarki. ◂ s. 197

31. Grafika, czyli kolejna składowa stron internetowych w HTML

NA TEJ LEKCJI:

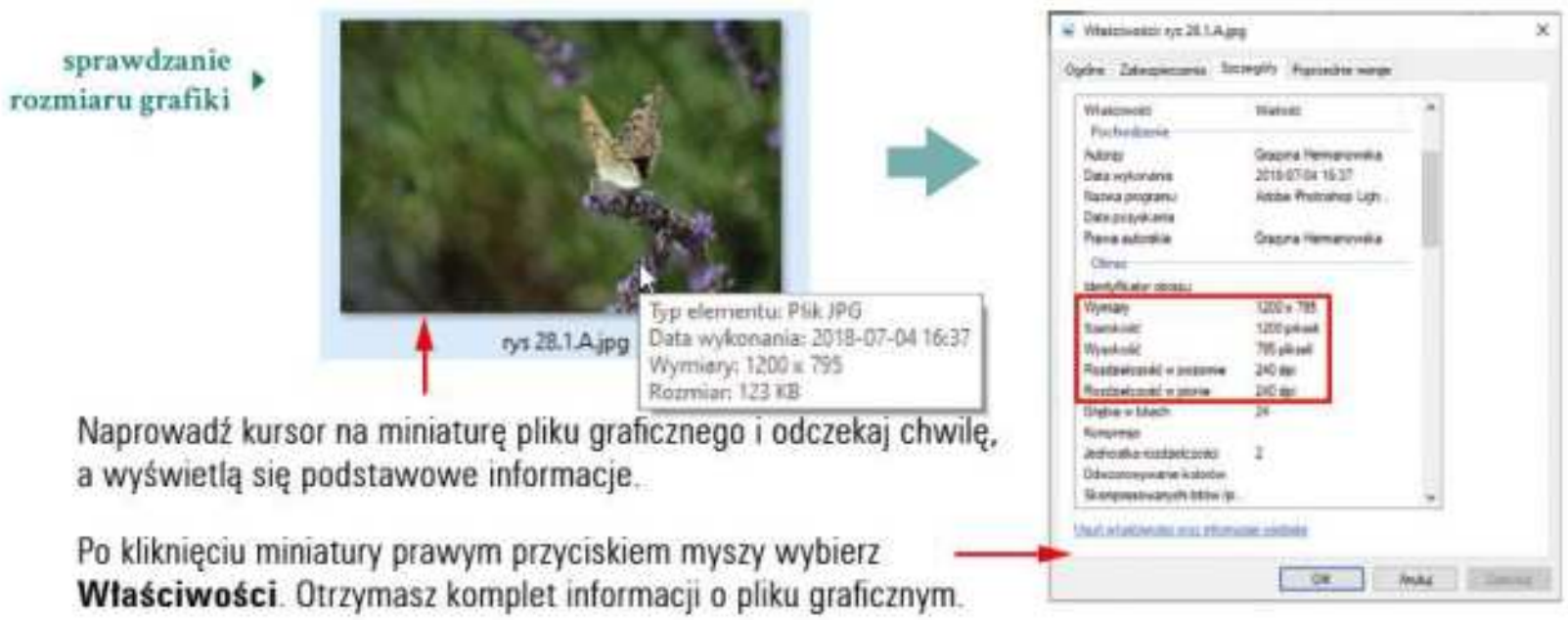
- wstawisz na stronę zdjęcie lub grafikę;
- dowiesz się, jak sprawdzić i dobrać rozmiar grafiki do prezentacji na stronie;
- utworzysz ilustrację będącą odnośnikiem do innej strony.

Niewiele jest stron pozbawionych grafiki. Zdjęcia, animacje i grafika są nieodzownym elementem każdej publikacji internetowej. Zanim przystąpisz do wstawiania obiektów graficznych do stron utworzonych w HTML, poznaj niektóre z zasad. Oprócz umiejętności umieszczania grafiki na stronie liczy się także przygotowanie pliku graficznego. Jak je przygotować i jak umieszczać na stronie? To nie jest trudne.

31.1. Przyspieszamy wczytywanie, czyli zasady przygotowania plików graficznych

Umiesz już posługiwać się edytorami grafiki. Retuszujesz, montujesz i zapisujesz zdjęcia i grafikę w różnych formatach i wiesz, jakie są między nimi różnice. Zmieniasz wymiary i rozdzielczość grafiki. Wszystkie wymienione umiejętności mają duży wpływ na atrakcyjność twojej strony. Zauważ, że podczas konstruowania strony parametry – rozdzielczość, wymiar i format pliku graficznego – mają znaczenie.

Pamiętasz zapewne, że istnieją dwa rodzaje tworzenia i zapisu grafiki – grafika wektorowa i grafika rastrowa. Zdjęcia i większość elementów graficznych na stronach internetowych są zapisywane w plikach .jpg, .png lub .gif, a więc rastrowo. Oznacza to, że można ściśle określić ich rozdzielczość i rozmiar. Łatwo to sprawdzić (rys. 31.1.).



Rys. 31.1. Jedna z metod odczytania danych pliku graficznego

Z danych pliku dowiesz się, jaki jest jego rozmiar, czyli liczba pikseli w pionie i poziomie – to jest ważna informacja dla webmastera. Korzystanie z CSS (o tym w następnych rozdziałach) uniezależnia rozmiar, w jakim element jest wyświetlany na ekranie, od jego fizycznych wymiarów. Niemniej skalowanie na stronie jedynie dopasowuje wysokość i szerokość, np. zdjęcia, do projektu. Niestety, plik graficzny jest pobierany w takiej postaci, jaka została umieszczona na serwerze. Rozmiary grafiki mają również znaczenie, gdy ważne są proporcje, np. dla nagłówka, banera itp.

31.2. Miejsce na obraz, czyli jak umieścić zdjęcie na stronie

◂ znacznik <img>

W HTML do umieszczania ilustracji używa się samozamykającego znacznika `<img>`. Domyślasz się zapewne, że to nie wystarczy i należy użyć jeszcze jego atrybutów. Musisz także wskazać miejsce na stronie i źródło. W wielu przypadkach dobrze jest też podać rozmiary bloku, w którym dana grafika ma być wyświetlona.

Załóżmy, że chcemy wyświetlić na stronie zdjęcie z rysunku 31.1.

Po pierwsze, musimy je umieścić w jednym z folderów strony, by móc wskazać atrybutem jego położenie (rys. 31.2.).



Rys. 31.2. Plik ze zdjęciem w przykładowym folderze strony

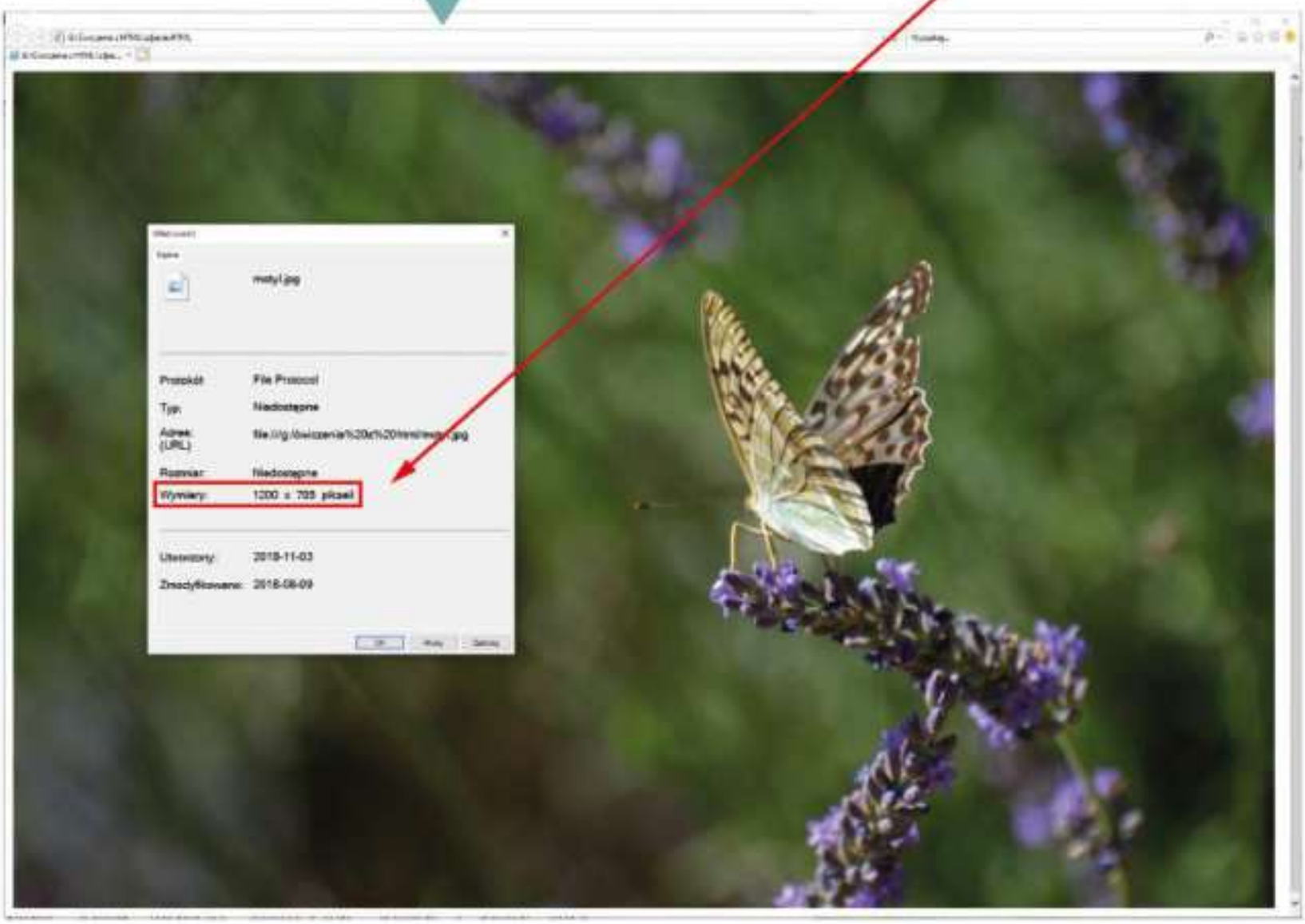
Teraz możemy użyć znacznika i atrybutu.

```
<img src="g:/ćwiczenia z html/motyl.jpg">
```

Jeśli zdjęcie (plik graficzny) znajdzie się w tym samym folderze co używający go HTML, nie musisz podawać pełnej ścieżki. Wystarczy jego nazwa. Ponieważ zdjęcie **motyl.jpg** ma wymiary 1200 x 795 pikseli (rys. 31.3.), w takiej wielkości zostanie wyświetlone na stronie. Nie jest to korzystne, zwłaszcza wtedy, gdy jest ono częścią artykułu. Domyślasz się dlaczego.

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
    </head>
    <body>
        <img src="g:/dwiczenia z html/motyl.jpg">
    </body>
</html>
```

Kliknij zdjęcie strony internetowej prawym przyciskiem i wybierz opcję **Właściwości**.



Rys. 31.3. Użycie znacznika **<img>** z atrybutem **scr=** na przykładowej stronie HTML

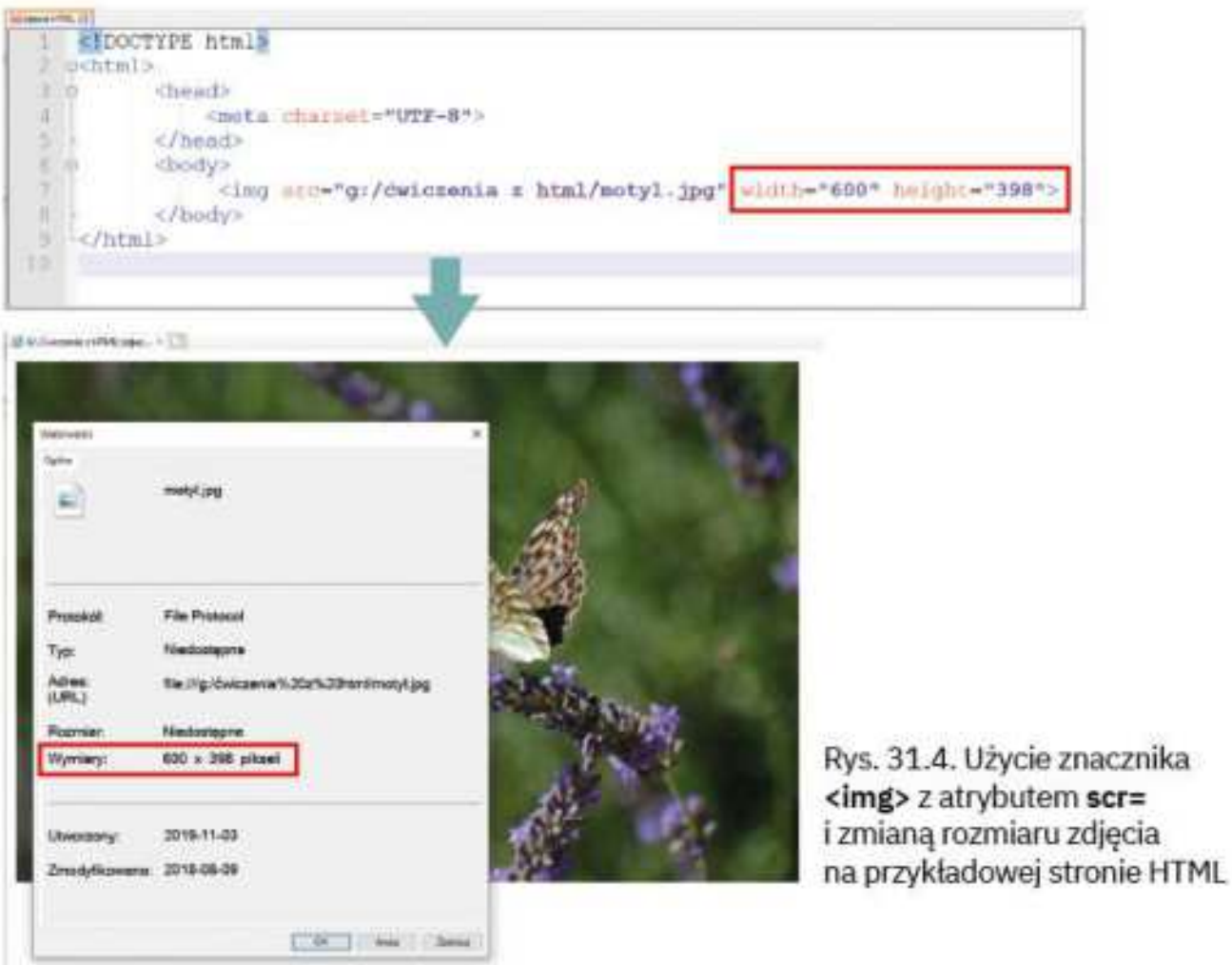
proporcje zdjęcia ▸ Dodajmy do znacznika `<scr>` kolejny atrybut. Tym razem zapanujemy nad wielkością wyświetlanego zdjęcia. **Proporcje** użytej fotografii to w przybliżeniu 1,5:1. Załóżmy, że chcemy wyświetlić to zdjęcie w polu o wymiarze poziomym 600 pikseli. Obliczmy, jaka powinna być jego wysokość.

$$\frac{1200}{795} = \frac{600}{y} \Rightarrow y = \frac{600 \times 795}{1200} \approx 398$$

Zmiany w kodzie HTML będą następujące:

```
<img src="g:/ćwiczenia z html/motyl.jpg" width="600" height="398">
```

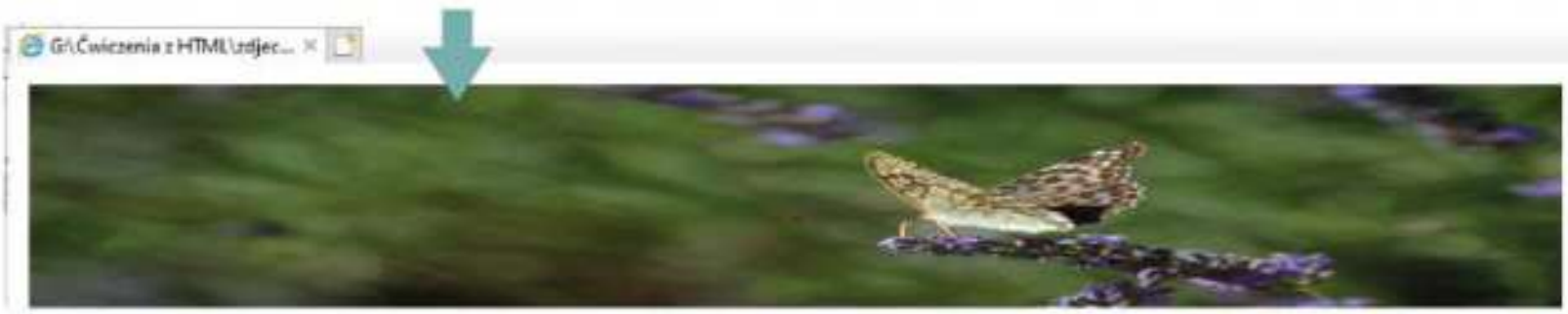
Sprawdźmy efekt działania atrybutu zmiany rozmiaru zdjęcia na stronie (rys. 31.4.).



Rys. 31.4. Użycie znacznika **<img>** z atrybutem **scr=** i zmianą rozmiaru zdjęcia na przykładowej stronie HTML

Jeśli w atrybutach `width=` i `height=` użyjesz danych niewynikających z proporcji zdjęcia, otrzymasz zniekształcony obraz na stronie (rys. 31.5.).

```
<img src="g:/ćwiczenia z html/motyl.jpg" width="600" height="100">
```



Rys. 31.5. Nieproporcjonalne wymiary zdjęcia podane w atrybutach znacznika **img**

Atrybuty `width=` i `height=` nie tylko zmieniają rozmiar zdjęcia, lecz także rezerwują dla niego miejsce na stronie. Łatwo to sprawdzić, usuwając plik graficzny z miejsca wskazanego w atrybucie `<scr>`. Nie wszystkie przeglądarki wyświetlają ramkę, co nie znaczy, że nie zostało zarezerwowane miejsce. Można je zobaczyć, zaznaczając obszar myszką (rys. 31.6.).



Rys. 31.6. Miejsce na zdjęcie zarezerwowane na stronie przez atrybuty **width=600** i **height=100** dla zdjęcia, którego przeglądarka nie znalazła we wskazanej lokalizacji

Jeśli w atrybutach wpiszesz wymiary większe niż oryginalne zdjęcie, otrzymasz obraz powiększony. Z racji użycia grafiki rastrowej straci on na jakości. Możesz także dodać atrybut `alt="tekst"`, który wyświetli tekst alternatywny, np. podający nazwę obiektu z ilustracji.

Spotkałeś/Spotkałaś się zapewne z przypadkiem, w którym po naprowadzeniu kursora na zdjęcie pojawiła się ramka z tekstem, np. z tematem zdjęcia. Ten efekt uzyskasz za pomocą atrybutu `title="napis"`.

```
<img src="g:/ćwiczenia z html/motyl.jpg" width="600" height="398"  title="motyl">
```

zdjęcie może być odnośnikiem

Rysunek, zdjęcie itp. mogą być, podobnie jak tekst, odsyłaczem. Wystarczy odpowiednio użyć znanego ci znacznika `<a> </a>`. Oczywiście możesz użyć także wszystkich atrybutów tego znacznika.

```
<a href="https://pl.wikipedia.org/wiki/Motyle">
<img src="g:/ćwiczenia z html/motyl.jpg" width="600" height="398"  title="motyl"></a>
```

Po naprowadzeniu kursora w obręb grafiki zmieni on swój wygląd w charakterystyczną dłoń. Kliknięcie spowoduje otwarcie danej strony.

Grafikę, np. nagłówki, można umieszczać w tabelach. Dzięki temu znajdą się tam, gdzie zaplanował projektant.

ZADANIA DO WYKONANIA

1. Sprawdź, jaki rozmiar mają zdjęcia twojego aparatu fotograficznego, np. w telefonie komórkowym. Oblicz proporcje, jakimi się charakteryzują. Zwróć uwagę, że różne programy tematyczne mogą tworzyć różnej wielkości zdjęcia. Pobierz jedno ze zdjęć na dysk komputera i sprawdź wielkość pliku. Zmniejsz proporcjonalnie wielkość zdjęcia, ustalając rozmiar poziomy na 400 pikseli. Ułóż tekst HTML, który kolejno wyświetli to zdjęcie w polach o podstawie 200, 400 i 600 pikseli. Porównaj te zdjęcia na stronie i opisz, jaki wpływ mają atrybuty znaczników na ich jakość na stronie.
2. Załóż, że masz zdjęcie o wymiarach 1600 x 900 pikseli. Podaj proporcje zdjęcia. Porównaj ze znanymi ci proporcjami ekranów komputerowych. Ile pikseli w pionie powinno mieć to zdjęcie, jeśli przekształcisz je do maksymalnego rozmiaru fotografii o proporcjach 4:3?
3. Ułóż HTML, który wyświetli trzy różne zdjęcia w trzech polach jednowierszowej tabeli. Postaraj się, by wszystkie były tej samej wielkości. Przedstaw, jak dobrać ich rozmiar, by mieściły się w polach tabeli.
4. Pobierz z sieci zdjęcie Fryderyka Chopina. Umieść je na stronie (HTML). Podpisz imieniem i nazwiskiem oraz datami urodzin i śmierci kompozytora. Utwórz z ilustracji odnośnik do strony Narodowego Instytutu Fryderyka Chopina w Warszawie.

PODSUMOWANIE LEKCJI

- Rozmiar zdjęcia sprawdzamy za pomocą opcji **Właściwości** z menu rozwiniętego po kliknięciu miniatury prawym klawiszem myszy. ◂ s. 200
- W HTML do wstawiania zdjęcia (grafiki) na stronę używa się **samozamykającego się znacznika** `<img>`. Należy pamiętać, by prawidłowo podać miejsce źródłowe grafiki wyświetlanej na stronie HTML. ◂ s. 201
- Zmniejszając zdjęcia na stronie, należy zachować ich proporcje. ◂ s. 202
- Zdjęcie może być **odnośnikiem** do miejsca na stronie, podstronie lub do innej strony w sieci. ◂ s. 204

32. Składnia stylów, czyli jak CSS pomaga w programowaniu strony

NA TEJ LEKCJI:

- poznasz rolę elementów w HTML jako selektorów CSS;
- utworzysz powiązanie kodu HTML z CSS;
- użyjesz CSS do zmiany atrybutów elementów strony.

Znasz już najważniejsze zasady tworzenia kodu HTML strony oraz podstawowe znaczniki. Poznałeś/Poznałaś także budowę dokumentu HTML. Wiesz także, jaką rolę przy powstawaniu strony odgrywa arkusz stylów CSS. Czas na wykorzystanie tej wiedzy i utworzenie pliku CSS formatującego dokument HTML. Użyjemy przykładu z poprzednich rozdziałów.

32.1. Identyfikujemy elementy, czyli elementy łączące CSS z HTML

Jeśli w HTML chcesz pewne elementy identyfikować i odróżniać od siebie dzięki nadaniu im oryginalnej, niewystępującej w całym dokumencie nazwy, musisz użyć **identyfikatora id**. Identyfikatory mają duże znaczenie w CSS. Między innymi dzięki nim można odnieść arkusze stylów do konkretnych elementów kodu HTML. Będą one tak zwanymi **selektorami CSS**, a nazwa identyfikatora w pliku CSS zostanie poprzedzona znakiem `#`.

◂ identyfikator

◂ selektor

klasy w HTML ▸ **Klasy** mogą być atrybutami wszystkich elementów w języku HTML. Pozwalają one na ich oznaczenie i mogą służyć jako **selektory** dla CSS. Słowem kluczowym dla klas jest `class=`.

W pliku CSS odwołanie do klasy to poprzedzona kropką jej nazwa, w odróżnieniu od `#` stosowanego dla `id`.

element blokowy <p> ▸ Do tej pory korzystaliśmy z kilku **elementów blokowych**, chociaż nie używaliśmy w stosunku do nich takiej nazwy. Były to: `<h1>` ... `<h6>`, `<ul>`, `<li>`, `<dl>`, `<dt>`, `<dd>`. W CSS ważny będzie także kolejny z nich – `<p> </p>`. Wyznacza on blok definiujący akapit, ale może także obejmować inne znaczniki i wykorzystywać atrybuty, na przykład `align=`.

sekcja ▸ W dokumencie HTML można tworzyć także **sekcje**. Oznaczamy je znacznikiem `<div> </div>`. Sekcje umożliwiają operowanie stylami CSS w pliku HTML. Każdej sekcji można nadać nazwę tworzącą selektor (id lub **klasę**). Dzięki nim CSS odwołuje się do konkretnej sekcji.

reguła CSS ▸ Poniżej zaprezentowano przykład, w którym CSS zmienia kolor tła na żółty dla tekstu wyróżnionego identyfikatorem `id=tekst`. Tekst CSS zawarty pomiędzy `{}` tworzy tak zwaną **regułę CSS**.

```
#tekst { background-color: yellow; }
```

gdzie `background-color` – jest **właściwością reguły**, a `yellow` jej **wartością**.

Identyfikator jest wykorzystywany także do odwołań do miejsc na stronie. W tym celu, podobnie jak do tworzenia linków zewnętrznych, używa się znacznika `<a> </a>`. Atrybutowi `href=` przypisujemy wtedy wartość równą nazwie odnośnika utworzonego przez `id`. Należy ją jednak poprzedzić znakiem `#`.

Przykład (po kliknięciu w **Nagłówek1** przeglądarka wyświetli stronę od miejsca oznaczonego identyfikatorem nagłówek1):

```
<a href="#nagłówek1">Nagłówek1</a>
```

Znasz już definicje najważniejszych elementów, które mają wpływ na połączenie CSS z HTML. W następnym podrozdziale poznasz ich działanie na przykładzie definiowania stylu dla znacznika.

32.2. Dwa języki tworzą stronę, czyli dodajemy CSS do HTML

Wiesz już, czym jest reguła CSS. Reguły (style) stosujemy do formatowania treści na stronie, podobnie jak opcje formatowania w edytorach tekstu. Przeglądarka musi jednak wiedzieć, jaki styl oraz do których fragmentów strony ma zastosować. Obok pliku HTML należy więc umieścić na serwerze plik z rozszerzeniem nazwy .css (dla ułatwienia zapisujemy go w tym samym folderze). Ponieważ przeglądarka interpretuje plik HTML, to w nim należy dodać odpowiednią informację wskazującą, który z plików CSS ma być odpowiedzialny za formatowanie danej strony.

Załóżmy, że plik stylów nazwiemy **styleegzaminy.css**. Plikowi możesz nadać dowolną nazwę (bez spacji), jednak postaraj się w pewien sposób odnieść ją do nazwy pliku HTML, który ma formatować. W przykładzie odniesiemy style do pliku egzaminy.html (rys. 32.1.).

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
    </head>

    <body>
      <dl>
          <h3>
          <dt class="naglowek">Odwiedź przed egzaminami</dt>
          </h3>
          <dd><a href="https://www.men.gov.pl" target="_blank">Ministerstwo Edukacji Narodowej</a></dd>
          <dd><a href="https://cke.gov.pl" target="_blank">Centralna Komisja Egzaminacyjna</a></dd>
          <h3>
          <dt class="naglowek">Wyniki egzaminów</dt>
          </h3>
          <dd><a href="https://cke.gov.pl/wyniki-egzaminow/egzamin-maturalny/" target="_blank">Wyniki egzaminów maturalnych</a></dd>
          <dd><a href="https://cke.gov.pl/wyniki-egzaminow/egzamin-zawodowy-2/" target="_blank">Wyniki egzaminów zawodowych</a></dd>
      </dl>
    </body>
</html>
```



Odwiedź przed egzaminami

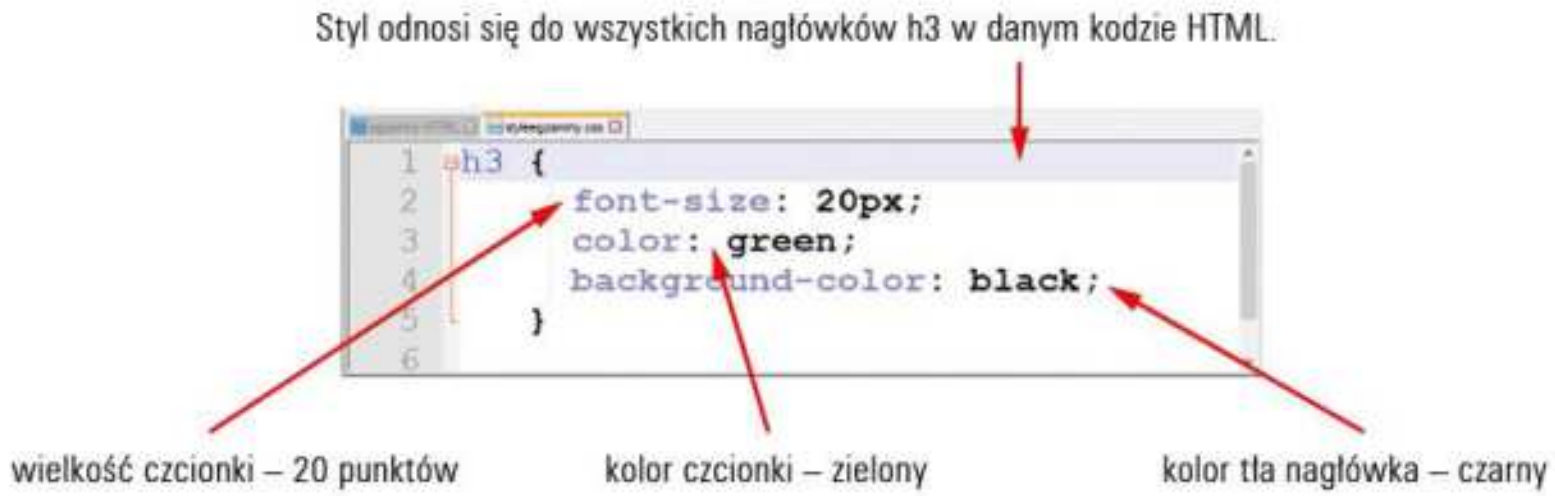
Ministerstwo Edukacji Narodowej
Centralna Komisja Egzaminacyjna

Wyniki egzaminów

Wyniki egzaminów maturalnych
Wyniki egzaminów zawodowych

Rys. 32.1. Przykładowa strona HTML przed dodaniem pliku stylów CSS

Nagłówki `h3` w naszym kodzie HTML są formatowane parametrami domyślnymi (rys. 32.1.). Zmieńmy kolor czcionki w nagłówkach `h3` na zielony i nadajmy tekstowi czarne tło. W tym celu utwórzmy styl (rys. 32.2.). Jeśli użyjesz odpowiedniego edytora, np. Notepad++, i zapiszesz plik pod nazwą z rozszerzeniem **.css**, uzyskasz ułatwiające edycję kolorowanie tekstu i podpowiedzi w trakcie jego pisania. Tego samego edytora użyto do tworzenia naszego przykładu w HTML.



Rys. 32.2. Treść przykładowego pliku stylów CSS formatującego nagłówek h3

Skoro mamy już plik CSS, musimy zmodyfikować program strony (HTML) w taki sposób, by style mogły formatować tę stronę. Dodajmy do programu kod podłączający do niej plik stylów (rys. 32.3.).

```
1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <meta charset="UTF-8">
6          <link rel="stylesheet" href="styleegzaminy.css" type="text/css">
7      </head>
```

Rys. 32.3. Fragment kodu HTML podłączający plik stylów styleegzaminy.css

Pozostaje już tylko się upewnić, czy oba pliki są zapisane w tym samym folderze, i uruchomić plik HTML w przeglądarce (rys. 32.4.).

Odwiedź przed egzaminami

Ministerstwo Edukacji Narodowej
Centralna Komisja Egzaminacyjna

Wyniki egzaminów

Wyniki egzaminów maturalnych
Wyniki egzaminów zawodowych

Rys. 32.4. Efekt działania przykładowego stylu CSS na stronę HTML

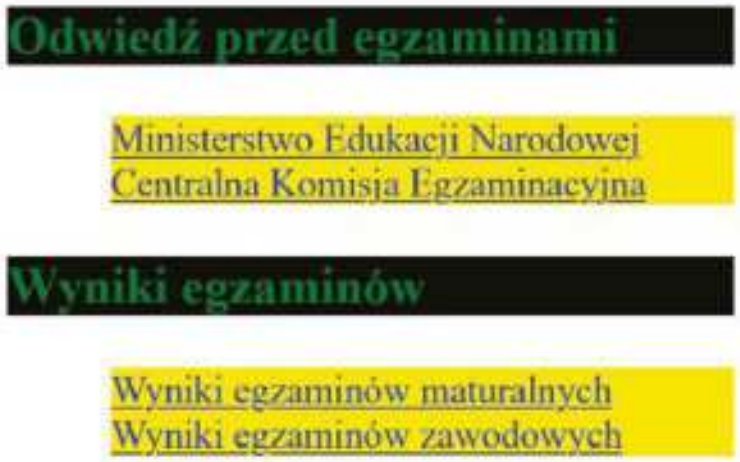
W naszym przykładzie regułę CSS odnieśliśmy do nagłówka h3. Zauważ, że w związku z tym **wszystkie** nagłówki h3 zostały sformatowane według danego stylu. Jeśli nagłówki h3 wystąpiłyby jeszcze wielokrotnie, ich tekst także zostałby wyświetlony zgodnie z regułą.

Jeśli formatowaniem w naszym przykładzie chcielibyśmy objąć wszystkie bloki <dd> (teksty odnośników) i nadać im żółty kolor tła i wielkość liter 15 punktów, tekst naszego CSS miałby postać jak na rysunku 32.5.

```
1  h3 {
2        font-size: 20px;
3        color: green;
4        background-color: black;
5     }
6  dd {
7      font-size: 15px;
8      background-color: yellow;
9     }
```

Rys. 32.5. Przykład dodania do strony stylu dla bloku <dd>

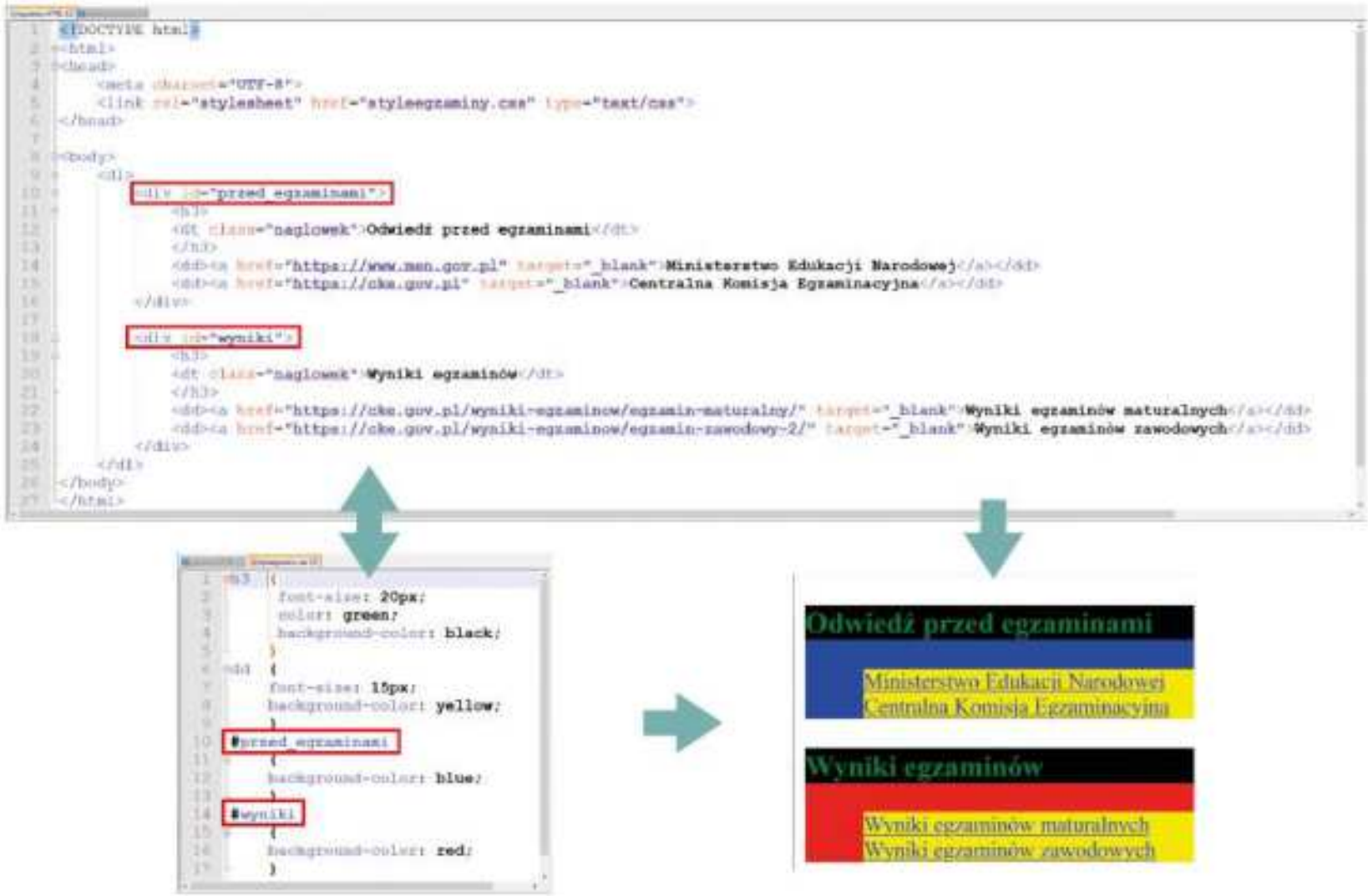
Ponieważ nie zmienialiśmy nazwy pliku CSS, nie zmieniamy nic w kodzie dołączającym style do strony. Efekt działania stylów przedstawiono na rysunku 32.6. Tu także wyraźnie widać, że reguła zapisana w pliku CSS dla bloków <dd> została zastosowana do wszystkich czterech występujących w programie HTML.



Rys. 32.6. Efekt działania stylów CSS dla nagłówków h3 i bloków <dd>

Do formatowania strony za pomocą CSS możesz oczywiście zastosować identyfikatory i **sekcje**. Dzięki temu nagłówki `h3` z naszego przykładu mogłyby być różnie formatowane.

Załóżmy, że chcesz nadać inny kolor tła dla listy z odnośnikami *Odwiedź przed egzaminami* i inny dla *Wyniki egzaminów*. Użycie stylu dla `<dl>` spowodowałoby nadanie takiego samego tła dla obu. Utwórzmy w naszym programie HTML sekcje za pomocą znacznika `<div>` i wprowadźmy atrybut `id` (identyfikator) jako selektor dla CSS (rys. 32.7.).



Rys. 32.7. Przykład zastosowania sekcji i selektora do formatowania przez CSS

Używając `id=`, należy pamiętać, że w całym pliku HTML dany selektor (identyfikator) może wystąpić tyko raz. W naszym pliku ta zasada została zachowana. Nazwy selektorów powinny być tak dobrane, by łatwo było je zidentyfikować w obu plikach.

ZADANIA DO WYKONANIA

1. Korzystając z podręcznika i wiarygodnych publikacji, opracuj rysunek (planszę) przedstawiający różnice pomiędzy elementem blokowym, identyfikatorem, selektorem i klasą w HTML.
2. Dokonaj modyfikacji kodów HTML i CSS z rozdziału i ustal nowe style dla list oraz nagłówków. Samodzielnie dobierz kolory i wielkości liter. Jako selektorów użyj klas.
3. Dokonaj modyfikacji kodów HTML i CSS z rozdziału i zmień krój liter dla nagłówków. Potrzebne do tego informacje znajdź w wiarygodnych źródłach. Przedstaw sposób, którym się posłużyłeś/posłużyłaś. Zastosuj selektory identyfikatorów id.
4. Odnajdź w podręczniku przyklad budowania tabeli. Zmodyfikuj kod HTML i utwórz plik CSS, by zmienić kolory kolumn i wielkość czcionki dla nagłówków tabeli.

5*. Zmodyfikuj plik CSS i HTML z przykładu w rozdziale w taki sposób, by tlem dla strony było zdjęcie. Znajdź w wiarygodnych źródlach sposoby takiego formatowania tła i skalowania zdjęcia.

PODSUMOWANIE LEKCJI

s. 205 • **Elementy blokowe, sekcje, identyfikatory i klasy** mogą być wykorzystane do formatowania strony HTML za pomocą stylów CSS.

s. 205 • Dzięki **selektorom** możliwe jest odniesienie formatowania stylami z pliku CSS do konkretnych elementów strony HTML.

s. 206 • **Rozszerzeniem** nazwy pliku ze stylami CSS dla kodu HTML powinno być .css.

s. 206 • W kodzie HTML w nagłówku `<head></head>` **należy umieścić odniesienie do konkretnego pliku CSS** formatującego stylami daną stronę, np.:

```
<link rel="stylesheet" href="nazwa_pliku_css.css" type="text/css">
```

s. 206 • **Nie można nadawać takich samych nazw** identyfikatorom id w tym samym kodzie HTML.

s. 206 • W pliku CSS odwołanie do części objętej selektorem `id` dokonuje się za pomocą **#**. W przypadku użycia **klasy** używa się **kropki**. Dla elementów zawartych **pomiędzy znacznikami** używa się ich nazwy bez nawiasów, np. **p**, **dd**.

33. Pliki stylów, czyli CSS w akcji

NA TEJ LEKCJI:

- utworzysz menu strony składające się z przycisków za pomocą CSS;
- nauczysz się zmieniać atrybuty dla selektorów;
- zastosujesz pseudoklasę hover w celu uwidocznienia relacji pomiędzy kursorem a przyciskiem.

Pierwsze kroki w użyciu CSS już za tobą. Zmiany kolorów to jednak jedynie niewielki zakres zastosowań stylów. Przypomnij sobie sposoby tworzenia menu z odsyłaczami. Umiesz je zaprogramować jako aktywny tekst odsyłający po kliknięciu do danego miejsca na stronie lub sieci. Utwórzmy menu z przyciskami. Użyjmy w tym celu pliku stylów CSS.

33.1. Aktywne klawisze, czyli tworzymy menu strony

Przypomnij sobie, jak w kodzie HTML tworzyliśmy listę definicji, a następnie pewnym fragmentom nadawaliśmy funkcję odsyłacza. Odsyłacze w postaci tekstu bardzo dobrze się sprawdzają wewnątrz artykułów umieszczanych na stronie. Ale co zrobić, jeśli chcemy, by odsyłacze stały się elementami menu strony? Oczywiście można to zapisać w kodzie HTML, ale wygodniej jest utworzyć odpowiedni program CSS.

Załóżmy, że chcemy utworzyć menu z pięcioma przyciskami ułożonymi pionowo. Oczywiście pracę rozpoczniemy od ułożenia kodu HTML. Skupimy się jedynie na menu, nie będziemy zajmować się pozostałą treścią strony (rys. 33.1.). Kod programu powinien być dla ciebie czytelny.

Uwaga!

Dla zachowania przejrzystości kodu w przykładzie nie użyto konkretnych adresów odsyłaczy. Możesz je wpisać w swoim projekcie.

```
<html>

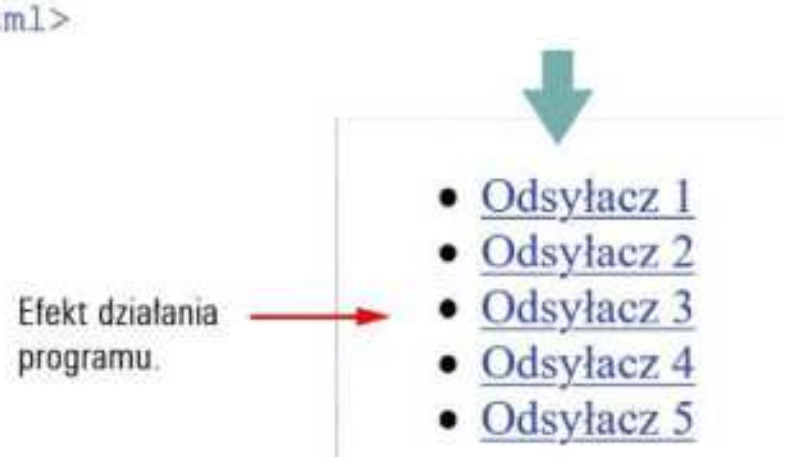
<head>
    <meta charset="utf-8">
    <title>Menu bez przycisków</title>
</head>

<body>

        <nav>
            <ul>
                <li><a href="link 1">Odsyłacz 1</a></li>
                <li><a href="link 2">Odsyłacz 2</a></li>
                <li><a href="link 3">Odsyłacz 3</a></li>
                <li><a href="link 4">Odsyłacz 4</a></li>
                <li><a href="link 5">Odsyłacz 5</a></li>
            </ul>
        </nav>

</body>

</html>
```



Rys. 33.1. Przykładowy program HTML tworzący na stronie odsyłacze

Utwórzmy teraz plik z kodem CSS (rys. 33.2.) i zapiszmy go pod nazwą **menu_pionowe.css** w tym samym folderze co plik **menu.html**. Przeczytaj uważnie opis znaczenia poszczególnych linii umieszczony w tabeli (tab. 33.1.).



Rys. 33.2. Przykładowa lista stylów CSS tworząca pionowe menu przycisków

W naszym CSS selektor `ul` oznacza, że formatowanie pomiędzy nawiasami { } opisanym niżej stylem będzie dotyczyło listy nieuporządkowanej z programu HTML. Oczywiście jeśli użyjemy na stronie więcej znaczników `<ul>`, to formatowanie będzie aktywne dla wszystkich. Jeśli ma dotyczyć tylko jednej listy, należy użyć innego selektora, np. `id`.

Atrybuty selektora	Opis
`float: left;`	Pozycjonuje element: left – po lewej stronie względem elementów sąsiadujących, right – po prawej stronie względem elementów sąsiadujących, none – element nie sąsiaduje z żadnym elementem.
`list-style: none;`	Określa sposób oznaczania pozycji listy. W tym przypadku nie będą one oznaczane, np. kropką.
`display: block;`	Element wskazany selektorem będzie wyświetlony w bloku.
`width: 100px;`	Szerokość bloku zostaje ustalona na 100 punktów i nie będzie stała.
`font-size: 1em;`	Wielkość czcionki zostaje ustalona na taką samą, jaka została ustalona w kodzie HTML.
`text-align: center;`	Tekst będzie ustawiony w bloku centralnie.
`text-decoration: none;`	Atrybut none oznacza, że nie będzie widoczna linia podkreślająca odsyłacz.
`color: white;`	Ustalenie koloru liter – biały.
`background-color: gray;`	Ustalenie koloru tła bloku – szary.

Atrybuty selektora	Opis
`padding: 5px 10px;`	Ustalenie marginesów wewnętrznych. W związku z ustaleniem szerokości bloku na stałą wartość, niemieszczące się w linii napisy będą przesuwane do nowej linii. Marginesy będą zachowane.
`margin: 0px 0px 4px 0px;`	Ustalenie marginesów zewnętrznych. Od nich zależy odległość, w jakiej mogą się pojawić inne obiekty, np. następny przycisk.
`border-radius: 4px;`	Ustalenie promienia zaokrąglenia narożników bloku.
`box-shadow: 2px 2px 2px silver;`	Ustalenie wymiarów, głębokości i koloru cienia bloku.

Tab. 33.1. Znaczenie atrybutów selektorów dla przykładowego pliku CSS

Uwaga!

Opisane w tabeli elementy stylów mogą mieć wiele innych atrybutów. Ich opisy znajdziesz w wiarygodnych źródłach.

Po uruchomieniu naszej strony przyciski staną się odnośnikami. Będzie o tym świadczyć charakterystyczna zmiana wyglądu kursora. Jeśli w pliku HTML wpiszesz istniejące adresy, np. innych stron, miejsc na danej stronie lub podstron, kliknięcie przycisku do nich przeniesie.

33.2. Element reaguje na kursor, czyli dodajemy efekty w menu

Nasze menu z przyciskami jest w zasadzie gotowe. Można jednak je uatrakcyjnić przez dodanie efektu pojawiającego się np. po naprowadzeniu kursora na przycisk. Kod HTML pozostawimy bez zmian. Zajmijmy się zmianami w programie CSS (rys. 33.3.).

```
ul
    {
    float: left;
    list-style: none;
    }

li a
    {
    display: block;
    width: 100px;
    font-size: 1em;
    text-align: center;
    text-decoration: none;
    color: white;
    background-color: gray;
    padding: 5px 10px;
    margin: 0px 0px 4px 0px;
    border-radius: 4px;
    box-shadow: 2px 2px 2px silver;
    }

li a:hover
    {
    background: silver;
    }
```

Rys. 33.3. Przykładowy CSS powodujący reakcję przycisku na kursor

pseudoklasa ▸ **Pseudoklasy** to w zasadzie identyfikatory, które przeglądarka stosuje bez konieczności określania danych elementów za pomocą stylów.

pseudoklasa hover ▸ Pseudoklasa `hover` przypisana w CSS do atrybutu modyfikuje zachowanie się obiektu wchodzącego w interakcję, np. z kursorem. Nie oznacza to aktywowania obiektu. W naszym przykładzie kolor przycisku zmieni się po naprowadzeniu na niego kursora myszy (rys. 33.4.). W CSS można użyć jeszcze kilku innych pseudoklas. Poczytaj o nich w wiarygodnych źródłach.



Rys. 33.4. Efekt działania kodu CMS używającego pseudoklasy hover

Za pomocą pseudoklasy `hover` zaprogramujesz także inne zachowanie się obiektu, np. zanik cienia, co da złudzenie obniżenia się przycisku. Oczywiście oba efekty można zastosować jednocześnie (rys. 33.5.).

```
24  li a:hover
25  {
26  background: silver;
27  box-shadow: 0px 0px 0px;
28  }
```



Rys. 33.5. Inny przykład użycia w CSS pseudoklasy hover

Na stronach często spotyka się także menu ułożone poziomo. W naszym przykładzie do zmiany ułożenia przycisków wystarczy mała zmiana w pliku CSS – znajdź poprawkę w kodzie na rysunku 33.6.

```css
ul
    {
    list-style: none;
    }

li a
    {
    display: block;
    width: 100px;
    font-size: 1em;
    text-align: center;
    float: left;
    text-decoration: none;
    color: white;
    background-color: gray;
    padding: 5px 10px;
    margin: 0px 4px 4px 0px;
    border-radius: 4px;
    box-shadow: 2px 2px 2px silver;
    }

li a:hover
    {
    background: silver;
    box-shadow: 0px 0px 0px;
    }
```



Rys. 33.6. Korekta w przykładowym CSS zmieniająca układ przycisków menu na poziomy

33.3. Menu na swoim miejscu, czyli układ elementów strony

Menu gotowe. Oczywiście można je zaprogramować jeszcze na kilka innych sposobów. Nie zmieniajmy jednak podejścia i użyjmy menu z przykładu do budowy prostej strony. Załóżmy, że menu ma się znaleźć po lewej stronie, po prawej zarezerwujmy miejsce na wiadomości i artykuły. Na górze strony utwórzmy miejsce na nagłówek, a na dole – na stopkę.

W **nagłówku** znajdującym się na górze strony najczęściej umieszcza się tytuł, logo, nazwę itp.

◂ nagłówek

Stopkę umieszcza się na dole strony i zamieszcza w niej informacje o autorach i właścicielach strony, rok wykonania, kontakty itp.

◂ stopka

Zmodyfikujmy nieco kod HTML, a następnie CSS. Użyjmy identyfikatorów `id`, by stworzyć **selektory** dla CSS. Chcielibyśmy, aby poszczególne bloki strony

zmieniały swoją wielkość w zależności od rozdzielczości ekranu. Miejsce na wiadomości nie powinno być ograniczone od dołu. Powinno się wydłużać wraz z wydłużaniem się tekstu. Wysokość stopki i nagłówka powinna zostać ustalona na stałe.

```html
<!DOCTYPE html>

<html>

<head>
    <meta charset="utf-8">
    <title>szablon strony</title>
    <link rel="stylesheet" href="formatowanie strony.css" type="text/css">
</head>

<body>

    <div id="naglowek">
        Miejsce na nagłówek np. graficzny
    </div>

    <div id="menu">
        <nav>

            <li><a href="link 1">Odsyłacz 1</a></li>
            <li><a href="link 2">Odsyłacz 2</a></li>
            <li><a href="link 3">Odsyłacz 3</a></li>
            <li><a href="link 4">Odsyłacz 4</a></li>
            <li><a href="link 5">Odsyłacz 5</a></li>

        </nav>
    </div>

    <div id="informacje">
        Tu będą wstawiane artykuły<br>
        z życia szkoły i zdjęcia<br>
        naszych reporterów<br>
    </div>

    <div id="stopka">
        Stopka informująca o właścicielu, autorach strony itp.
    </div>

</body>

</html>
```

Dodanie identyfikatorów id do bloków elementów blokowych pozwoli na użycie selektorów w CSS.

```css
li a
    {
    display: block;
    font-size: 1em;
    text-align: center;
    text-decoration: none;
    color: white;
    background-color: gray;
    padding: 5px 10px;
    margin: 5px 0px 4px 0px;
    border-radius: 4px;
    box-shadow: 3px 3px 3px silver;
    }

li a:hover
    {
    background: silver;
    box-shadow: 0px 0px 0px;
    }

#naglowek
    {
    background: gray;
    text-align:center;
    min-height:100px;
    }

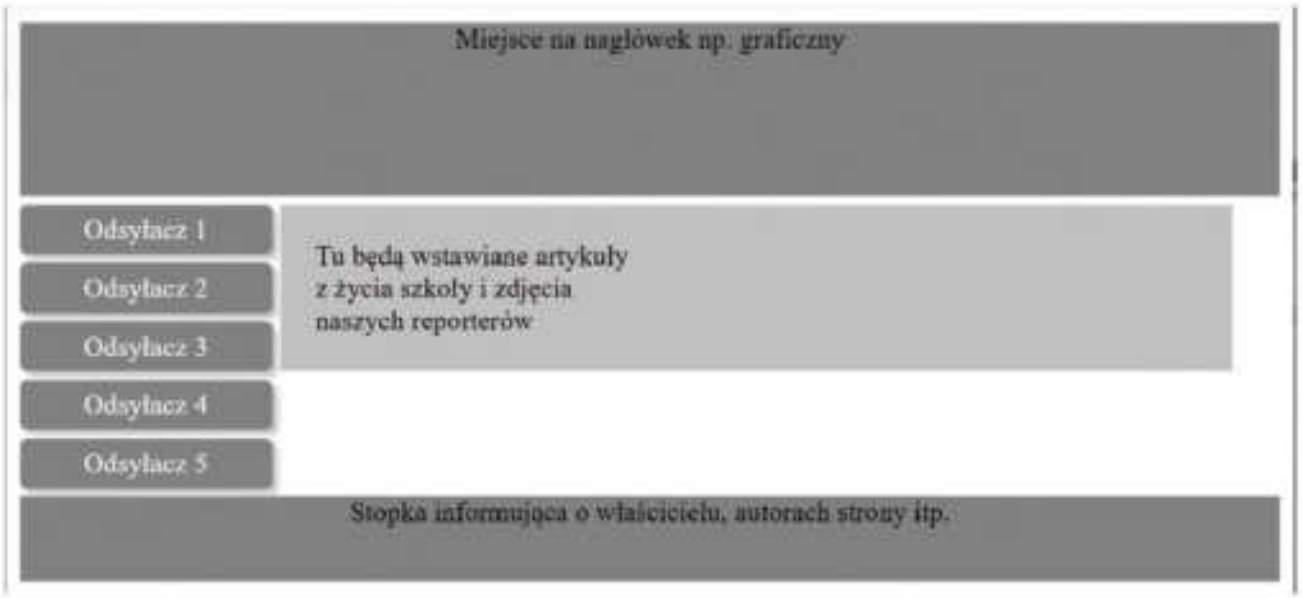
#menu
    {
    float: left;
    width: 20%;
    }

#informacje
    {
    float:left;
    width: 70%;
    background: silver;
    padding:20px;
    margin: 5px 5px 5px 5px;
    min-height:100%;
    }

#stopka
    {
    clear:both;
    background: gray;
    text-align: center;
    min-height:50px;
    }
```

Główny blok strony będzie się wydłużał wraz z wielkością obiektów i zajmie 70% szerokości okna przeglądarki.

Rys. 33.7. Kod HTML i CSS przykładowego rozmieszczenia elementów strony



Rys. 33.8. Efekt działania przykładowego programu HTML i CSS

Jeśli chcesz poznać wszystkie tajniki HTML i CSS, sięgnij do książek i wiarygodnych stron internetowych. Znajdziesz tam wiele kursów przeznaczonych dla użytkowników o różnym stopniu zaawansowania. Nic jednak nie zastąpi eksperymentowania i ćwiczeń. Konsultuj swoje prace z innymi programistami i korzystaj z ich uwag.

Wikipedia podaje pełną definicję walidacji w odniesieniu do informatyki:

◂ walidacja programu CSS

„**Walidacja** – w naukach technicznych i informatyce działanie mające na celu potwierdzenie w sposób udokumentowany i zgodny z założeniami, że procedury, procesy, urządzenia, materiały, czynności i systemy rzeczywiście prowadzą do zaplanowanych wyników".

Jak sprawdzić poprawność kodu CSS, czyli jak przeprowadzić walidację? Istnieją w sieci specjalne usługi, tak zwane **Validation Service** (czyt. walidejszyn serwis). Przykładem może być strona *http://jigsaw.w3.org/css-validator*, która sprawdzi twój CSS (rys. 33.9.).



Rys. 33.9. Przykładowa usługa walidacyjna sprawdzająca poprawność kodu CSS

Jeśli chcesz zostać programistą/programistką stron, przed tobą jeszcze długa droga. Nie ustawaj w nauce i pracuj nad doskonaleniem swojego warsztatu.

ZADANIA DO WYKONANIA

1. Przeanalizuj kod z rysunku 33.2. i zmień kolor oraz czcionki przycisków. W odpowiednim miejscu kodu HTML wstaw odnośniki do stron Ministerstwa Edukacji Narodowej, Centralnej Komisji Egzaminacyjnej, Okręgowej Komisji Egzaminacyjnej i wyników matur. Zmień odpowiednio opisy przycisków. Dokonaj walidacji pliku CSS.
2. W przykładzie z rysunku 33.2. zmień kolory przycisków tak, by każdy miał inną barwę. Omów zmiany konieczne do wykonania zadania. Dokonaj walidacji pliku CSS.
3. Zmodyfikuj plik CSS z przykładu w taki sposób, by interakcja kursora z przyciskiem była sygnalizowana nieznacznym powiększeniem przycisku. Zwróć uwagę na to, że także cień powinien się zmienić. Dokonaj walidacji pliku CSS.
4. Znajdź w sieci strony, na których można dokonywać walidacji plików HTML lub CSS. Oceń ich wiarygodność poprzez sprawdzenie właściciela usługi lub miejsca, z którego pochodzi usługa.

5*. Umieść przyciski menu w tabeli w taki sposób, by znajdowały się zawsze po lewej stronie strony. Tabela powinna mieć trzy pola: dla nagłówka, dla menu (po lewej) i dla treści, czyli głównej części strony.

PODSUMOWANIE LEKCJI

s. 211 ▸ • **Listę z odsyłaczami** można zmienić w **aktywne przyciski** za pomocą formatowania w CSS.

s. 214 ▸ • **Pseudoklasa** przypomina identyfikator, jednak przeglądarka stosuje go bez konieczności określania danych elementów za pomocą stylów.

s. 214 ▸ • Pseudoklasa `hover` przypisana w CSS do atrybutu modyfikuje zachowanie się obiektu wchodzącego w interakcję, np. z kursorem.

s. 217 ▸ • **Walidacja** to proces sprawdzania poprawności kodu.

34. Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej

NA TEJ LEKCJI:

- dowiesz się, czym są dynamiczne strony WWW;
- umieścisz na stronie działającą aplikację udostępnioną przez usługę sieciową;
- umieścisz na stronie widżet z prognozą pogody.

Zakupy w sklepie internetowym zapewne nie są ci obce. Przed wyborem towaru wpisywałeś/wpisywałaś do wyszukiwarki sklepu cechy towaru, które cię interesują. Jak to się dzieje, że wyświetliły ci się propozycje zgodne z oczekiwaniami? Skąd na stronach biorą się widżety? Czym są dynamiczne strony internetowe? Tego wszystkiego dowiesz się z tego rozdziału.

34.1. Strona czy aplikacja internetowa, czyli czym różni się strona statyczna od dynamicznej

Intuicyjnie rozumiesz różnicę pomiędzy pojęciami „dynamiczne" i „statyczne". Pierwsze opisuje zjawiska, publikacje itp., które zmieniają się pod wpływem różnych okoliczności, a drugie – które są niezmienne i pozostają takie, jakie zostały utworzone. Jak to można odnieść do stron internetowych? Ze statycznymi i dynamicznymi stronami spotykasz się na co dzień. Przykładem dynamicznej publikacji internetowej może być strona prezentująca aktualny stan powietrza w danej okolicy czy notowania giełdowe. Te dane zmieniają się dynamicznie, więc i treści na stronie muszą tak się zmieniać. Stroną statyczną jest każda strona, która zmienia się jedynie po wprowadzeniu modyfikacji przez jej autora. Wszystkie strony z naszych ćwiczeń są oczywiście statyczne. Zdefiniujmy zatem strony dynamiczne.

◂ strony dynamiczne

Stroną dynamiczną nazywamy taką publikację internetową, w której część lub wszystkie treści nie mają określonej, stałej zawartości (część kodu nie została określona) i mogą się zmieniać w zależności od decyzji użytkownika, zawartości określonej bazy danych itp. (rys. 34.1.). Najczęściej jest ona w całości lub w części aplikacją internetową.

Proces wyświetlania strony statycznej





Rys. 34.1. Różnica pomiędzy działaniem stron statycznych i dynamicznych

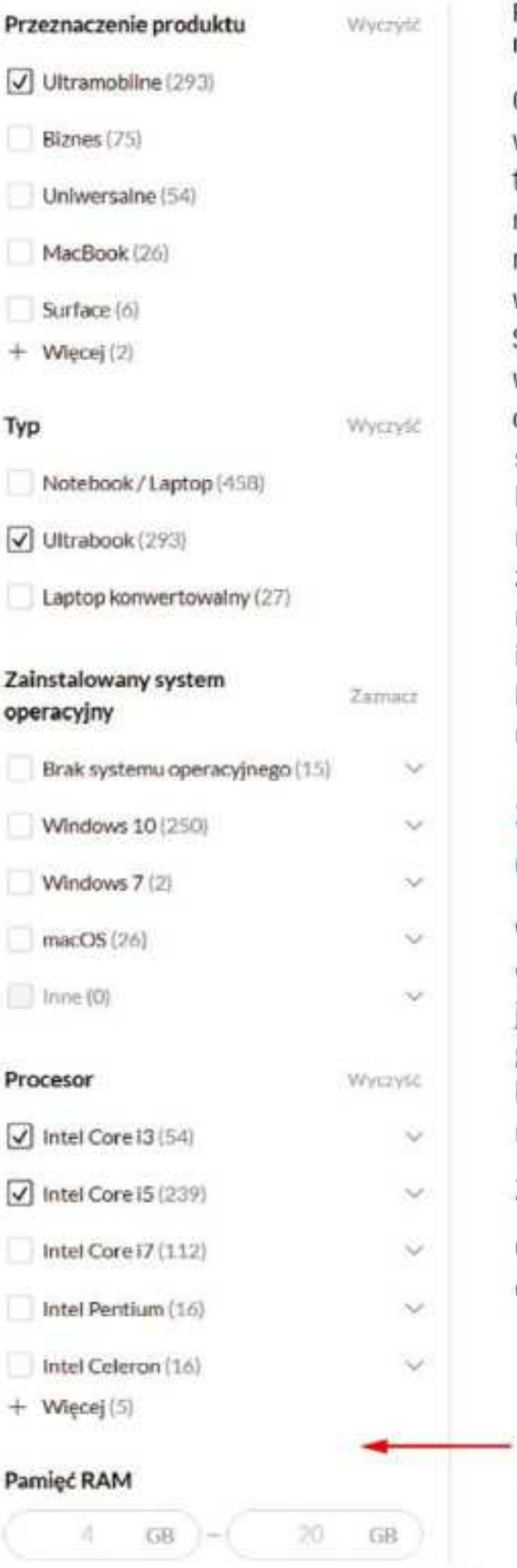
Część stron dynamicznych korzysta z baz danych. Jest to wygodne, ponieważ wystarczy przygotować jedynie wygląd takiej strony, a wypełnienie jej informacjami, treściami itp. można zostawić programowi uruchomionemu w serwerze. Dzięki temu programista strony nie musi zmieniać jej wraz ze zmianami danych, np. na stronie stacji pogodowej. Stacja zapisuje dane w bazie na serwerze. W ten sposób wiele stron może korzystać z tych samych danych i będą one zawsze aktualne, bez konieczności ingerowania w kod strony. Dynamiczne WWW pozwalają także na udział użytkownika z wyborze informacji (danych) prezentowanych na stronie, np. z zastosowaniem odpowiednich formularzy z zapytaniami lub parametrami wyszukiwania. Klasycznymi przykładami takiego rozwiązania są strony sklepów internetowych. Klient wysyła do bazy kryteria, według których mają być wyszukiwane towary, a oprogramowanie uwzględni je w procesie wyszukiwania (rys. 34.2.).

34.2. Krzyżówka, czyli dodajemy element dynamiczny do strony

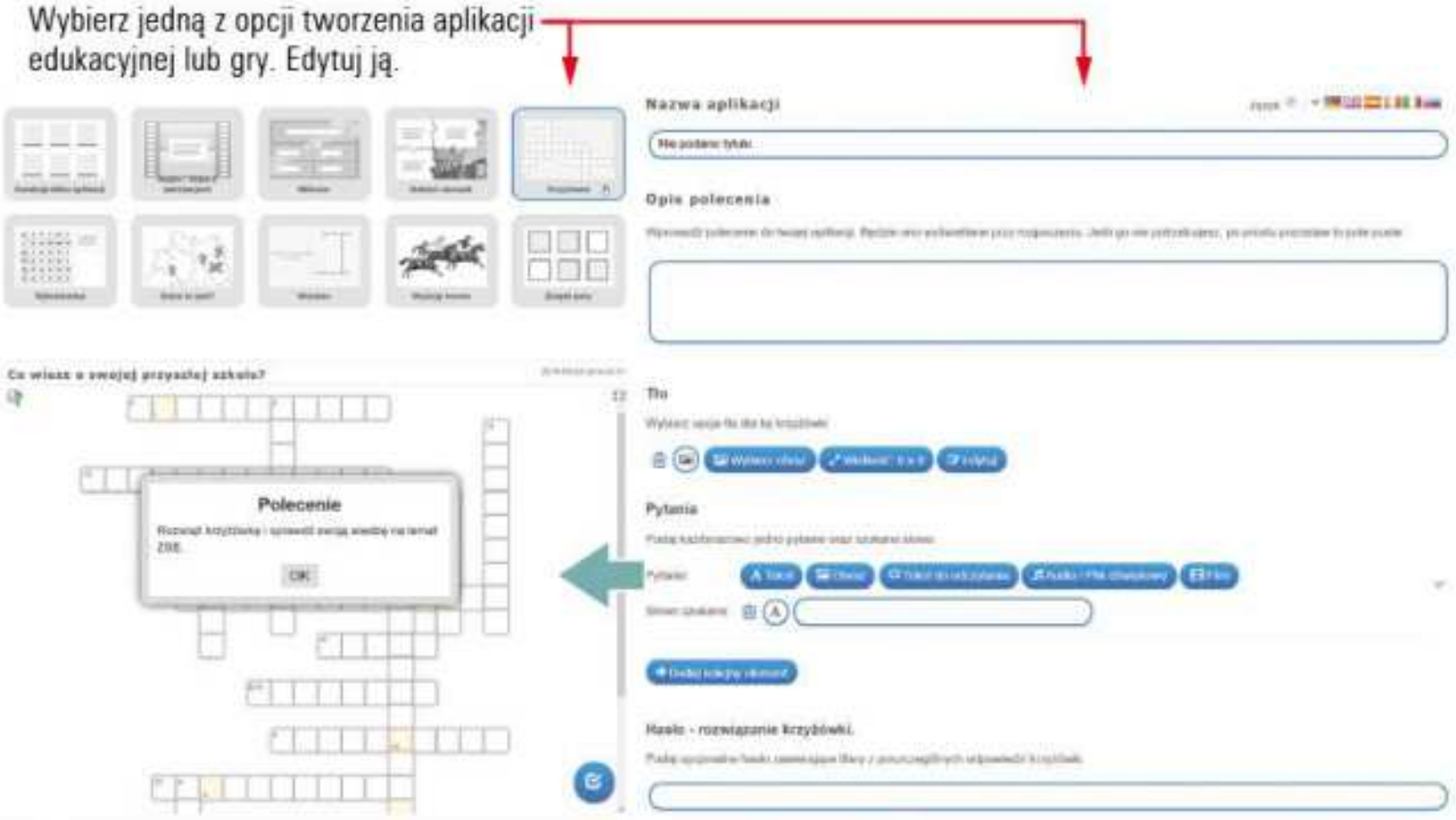
W naszym przykładzie dołączymy krzyżówkę utworzoną w darmowym serwisie *learningapps.org*. Jej opracowanie jest bardzo łatwe i z pewnością sobie z tym poradzisz. Oprogramowanie tej strony pozwala udostępniać utworzone aplikacje – łamigłówki, krzyżówki itp. – na różne sposoby. Jednym z nich jest wyświetlanie na własnej stronie WWW.

Zacznijmy od utworzenia krzyżówki (rys. 34.3.).

Oczywiście stronę *learningapps.org* także zaliczamy do dynamicznych.

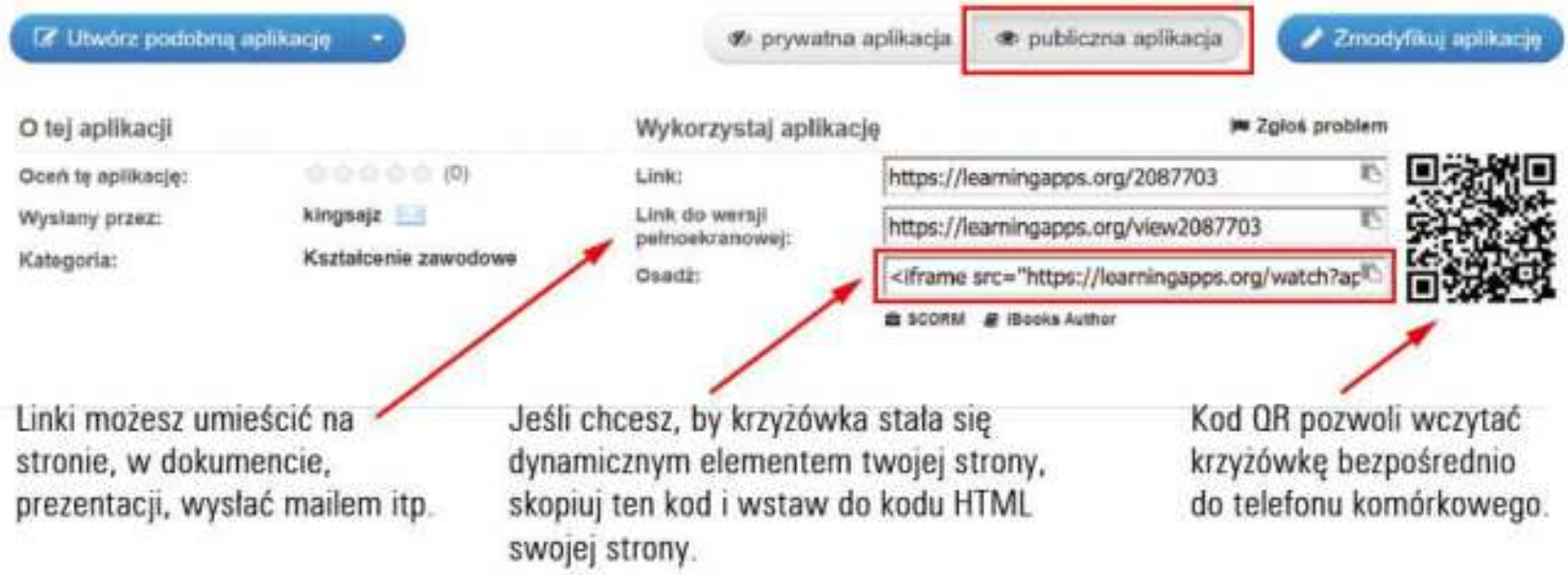


Rys. 34.2. Przykładowe kryteria wyszukiwania ofert laptopów w sklepie internetowym



Rys. 34.3. Przykład tworzenia aplikacji edukacyjnej w usłudze *learningapps.org*

Gdy twoja krzyżówka lub inna forma łamigłówki jest już gotowa, możesz wybrać sposób jej prezentacji (rys. 34.4.).



Rys. 34.4. Opcje udostępniania aplikacji utworzonych w *learningapps.org*

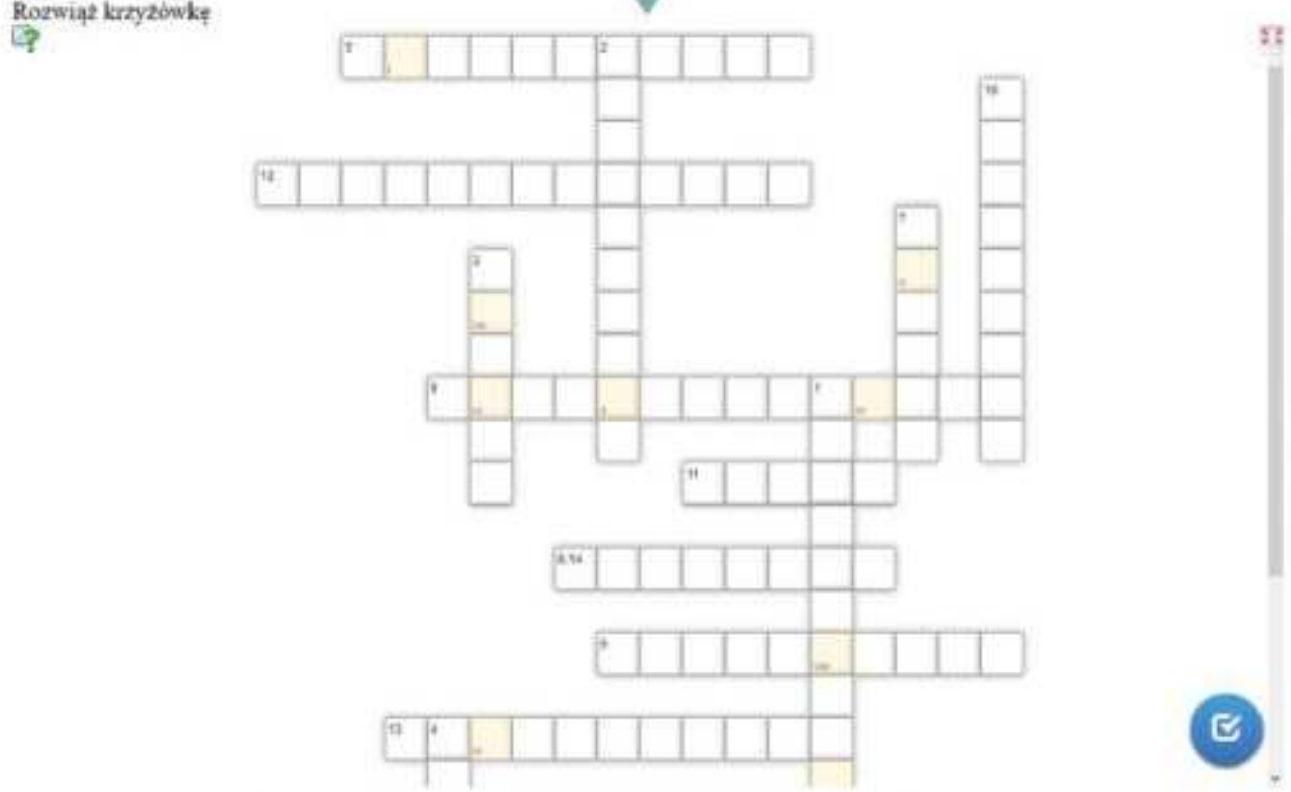
Wklejmy skopiowany fragment kodu (rys. 34.5.) do prostego kodu HTML (rys. 34.6.).

```
<iframe
src="https://learningapps.org/watch?app=2087703"
style="border:0px;width:100%;height:500px"
webkitallowfullscreen="true"
mozallowfullscreen="true"></iframe>
```

Dwie ostatnie linie mają zapewnić wyświetlanie aplikacji na pełnym ekranie w starszych wersjach przeglądarek Safari i Firefox.

Rys. 34.5. Skopiowany z learningapps.org kod HTML powodujący wyświetlenie utworzonej aplikacji w oknie strony internetowej

```html
<!DOCTYPE html>
<html>
        <head>
            <meta charset="UTF-8">
        </head>
        <body>
            Rozwiąż krzyżówkę
              <iframe src="https://learningapps.org/watch?app=2087703"
              style="border:0px;width:100%;height:500px"
              webkitallowfullscreen="true"
              mozallowfullscreen="true">
              </iframe>
        </body>
</html>
```



Rys. 34.6. Przykładowy kod HTML wyświetlający okno aplikacji z *learningapps.org*

Przeglądarka wyświetli tekst „Rozwiąż krzyżówkę". Oprócz tego, zgodnie z rysunkiem 34.1., odwoła się do serwera z aplikacją i wyświetli otrzymany od niego zasób.

Uwaga!

Strona do tworzenia aplikacji *learningapps.org* została użyta w celu przedstawienia mechanizmu działania stron dynamicznych. W swoich ćwiczeniach możesz wykorzystać inne usługi.

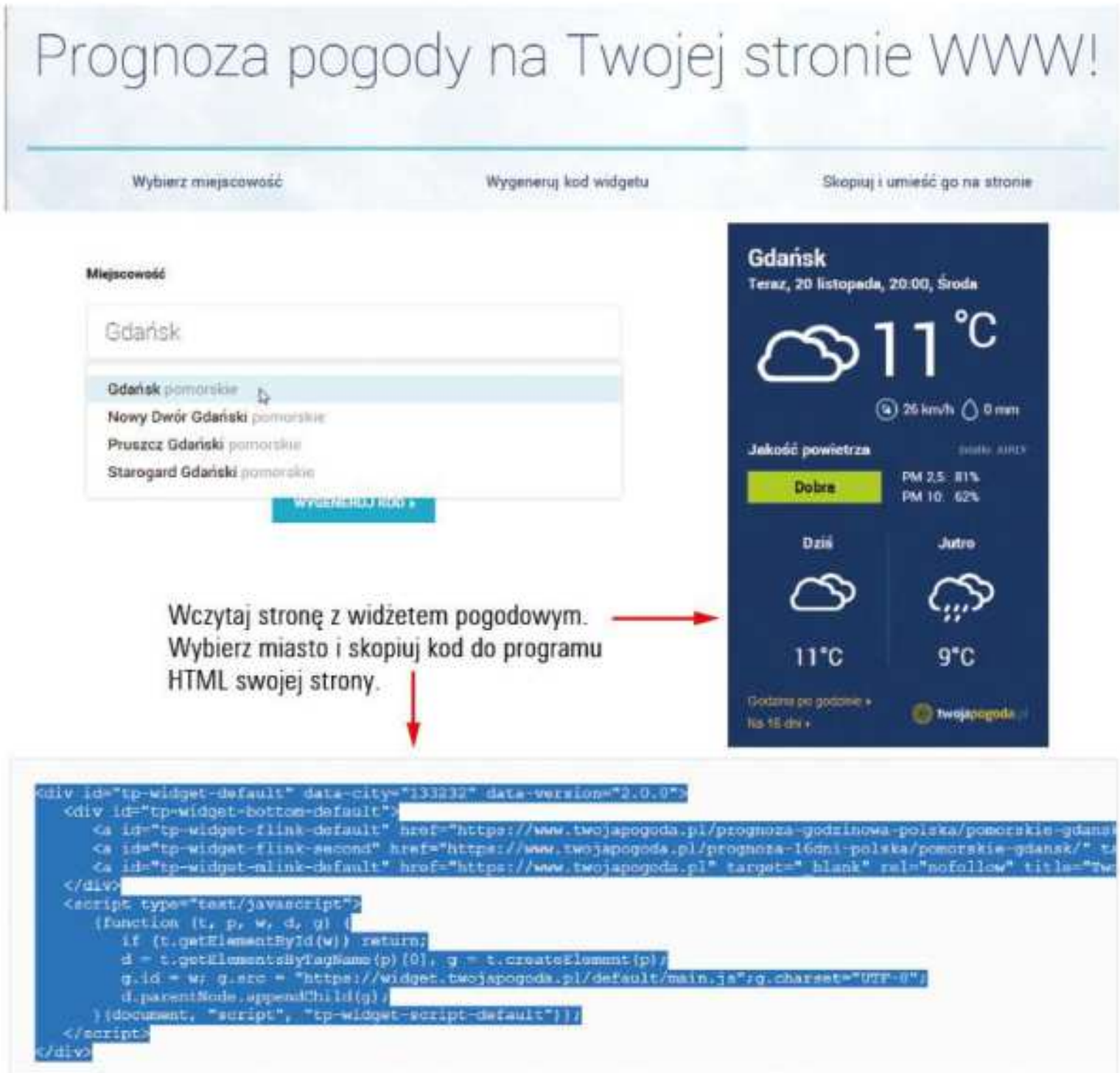
34.3. Pogoda na bieżąco, czyli dynamiczny widżet na stronie

W internecie znajdziesz wiele serwisów oferujących ciekawe widżety.

widżet ▸ **Widżet** (ang. *widget*) jest elementem graficznym umożliwiającym komunikację użytkownika z programem. Może być przyciskiem, polem edycyjnym, listą

rozwijaną itp. Na stronach internetowych często pełni funkcję **gadżetu**, np. wyświetla czas do końca roku szkolnego, pogodę, notowania walut itp.

Załóżmy, że na swojej stronie chcesz wyświetlać aktualny stan pogody. Oczywiście bez stacji pogodowej trudno byłoby ci uzyskać takie dane. Możesz jednak wykorzystać widżet pogodowy. Znajdziesz go za pośrednictwem wyszukiwarki. Przykładem może być *www.twojapogoda.pl/widget-na-twoja-strone/*. Podobnie jak w przypadku aplikacji (krzyżówki), tak i tutaj można pobrać gotowy kod do wklejenia do programu HTML (rys. 34.7.). Tym razem zawiera on także skrypt w języku JavaScript (czyt. dżawaskript). Aby uruchomić go na swojej stronie, nie musisz wiedzieć, jak on działa.



Rys. 34.7. Przykład pobierania widżetu pogodowego do własnej strony HTML

Uwaga!

Strona *twojapogoda.pl* została użyta jako przykład usługi udostępniającej widżety pogodowe. W swoich ćwiczeniach lub stronach możesz użyć dowolnej.

ZADANIA DO WYKONANIA

1. Odwiedź kilka portali informacyjnych. Sprawdź, czy są na nich elementy świadczące o tym, że są to strony dynamiczne. Wskaż je i uzasadnij swój wybór.
2. Za pomocą *learningapps.org* utwórz aplikację pomagającą zapamiętać informacje z niniejszego rozdziału. Umieść ją na swojej stronie. Sprawdź poprawność działania i możliwość wyświetlania na całej stronie okna przeglądarki.
3. Wyszukaj strony oferujące ciekawe widżety. Sprawdź ich działanie poprzez umieszczenie ich na swojej stronie. Oceń ich przydatność z punktu widzenia portalu informacyjnego.
4. Na jednej stronie HTML umieść kilka widżetów pogodowych z różnych serwisów i porównaj wyświetlane przez nie dane. Wybierz jeden, który według ciebie jest najatrakcyjniejszy. Uzasadnij swoją decyzję.

5*. Sprawdź działanie kodu z rysunku 34.5. pozbawionego dwóch ostatnich linii. Na podstawie informacji z internetu, forów dyskusyjnych itp. uzasadnij brak zmian na stronie w działaniu pobranego kodu.

PODSUMOWANIE LEKCJI

s. 219 • **Dynamiczna strona** internetowa wykorzystuje działanie aplikacji internetowej.

s. 219 • **Aplikacja internetowa** modyfikuje stronę, wyświetlając na niej elementy zgodne z parametrami podanymi przez użytkownika, np. pogodę w danym mieście, zgodnie z danymi z bazy danych itp.

s. 222 • W sieci istnieją usługi oferujące aktualne informacje, np. prognozy pogody, notowania walut itp. w postaci **widżetów**. **Widżet** na stronie internetowej pełni często funkcję **gadżetu** wyświetlającego informacje, rysunki itp. Jest także aktywnym odsyłaczem.

s. 223 • **Umieszczenie widżetu na stronie** następuje poprzez wklejenie kodu udostępnionego przez właściciela widżetu do swojej strony HTML.

35. Widoczna w internecie, czyli jak opublikować stronę

NA TEJ LEKCJI:

- dowiesz się, czym jest domena i jak ją zarezerwować;
- założysz konto na serwerze WWW i prześlesz do serwera elementy strony.

35.1. Adres strony, czyli rejestrujemy domeny

Wiesz już, że każdy komputer w sieci ma swój unikatowy adres IP. Serwer stron także jest komputerem, a więc również ma taki adres. Oczywiście jest on stały i niezmienny, dlatego możemy bez problemów odwoływać się do jego zasobów, wykorzystując ten sam IP. Istnieje jednak możliwość wybrania adresu, który co pewien czas się zmienia. W związku z tym ustawienie serwera z takim łączem jest kłopotliwe. Należy wówczas korzystać z pośrednictwa odpowiedniej usługi. Dzięki niej nasz komputer mimo zmiennego IP przydzielonego przez dostawcę internetu będzie w sieci widoczny pod stałym adresem.

◂ domena – adres symboliczny

Wykorzystanie **domen**, czyli adresów symbolicznych, znacznie ułatwia użytkownikom internetu wczytywanie stron. Proces ten może przebiegać jak na rys. 35.1.



Rys. 35.1. Proces wczytywania strony do przeglądarki internetowej na podstawie domeny

◂ rodzaje DNS

Można wyróżnić dwa rodzaje serwerów DNS. W jednym są przechowywane adresy IP związane z danymi domenami (**serwery autorytatywne**). Łączą się one jedynie z innymi serwerami DNS. Drugi rodzaj (**serwery rekurencyjne**) to serwery odpowiedzialne za odnajdowanie IP domen na podstawie zapytań, np. przeglądarek. Odnajdują je oczywiście w serwerach autorytatywnych.

Mechanizm jest więc stosunkowo prosty, choć rozwiązania techniczne i informatyczne są oczywiście bardzo skomplikowane. Ale skąd się biorą domeny i jak je przyporządkować do konkretnego IP? Domenę należy oczywiście kupić. Pierwsza opłata jest zazwyczaj bardzo niska. Aby jednak domenę utrzymać, należy opłacać roczny abonament.

◂ domeny najwyższego poziomu i domeny krajowe

Istnieją **domeny najwyższego poziomu**, czyli gTLDs (ang. *generic Top-Level Domains*, czyt. dżenerik top-lewel domejns), ustawione najwyżej w hierarchii, np. com. Oprócz tego są domeny krajowe, ccTLD (ang. *country code Top-Level Domains*, czyt. kantri kołd top-lewel domejns), np. *.pl*, *.de*, a także domeny

regionalne, np. *opole.pl*, *bydgoszcz.pl*, *malopolska.pl*. Podkreślają one miejsce działanie danej instytucji, zamieszkania autora itp.

Kto sprzedaje domeny? Istnieje wiele instytucji i firm, tak zwanych rejestratorów, którzy za pośrednictwem swojej strony sprzedają domeny.

Sprawdźmy, jak przebiega rejestracja domeny.

Uwaga!

Poniższy przykład dotyczy jednego z wielu rejestratorów. Proces rejestracji u pozostałych przebiega bardzo podobnie.



Rys. 35.2. Przykład procesu rejestracji domeny

Uwaga!

Jeśli nie chcesz zamówić domeny, a jedynie sprawdzić, jak przebiega sam proces, nie potwierdzaj zakupu. Dla uniknięcia kosztów ogranicz się jedynie do sprawdzenia, czy dana domena jest wolna.

subdomena ▸ Właściciel domeny może utworzyć samodzielnie **subdomenę**. Tworzy ją nazwa umieszczona przed domeną, np. *forum.lo3info.pl*.

35.2. Miejsce na stronę, czyli jak zamówić usługę hostingu

Wraz z domeną można kupić także pewne usługi, np. hosting strony, poczty i dostęp do edytora tworzenia stron. Nie wymaga on znajomości języków programowania, dlatego staje się pomocny w tworzeniu niewielkich publikacji internetowych. W sieci znajdziesz wielu usługodawców oferujących takie

rozwiązania. Zanim jednak zdecydujesz, który będzie dla ciebie odpowiedni, zasięgnij informacji na forach i portalach społecznościowych.

Hostingiem nazywamy udostępnianie usług internetowych, np. na serwerze miejsca na stronę WWW czy pocztę elektroniczną. Usługodawca zapewnia odpowiednią ilość miejsca na dysku serwera, łącza o wystarczającej przepustowości, oprogramowanie, np. bazy danych i inne zawarte w umowie usługi. ◂ hosting

Niektóre firmy świadczą **darmowe usługi hostingowe** w zamian za zgodę na wyświetlanie reklam na naszych stronach. Warto jednak z nich skorzystać, np. gdy chcemy sprawdzić, jak strona będzie się prezentowała w sieci. Zawodowi webmasterzy mają na to inne sposoby, początkującym pozostaje opisane rozwiązanie. Wiąże się to także z użyciem domeny danej usługi.

Jako klient usługi hostingowej, będziesz musiał/musiała zmierzyć się z problemem wysłania swojej strony na serwer. Posłuży do tego protokół FTP.

FTP (ang. *File Transfer Protocol*, czyt. fajl transfer protokol) umożliwia przesyłanie plików (np. plików stron internetowych) na serwer oraz ich pobieranie poprzez sieć. ◂ FTP

Z FTP można korzystać w przeglądarkach internetowych, a także za pośrednictwem klientów – programów obsługujących ten protokół. Niektórzy usługodawcy hostingu oferują własne narzędzia to transmisji plików.

Jednym z wielu dostawców usług hostingowych jest *cba.pl*. Na przykładzie jego strony prześledźmy proces rezerwowania miejsca na stronę WWW. W przypadku wyboru darmowej wersji nasza strona zyska adres będący subdomeną, np. *lo3info.cba.pl*.



Rys. 35.3. Przykład zakładania konta na darmowym serwerze kont dla stron WWW

Usługodawca może także zaoferować kilka dodatkowych usług (rys. 35.4.).



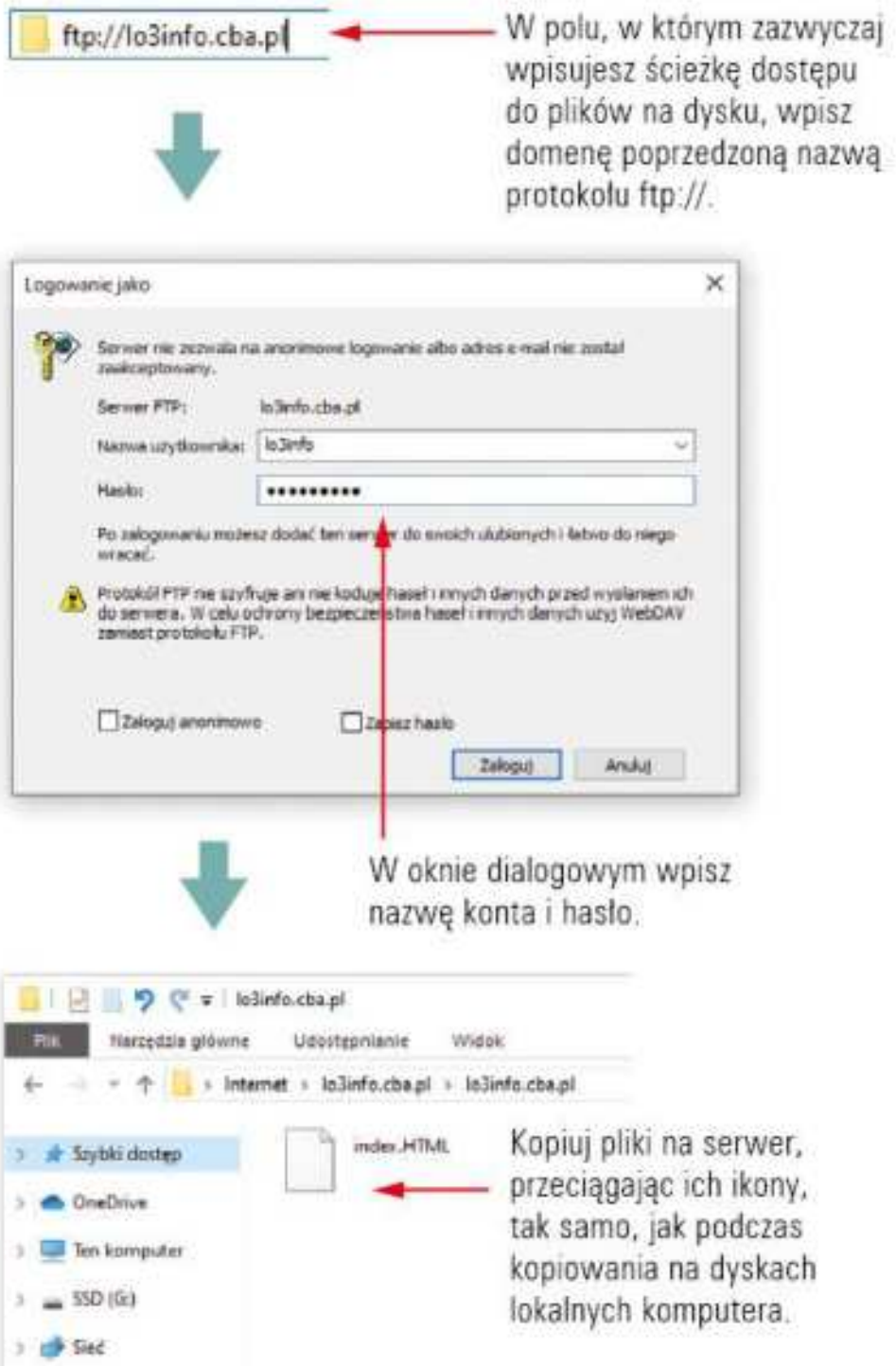
Rys. 35.4. Przykładowe dodatkowe opcje oferowane przez usługodawców hostingowych

Serwer stron można także ustawić na własnym komputerze-serwerze. Jednak wymaga to stałego adresu IP i dobrych zabezpieczeń przed nieuprawnionym dostępem do kont.

35.3. Wysyłamy stronę do serwera, czyli korzystamy z protokołu FTP

Gdy twoje konto na serwerze i usługa hostingowa będą aktywne, możesz wysyłać pliki strony. Istnieje kilka sposobów:

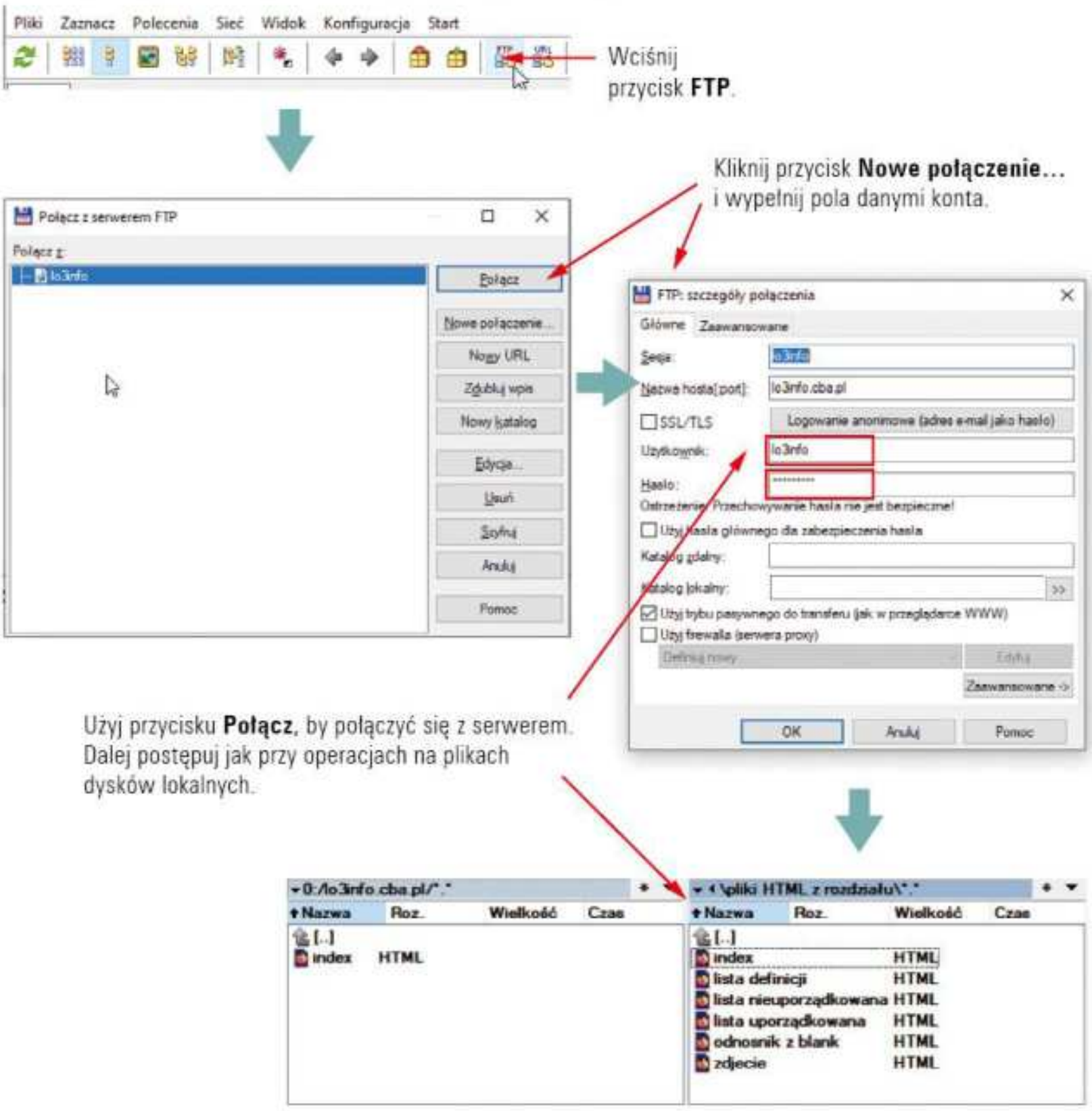
1. Korzystanie z Eksploratora plików Windows (rys. 35.5.)



Rys. 35.5. Przykład użycia **Eksploratora plików** z Windows 10 do przesyłania plików z zastosowaniem protokołu FTP

2. Korzystanie z klienta FTP

Jako klienta FTP można wykorzystać program do zarządzania plikami, np. Total Commander (czyt. total komander, rys. 35.6.), FileZilla (czyt. fajlzila) lub programy przeznaczone dla tego protokołu, np. CuteFTP (czyt. kjuteftipi). Należy jednak pamiętać, że część z nich to programy komercyjne.



Rys. 35.6. Przykład zastosowania programu **Total Commander** do przesyłania plików protokołem FTP

Dla sprawdzenia działania serwera prześlij jeden z plików HTML z poprzednich rozdziałów.

ZADANIA DO WYKONANIA

1. Wyszukaj w internecie strony, na których są świadczone usługi rejestrowania domen. Przyjmij, że chcesz założyć stronę swojej szkolnej drużyny piłkarskiej. Porównaj ceny i opinie na forach oraz portalach społecznościowych. Wybierz najlepszą ofertę i uzasadnij swój wybór.
2. Używając dowolnego edytora grafiki, narysuj graf ilustrujący mechanizm zamiany adresu symbolicznego (domeny) na adres IP serwera. Zanim przystąpisz do pracy, przeczytaj artykuły z zaufanych portali poświęconych tematyce sieciowej.
3. Wybierz jednego z dostawców darmowych usług hostingowych, na którego serwerze chcesz umieścić swoją stronę. Wyjaśnij dokładnie, czym się kierowałeś/kierowałaś, dokonując wyboru.
4. Zapoznaj się z możliwościami różnych programów do przesyłania plików za pośrednictwem protokołu FTP. Wybierz twoim zdaniem najlepszy i uzasadnij swój wybór.

5*. Jeśli masz konto na serwerze FTP, sprawdź, czy jest możliwe przesyłanie plików za pośrednictwem przeglądarki internetowej.

PODSUMOWANIE LEKCJI

s. 225 • Dla ułatwienia korzystania z sieci WWW zastosowano łatwe do zapamiętania adresy – **domeny**.

s. 225 • Każdej domenie może być przypisany adres IP. Dane te są zapamiętane na serwerze **DNS**.

s. 225 • Wpisany do przeglądarki adres (domena) zostaje wysłany do DNS, który odczytuje z bazy adres IP serwera strony i zwraca go przeglądarce. Przeglądarka wczytuje stronę spod danego IP.

s. 227 • Do wysyłania plików na serwer służy **protokół FTP**.

s. 228 • Menadżer plików daje możliwość korzystania z FTP, np. Eksplorator plików z Windows.

36. CMS, czyli system zarządzania treścią strony internetowej

NA TEJ LEKCJI:

- dowiesz się, czym jest CMS;
- poznasz podstawowe pojęcia dotyczące systemów zarządzania treścią;
- zainstalujesz WordPress w środowisku Laragon.

Programowanie strony w HTML i CSS jest czasochłonne i wymaga wielu ćwiczeń i doświadczenia. A co zrobić, jeśli chcemy osiągnąć szybko zadowalający efekt bez biegłej znajomości potrzebnych języków? Jest na to sposób.

36.1. Podstawy, czyli co musisz wiedzieć o CMS

Czy to możliwe, by twórca strony internetowej nie znał języka HTML? Możliwe. Do projektowania pewnej klasy stron można bowiem użyć specjalnego oprogramowania, np. uruchamianego na serwerze WWW. Sprawdza się ono w większości projektów o niezbyt rozbudowanej strukturze, np. stron szkolnych, witryn informacyjnych, albumów ze zdjęciami, portfolio itp. Niektóre pozwalają na tworzenie bardziej złożonych koncepcji, np. sklepów internetowych, stron z filmami itp. Całą klasę takich programów nazwano systemami zarządzania treścią (ang. CMS – *Content Management System*, czyt. kontent manadżment system).

◂ system zarządzania treścią

CMS jest to oprogramowanie pozwalające na łatwe i szybkie zarządzanie treściami stron (dodawanie i usuwanie treści, zmiany wyglądu itp.) oraz ich tworzenie za pomocą panelu administracyjnego.

Ponieważ większość z tych programów jest bardzo funkcjonalna, sprawna i wygodna w użyciu, wykorzystują je nawet profesjonaliści, którzy jednak często używają zaprojektowanych przez siebie interfejsów stron (ang. *layout*, czyt. lejaut), co czyni je oryginalnymi i spełnia wymagania klientów.

◂ layout

Layout jest elementem strony decydującym o jej wyglądzie i funkcjonalności w kontakcie z odbiorcą jej treści. Obejmuje wszystkie elementy wizualne, łącznie z ich ułożeniem i funkcjami, jakie pełnią. Pojęcie to dotyczy wszystkich stron, niezależnie od narzędzia, za pomocą którego zostały utworzone.

W przypadku CMS tworzy się skórki (ang. *skins*, *themes*, czyt. skins, tims), które generują wygląd i elementy aktywne strony. Dzięki temu projektant nie musi samodzielnie tworzyć jej layoutu.

◂ skórka dla CMS

Skórki to przygotowane przez twórców lub użytkowników CMS gotowe interfejsy użytkowników (layouty). Są dostępne w internecie, ale można je tworzyć

lub modyfikować samodzielnie. Część skórek jest darmowa, a ich używanie jest dozwolone w projektach komercyjnych. Niektóre jednak trzeba kupić.

Obecnie w sieci funkcjonuje wiele darmowych i ogólnodostępnych systemów zarządzania treścią. Należy jednak pamiętać, że jeśli samodzielnie ułożysz program strony internetowej i wyposażysz ją w panel administracyjny pozwalający na zmiany i aktualizację treści, utworzysz CMS. Jednak znacznie częściej używa się sprawdzonych i bezpiecznych programów dostępnych w sieci.

Najpopularniejsze obecnie systemy zarządzania treścią to **WordPress**, **Drupal**, **Joomla**. Jeśli przeszukasz sieć, zapewne znajdziesz ich znacznie więcej. Niektóre są polecane przez twórców profesjonalnych stron. Popularność zależy od wielu ich cech, np. prostoty tworzenia i modyfikacji, ciekawych skórek, bezpieczeństwa, tagowania, tworzenia i gromadzenia statystyk itp.

36.2. Jak zacząć, czyli instalacja

Trudno będzie ci wybrać właściwy CMS dla twojego projektu, jeśli nie znasz cech żadnego z nich. Zanim więc dokonasz wyboru, zapoznaj się opisami, opiniami i przejrzyj strony wykonane za pomocą danego systemu. Ważnymi argumentami będą także dostępne skórki i komponenty. Zapisz wszystkie za i przeciw, a potem dokonaj wyboru.

Wiesz już, że twoja strona musi być umieszczona na serwerze WWW. Wiele usług hostingowych oferuje także gotowe mechanizmy instalacji CMS (rys. 36.1.).



Rys. 36.1. Przykładowy proces instalacji systemu zarządzania treścią dla kont hostingowych wyposażonych w aplikację wspomagającą

Jeśli twój serwer nie ma takiej aplikacji, o instalację musisz zadbać samodzielnie. Niestety, nie jest to tak łatwe, jak użycie odpowiedniej aplikacji.

Przygotowania do instalacji:

1. Upewnij się, czy masz odpowiednią ilość wolnego miejsca na dysku serwera (w zależności od tego, ile wymaga wybrany CMS).
2. Sprawdź, jakie oprogramowanie serwerowe musi być jeszcze uruchomione na serwerze (np. bazy danych).
3. Jeśli w danych usługi hostingowej nie otrzymałeś/otrzymałaś nazwy ani hasła do bazy danych, załóż ją samodzielnie. Jej nazwa i hasło będą niezbędne do instalacji.
4. Pobierz CMS z wiarygodnego źródła. Najczęściej będzie to archiwum skompresowane ZIP.
5. Jeśli na serwerze działa program do dekompresji plików ZIP, wyślij archiwum do folderu głównego swojego konta i je rozpakuj. Jeśli nie ma takiego

programu, rozpakuj plik na swoim dysku i wyślij do serwera, zachowując pełną strukturę i nazwy folderów.

6. Uruchom za pomocą przeglądarki plik instalujący CMS na serwerze, a następnie postępuj według wskazań instalatora.

7. Wczytaj stronę i rozpocznij jej edycję w panelu zarządzania.

Jeśli nie masz konta na serwerze WWW, a chcesz ćwiczyć tworzenie stron w wybranym CMS, możesz użyć programu, który na twoim komputerze utworzy środowisko serwera ze wszystkimi potrzebnymi komponentami. Przykładami są **Xampp** (eksamp) oraz **Laragon**.

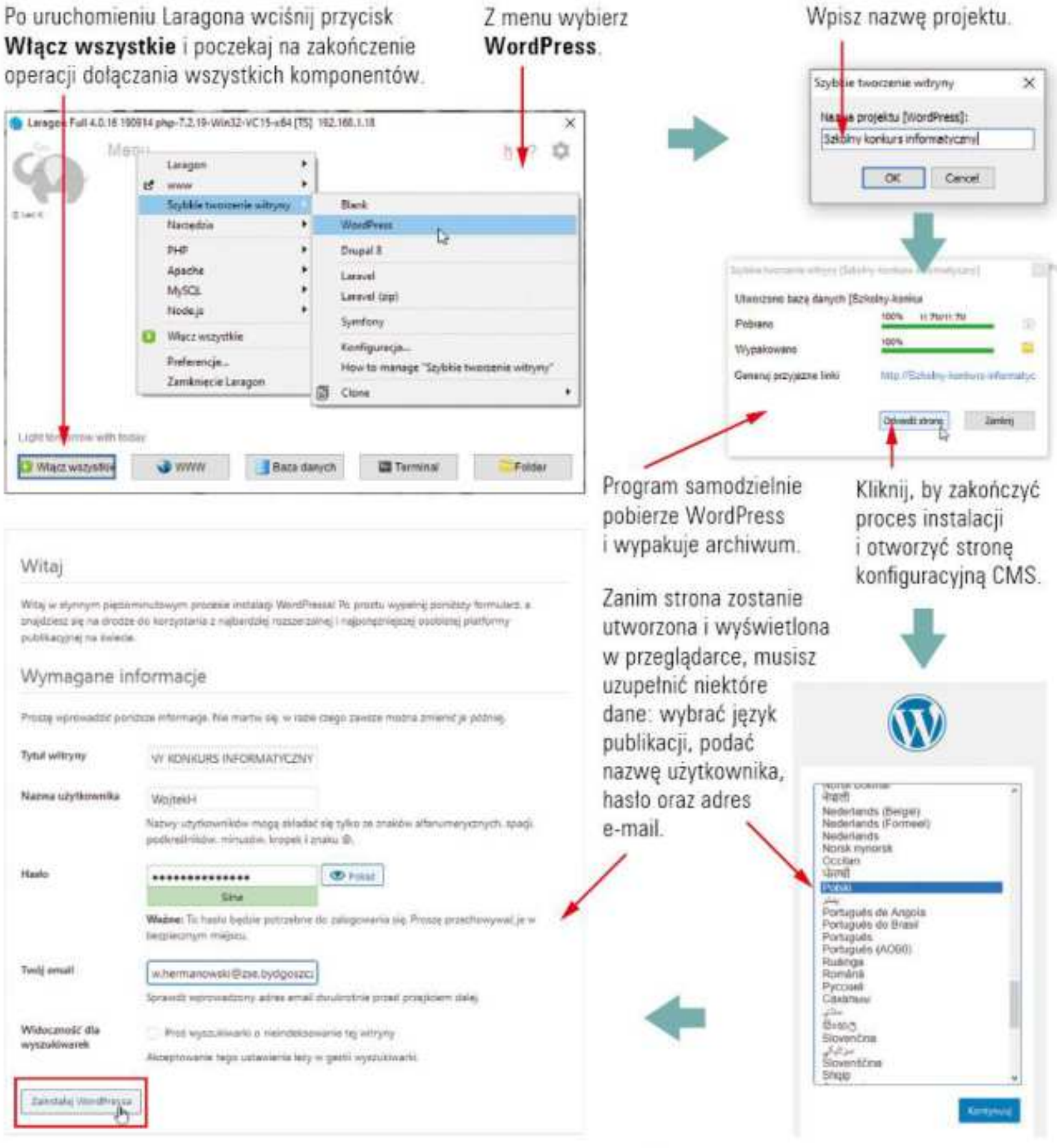
W naszym przykładzie użyjemy Laragona. To darmowe narzędzie dostępne pod adresem *laragon.org*. Po pobraniu pliku instalacyjnego wystarczy go uruchomić i wskazać miejsce na dysku, w którym ma zostać zainstalowany (rys. 36.2.).

Po uruchomieniu pliku instalacyjnego programu Laragon wybierz wersję językową.



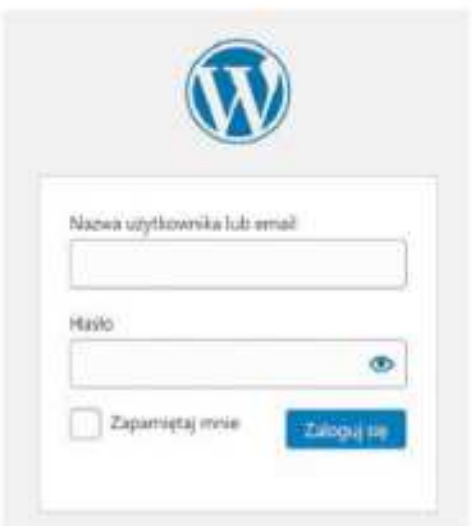
Rys. 36.2. Proces instalacji środowiska Laragon

Środowisko Laragon zainstalowane w twoim komputerze pozwoli na uruchamianie i testowanie stron internetowych w HTML, CSS i innych językach. Nam jednak zależy na zainstalowaniu CMS bez konieczności korzystania z usługi hostingowej. W przypadku Laragona przygotowanie CMS do pracy jest łatwe. Program wykona większość czynności, takich jak założenie bazy danych, automatycznie. Samodzielnie pobierze i zainstaluje także najnowszą wersję CMS. W przykładzie użyjemy **WordPress** (czyt. łordpres), ponieważ jest to jeden z najpopularniejszych systemów zarządzania treścią (rys. 36.3.). Mimo że ma wiele możliwości, jest łatwy w obsłudze. Utworzono dla niego także wiele skórek, z których można bezpłatnie skorzystać.



Rys. 36.3. Proces instalacji CMS WordPress w środowisku Laragon

Nasza przykładowa strona będzie dostępna w przeglądarce komputera pod adresem *http://szkolny-konkurs-informatyczny.test*. Oczywiście, jeśli użyjesz w trakcie instalacji CMS innej nazwy dla projektu, adres będzie z nią zgodny. Panel sterowania (w WordPress nazwany Kokpitem) przykładowego projektu uruchomisz pod adresem *http://szkolny-konkurs-informatyczny.test/wp-admin/index.php*. Jest on chroniony hasłem, które zostało wpisane w trakcie instalacji (rys. 36.4.).



Rys. 36.4. Logowanie do panelu sterowania (**Kokpitu**) CMS WordPress

Dzięki użyciu programu Laragon uzyskaliśmy możliwość testowania stron internetowych bez konieczności wysyłania ich do serwera w usłudze hostingowej.

ZADANIA DO WYKONANIA

1. Korzystając z dowolnego edytora, wykonaj prezentację, w której opiszesz zalety i wady kilku wybranych systemów zarządzania treścią. Zanim rozpoczniesz pracę, dokonaj ich selekcji na podstawie informacji z wiarygodnych źródeł, np. artykułów w czasopismach. Weź pod uwagę popularność, atrakcyjność skórek itp.
2. Przejrzyj oferty usług hostingowych pod kątem używania CMS. Sprawdź, które z nich proponują odpowiednie narzędzia oraz jakie systemy zarządzania treścią są w ofercie. Wybierz najlepszą według ciebie usługę i omów proces wyboru. Podaj źródła, które pozwoliły ci podjąć decyzję.
3. Po zainstalowaniu środowiska Laragon sprawdź doświadczalnie, czy możesz tworzyć w nim wiele stron, korzystając z różnych systemów zarządzania treścią. Omów swoje doświadczenia, ilustrując je zrzutami ekranowymi.
4. Znasz już Laragon. Poszukaj w sieci innych programów tego typu, np. Xampp. Zapoznaj się z ich możliwościami i opiniami na ich temat. Porównaj je z Laragonem. Omów swoje spostrzeżenia i opinie.
5*. Pobierz z wiarygodnej strony Xampp. Zainstaluj go w swoim komputerze. Przeprowadź proces instalacji WordPress. Omów ten proces i wskaż różnice w porównaniu z Laragonem.

PODSUMOWANIE LEKCJI

s. 231 • **CMS** jest oprogramowaniem pozwalającym na latwe i szybkie tworzenie stron internetowych oraz zarządzanie jej treściami.

s. 231 • Istnieją **CMS wykorzystywane przez webmasterów** dostępne w internecie. Możesz jednak **samodzielnie utworzyć** stronę z panelem administracyjnym pozwalającym na zarządzanie jej treścią. To też będzie CMS.

s. 231 • **Layout** jest elementem strony decydującym o jej wyglądzie.

s. 231 • **Skórka** w CMS jest gotowym do użycia **layoutem**. Można ją pobrać z sieci, kupić lub wykonać samodzielnie.

s. 232 • **WordPress**, **Drupal**, **Joomla** to przykłady często stosowanych systemów zarządzania treścią.

s. 233 • Do funkcjonowania CMS jest potrzebne konto w **bazie danych serwera**.

s. 233 • CMS lub dowolne projekty stron internetowych można testować w środowisku emulującym działanie serwera WWW, np. **Laragon** lub **Xampp**.

37. Panel i skórki, czyli tworzymy stronę w CMS

NA TEJ LEKCJI:

- nauczysz się używać panelu zarządzania CMS do tworzenia strony i zarządzania jej treścią;
- dobierzesz, zainstalujesz i zmodyfikujesz skórkę dla swojej strony;
- dołączysz użytkownika do zespołu zarządzającego stroną.

Umiesz już instalować CMS i logować się do panelu zarządzania CMS WordPress. Domyślasz się, że używanie innych systemów zarządzania treścią opiera się na podobnych zasadach. Skoro w twoim komputerze działa już środowisko serwerowe z zainstalowanym WordPressem, możesz przystąpić do budowania strony.

37.1. Skórka, czyli pierwszy krok w budowie strony w CMS

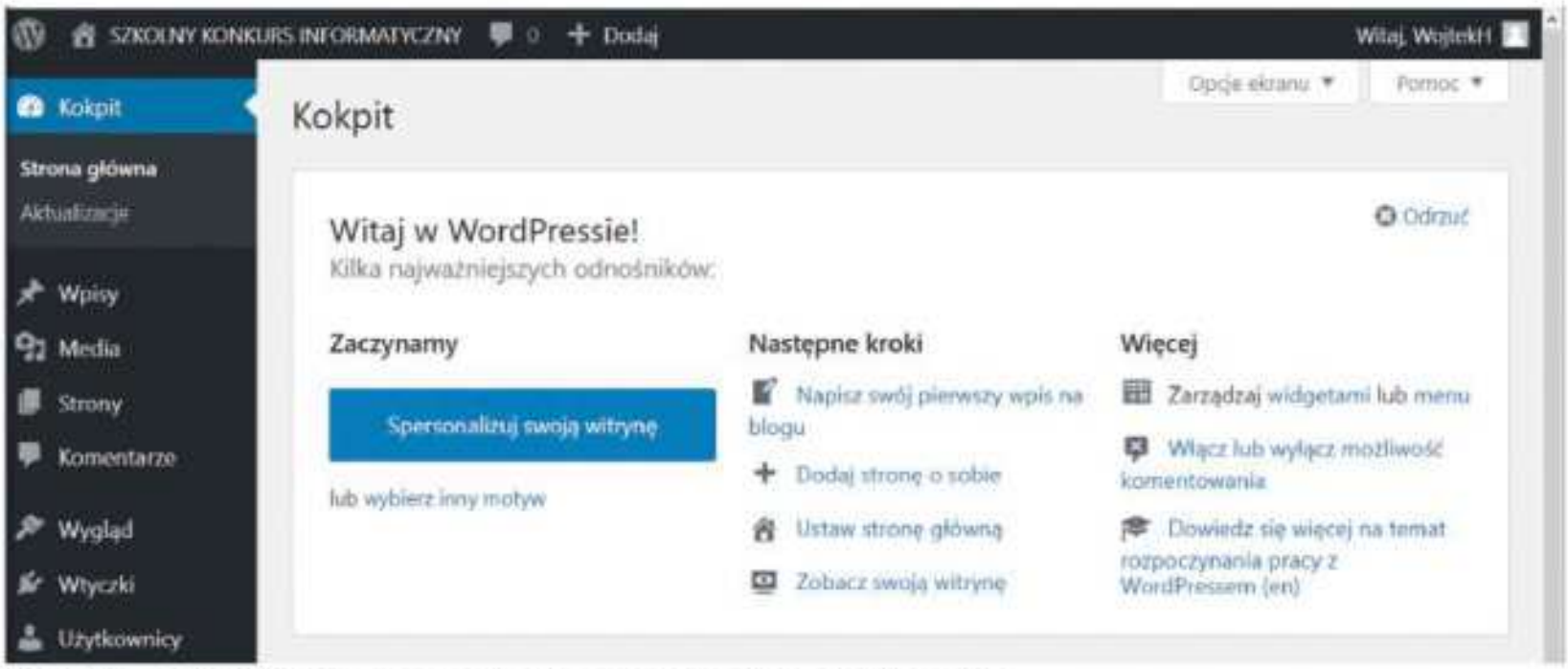
skórka, czyli szablon

Zanim zaczniesz szukać odpowiedniej skórki (szablonu) dla swojej strony, spróbuj dopasować jedną z oferowanych przez WordPress. Zacznij od uruchomienia środowiska Laragon i Kokpitu WordPressa. Jeśli wykonałeś/wykonałaś ćwiczenie opisane w rozdziale 36., będzie on dostępny pod adresem *http://szkolny-konkurs-informatyczny.test/wp-admin/index.php*. W oknie logowania wpisz nazwę użytkownika i hasło nadane podczas instalacji CMS.

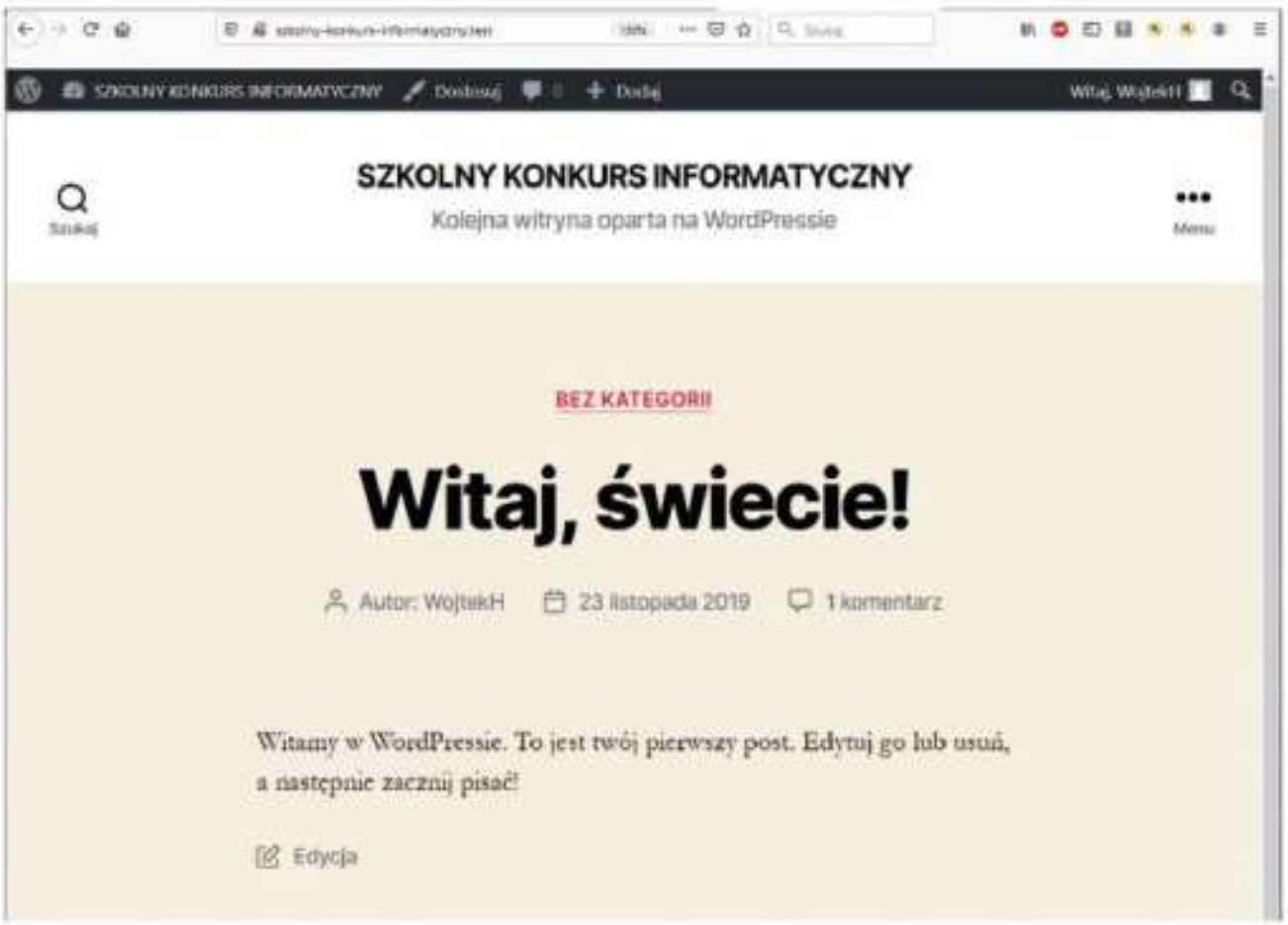
Uwaga!

Jeśli zmieniłeś/zmieniłaś nazwę projektu, zmodyfikuje ona adres twojego **Kokpitu** – wymieni się jego główna część (przed .test). Gdyby twoja instalacja WordPressa byłaby już na serwerze, do domeny dodaj /admin lub /wp-admin.

Kokpit twojego CMS przywita cię zachętą do skorzystania z samouczka (rys. 37.1.). Możesz wybrać dowolny krok i po kolei budować swoją stronę. Warto jednak najpierw sprawdzić, jaką domyślną skórkę ustawił WordPress (rys. 37.2.). Stronę znajdziesz pod adresem *http://szkolny-konkurs-informatyczny.test/*.

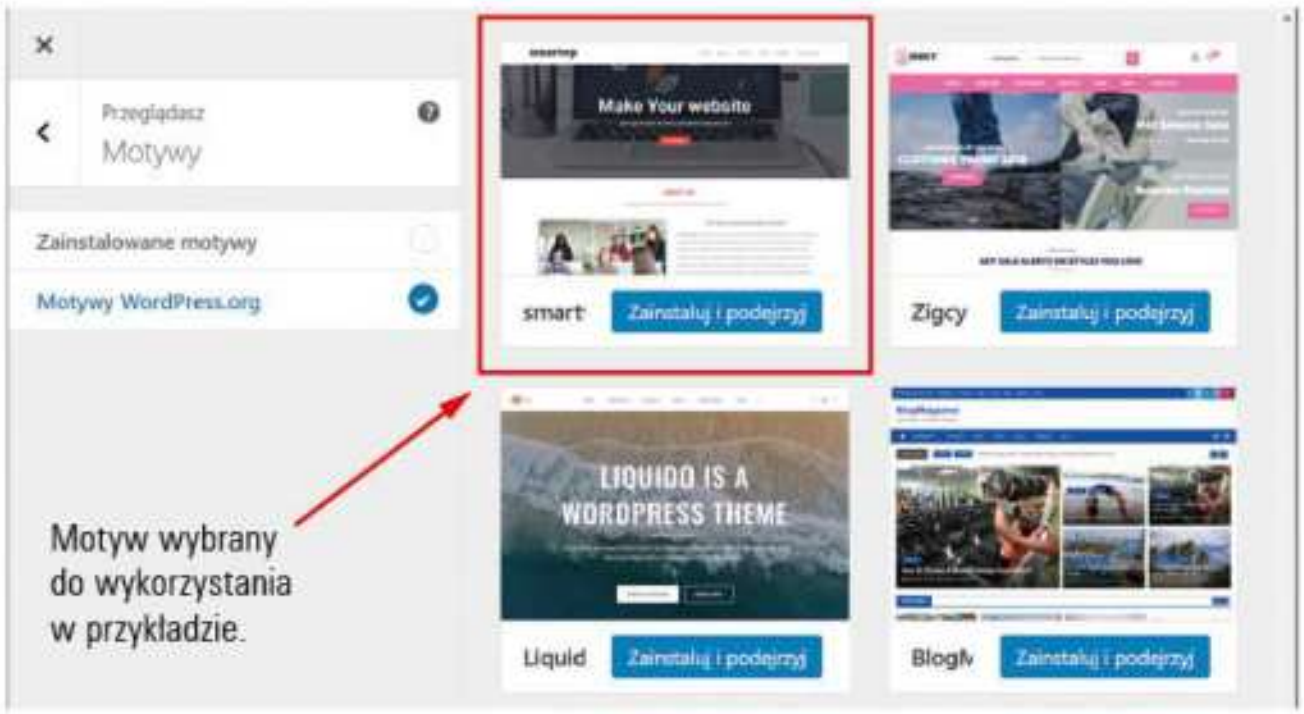


Rys. 37.1. Kokpit WordPressa z opcjami samouczka dla początkujących



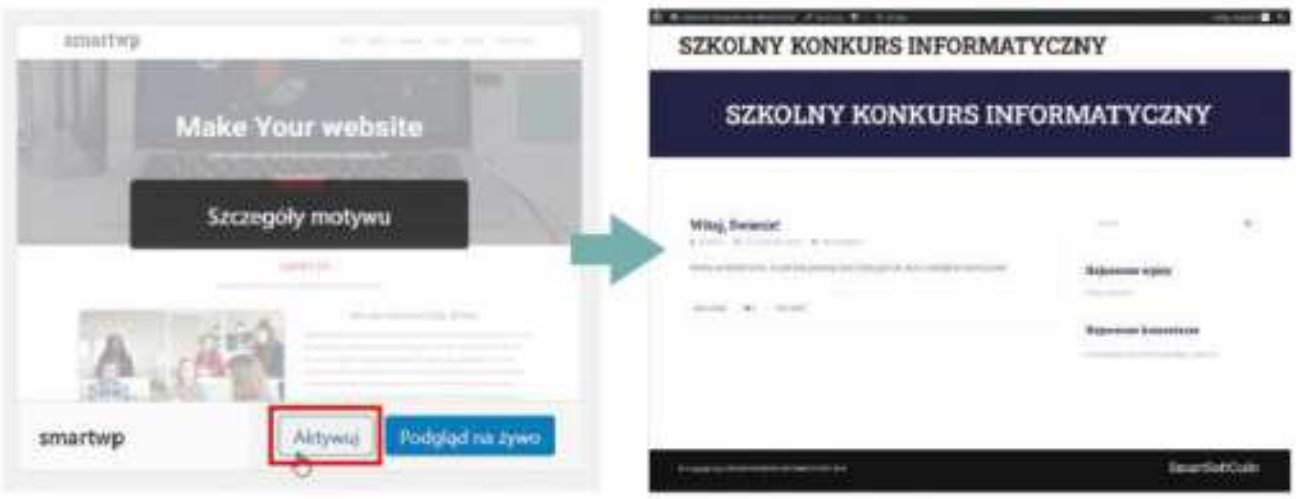
Rys. 37.2. Przykładowa, domyślna skórka strony

Przyznasz, że domyślna skórka nie wygląda atrakcyjnie. Wróć do **Kokpitu** i kliknij samouczek **wybierz inny motyw** lub opcję **Wygląd**. Tę drugą wybierz, gdy zamkniesz już blok samouczka **Witaj w WordPressie**. W naszej instalacji są już dostępne trzy inne motywy skórki. Jeśli nie podobają ci się te propozycje, możesz kliknąć opcję **Motywy WordPress.org** (rys. 37.3.) i wybrać jeden z nich, klikając przycisk **Zainstaluj i podejrzyj**.



Rys. 37.3. Przykładowe motywy do wyboru z kolekcji WordPress.org

To jednak nie koniec. W tym momencie motyw został pobrany i dołączony do naszego WordPressa. Należy go jednak aktywować w opcji **Wygląd** (rys. 37.4.).



Rys. 37.4. Aktywowanie pobranego motywu (skórki) w WordPress

37.2. Nie wszystko jest tak, jak planowaliśmy, czyli jak zmodyfikować motyw

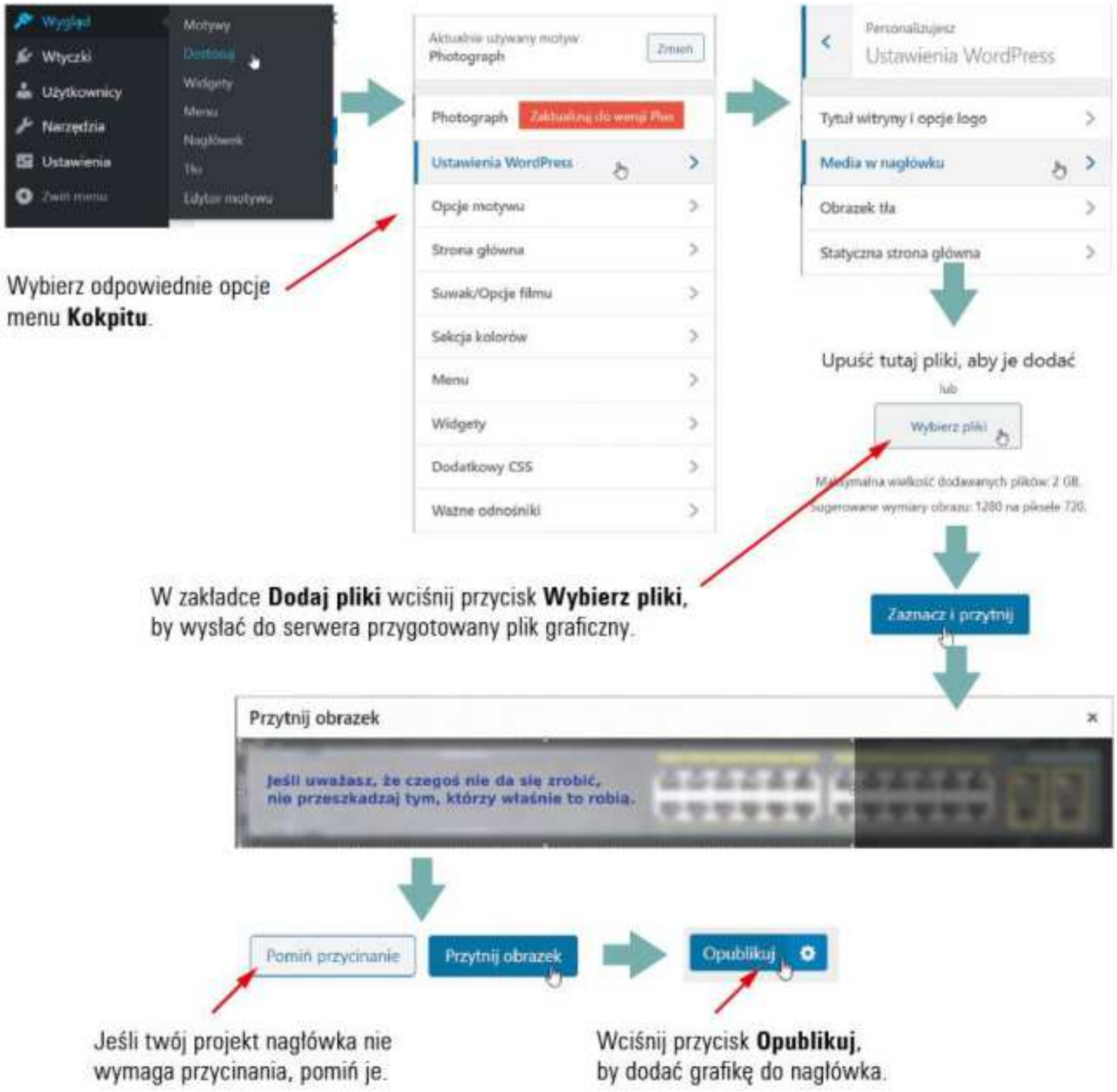
modyfikacja skórki ▸ Po wyświetleniu w przeglądarce nowej skórki naszego projektu strona nadal nie wygląda atrakcyjnie. Jest pozbawiona zdjęć i grafiki. Nie mamy wyjścia. Dodajmy te elementy z własnych zbiorów (rys. 37.5.) lub z sieci pod warunkiem, że będą na licencji CC. Musisz je jednak odpowiednio przygotować. Każdy z elementów skórki ma pewne ograniczenia, które dotyczą głównie rozmiarów ilustracji.

Przygotujmy pasek nagłówka (rys. 37.5.) w odpowiednich proporcjach. Dobierz je samodzielnie.



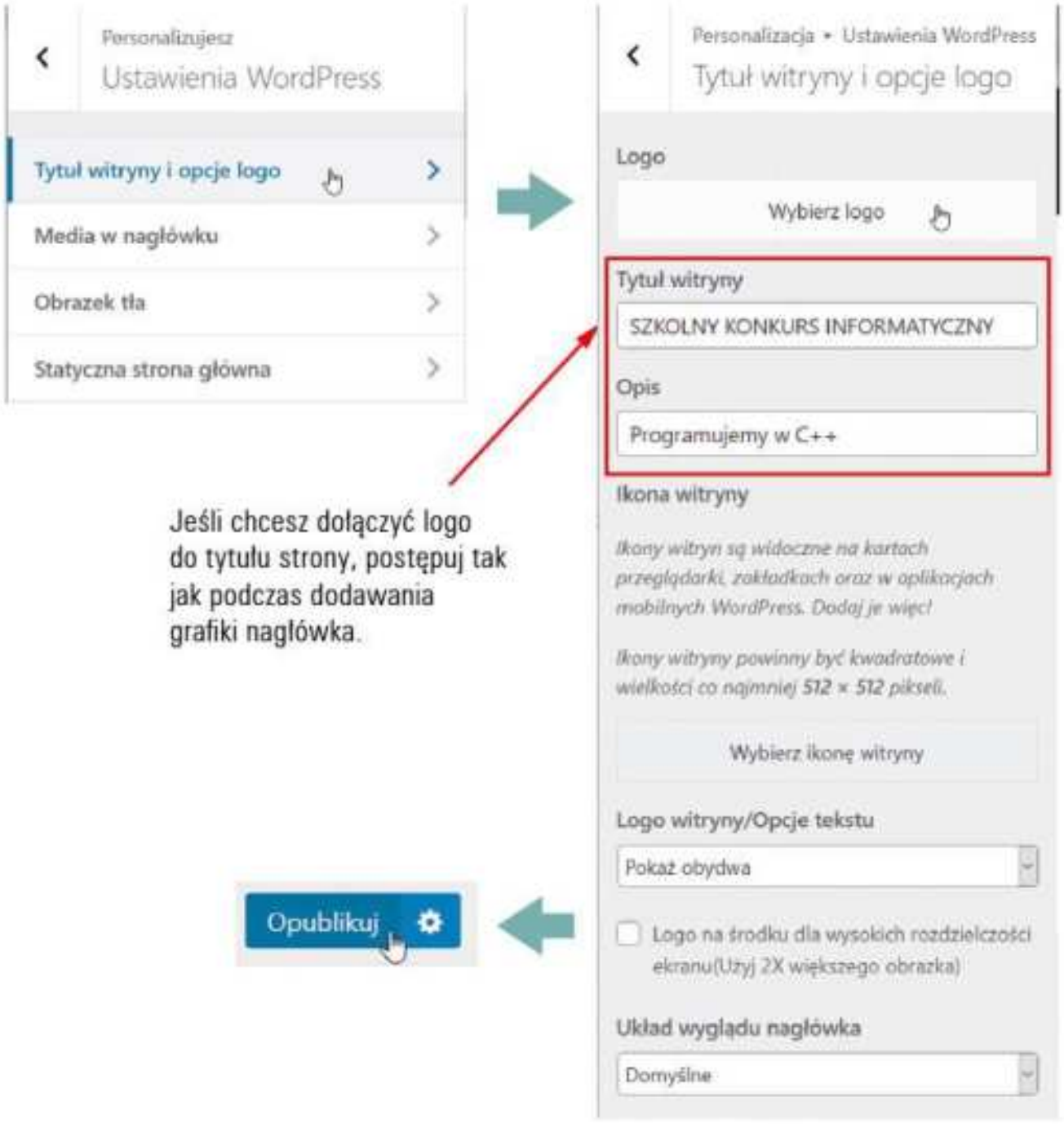
Rys. 37.5. Przykładowa grafika nagłówka strony

Wstawmy pasek nagłówka na naszą stronę (rys. 37.6.). Przygotowana grafika nie musi być dokładnie dopasowana do miejsca, w którym ma się znaleźć. W trakcie wklejania będzie okazja, by ją odpowiednio przyciąć.



Rys. 37.6. Przykład wstawiania grafiki do nagłówka skórki

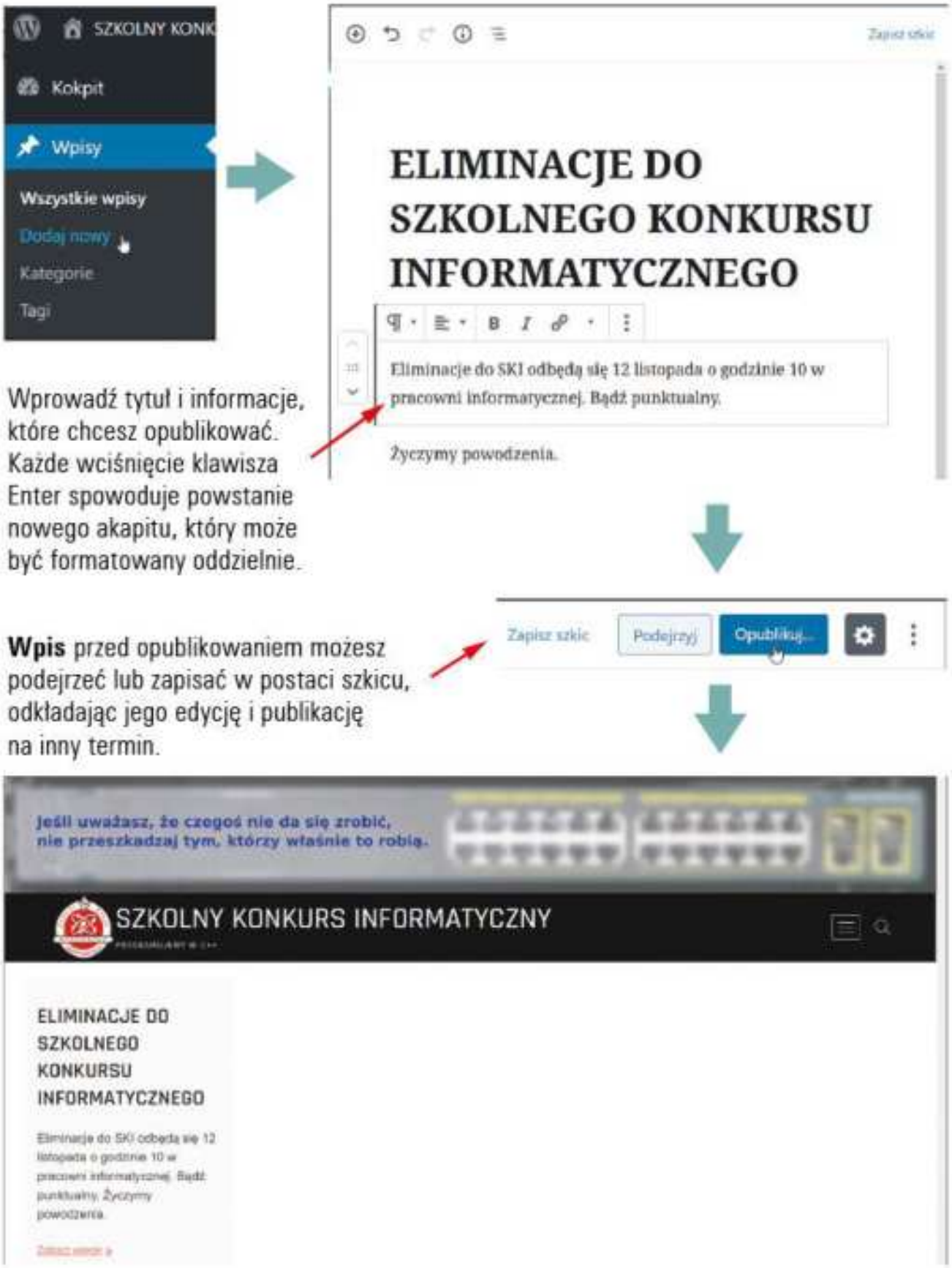
Kończąc modyfikację statycznych elementów motywu, zmieńmy tytuł i dodajmy logo (rys. 37.7.).



Rys. 37.7. Przykład modyfikacji tytułu i logo strony dla danej skórki

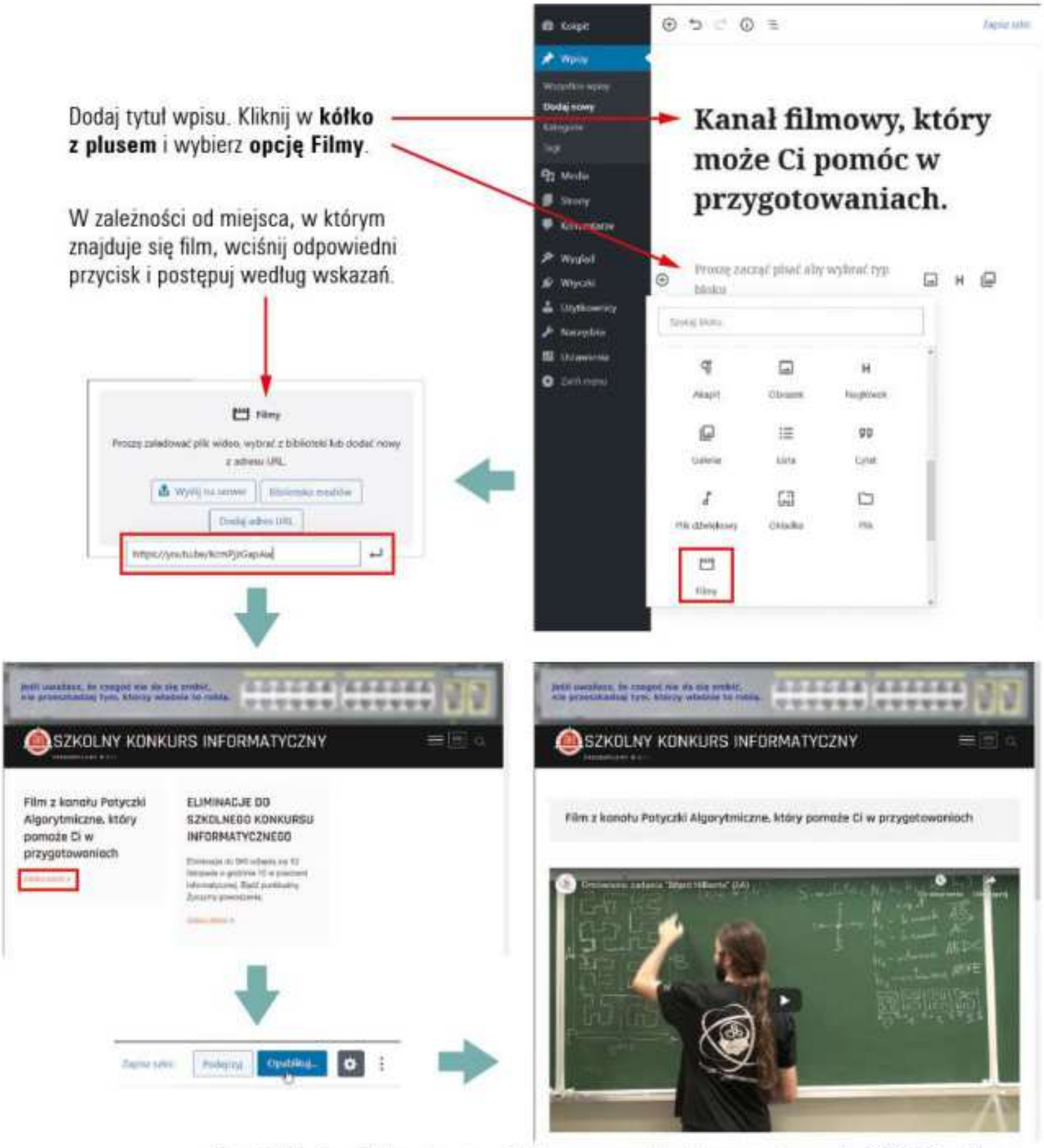
37.3. Wypełniamy treścią, czyli dodajemy wpisy do strony

wpisy do strony ▸ Dodajmy teraz pierwszy, krótki **wpis**. W każdym motywie są określone miejsca, w których pokazują się wpisy. Niektóre skórki pozwalają na modyfikację tych ustawień. W naszym przykładzie wpisy będą się pojawiały od lewej do prawej strony z efektem animacji. Wstawianie wpisów jest łatwe, należy jednak zwrócić uwagę na to, że mogą one zawierać wiele różnych form, np. grafikę, film, odsyłacze itp. Wprowadźmy krótką informacje tekstową (rys. 37.8.). Pomoże ci w tym system podpowiedzi ukazujący się w kolejnych fazach wprowadzania wpisu. Warto korzystać z tych porad.



Rys. 37.8. Przykład umieszczania wpisu na stronie w CMS WordPress

Warto teraz sprawdzić, jak umieszczać wpisy na przykład z filmem. Wykorzystaj umiejętności zdobyte podczas dodawania wpisu tekstowego i postaraj się znaleźć odpowiednią opcję. Oczywiście musisz także znać odsyłacz lub mieć plik z filmem. W naszym przykładzie wstawiliśmy film z kanału YouTube **Potyczki Algorytmiczne** *https://youtu.be/KcmPjzGapAw*. Własny film możesz wysłać bezpośrednio na serwer strony.



Rys. 37.9. Przykład umieszczania filmu we wpisie strony wykonanej w CMS WordPress

Treści, które dodajesz do strony, mogą pochodzić z wielu portali społecznościowych, stron itp. Zapoznaj się z możliwościami ich wprowadzania. Sprawdź, jak działa wklejanie postów z twojego portalu społecznościowego, np. Facebooka.

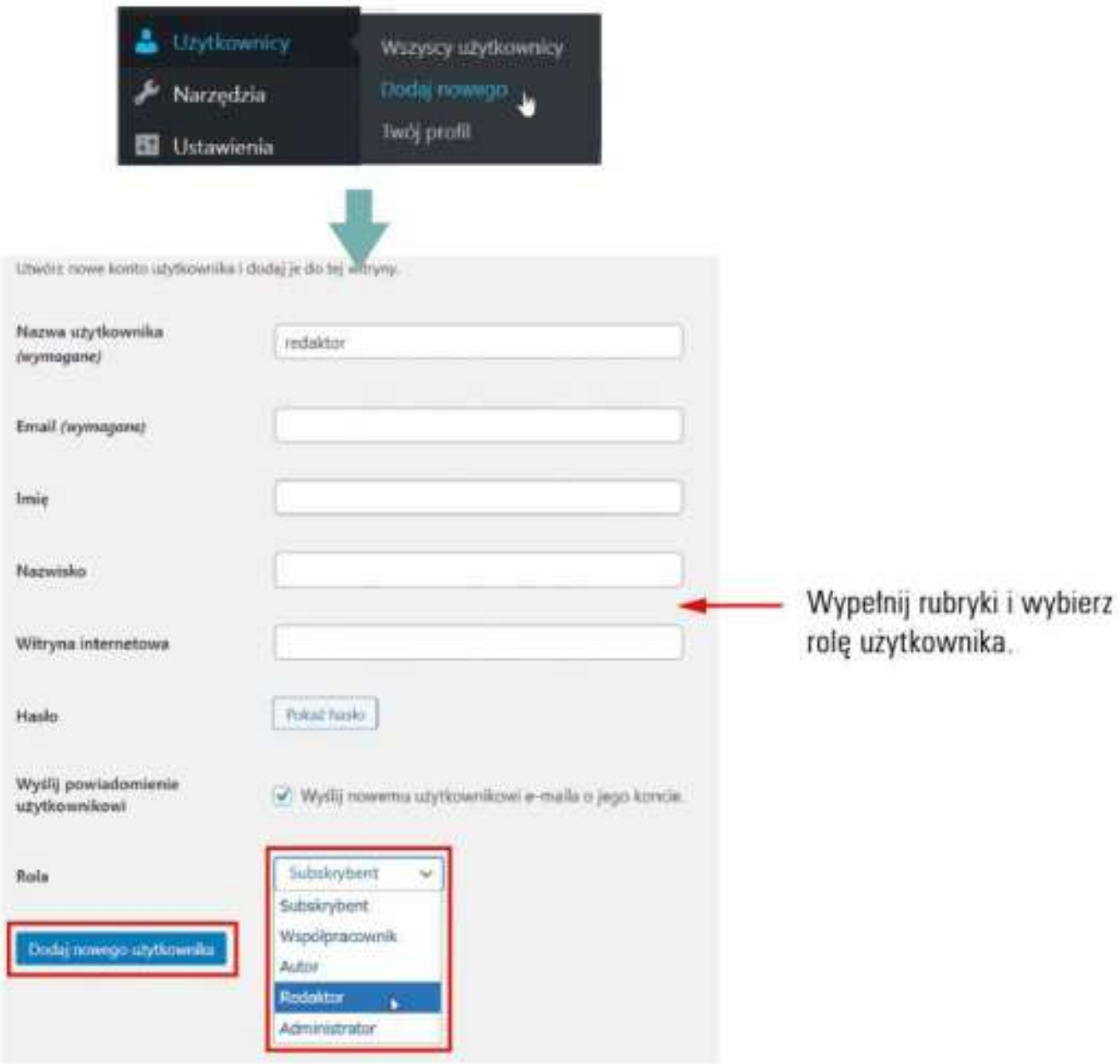
37.4. Więcej redaktorów, czyli nadajemy uprawnienia do zarządzania w WordPress

Portal informacyjny, strona szkoły i inne publikacje są prowadzone bardzo często przez zespół redakcyjny. CMS pozwala przydzielać uprawnienia poszczególnym

użytkownikom. Może to być np. zgoda na samodzielną edycję bloku informacyjnego (**Redaktor**) lub zarządzanie motywem (**Administrator**). Należy je przyznawać w sposób przemyślany i z uwzględnieniem umiejętności poszczególnych członków zespołu. Oczywiście jako właściciel i założyciel strony masz wszystkie możliwe uprawnienia przyznane domyślnie.

Dodajmy użytkownika do zespołu zarządzającego treściami na naszej stronie (rys. 37.10.).

dodawanie użytkownika



Rys. 37.10. Przykład dodawania nowego użytkownika w CMS WordPress

W przykładzie nowemu użytkownikowi nadano rolę **Redaktora**. Do niego mogą być przesyłane materiały (artykuły, zdjęcia itp.) przygotowane i zredagowane przez autorów lub współpracowników. Ukażą się dopiero po zaakceptowaniu przez **Redaktora**.

Łatwo zauważyć, że WordPress jest rozbudowany i nie ogranicza autorów stron w ich twórczym działaniu. Warto wypróbować wszystkie jego możliwości i sprawdzić ich wpływ na działanie i wygląd strony. Pamiętaj także o tym, że WordPress – podobnie jak inne systemy zarządzania treścią – jest systematycznie ulepszany i przystosowywany do nowych wymagań sieci. Twoja aktualna wersja może się nieznacznie różnić od opisanej w rozdziale. Niezmienne pozostaną jednak zasady tworzenia stron.

ZADANIA DO WYKONANIA

1. Utwórz prezentację przedstawiającą role, które mogą być przypisane poszczególnym użytkownikom CMS WordPress. Podaj przykłady zasięgu ich uprawnień.
2. Korzystając z emulatora Laragon i przykładowej strony, zmodyfikuj wygląd skórki, zmieniając grafikę jej nagłówka, tytuł strony i logo. Utwórz tutorial (samouczek) opisujący te czynności.
3. Korzystając z emulatora Laragon i przykładowej strony, umieść na stronie krótką recenzję wybranego artykułu z dowolnego, wiarygodnego portalu informacyjnego. Wstaw do swojego tekstu odnośnik kierujący do tekstu źródłowego.
4. Korzystając z emulatora Laragon i przykładowej strony, utwórz nowy wpis zawierający ilustracje. Omów sposób tworzenia takich wpisów.
5. Dodaj do przykładowej strony dwóch użytkowników. Jednemu nadaj rolę **Redaktora**, drugiemu **Autora**. Sprawdź różnice pomiędzy obiema rolami. Opisz je w krótkim dokumencie tekstowym.

PODSUMOWANIE LEKCJI

s. 236 ▸ • **Skórka**, inaczej zwana także **szablonem**, może być modyfikowana przez użytkowników. Możesz użyć darmowych szablonów z kolekcji *wordpress.org* lub dostępnych w sieci pod warunkiem, że są udostępnione bez opłat.

s. 238 ▸ • Dostosowanie skórki do treści twojej strony polega na modyfikacji z użyciem narzędzi Kokpitu, czyli panelu zarządzania stroną.

s. 240 ▸ • **Wpisy** umieszczane na stronie mogą zawierać informacje w różnych formach np. zdjęć, filmów itp. Wpisy ukazują się na stronie zgodnie z projektem skórki.

s. 243 ▸ • **Administrator** może dodawać nowych użytkowników i nadawać im role (uprawnienia).

38. Szybko i łatwo, czyli programy do tworzenia stron internetowych

NA TEJ LEKCJI:

- wyeksportujesz dokumenty różnych edytorów do formatu HTML;
- poznasz edytory blogów.

Ułożyłeś/Ułożyłaś kilka stron w HTML. Wiesz także, na czym polega używanie arkuszy stylów CSS. Poznałeś/Poznałaś zasady tworzenia stron WWW za pomocą systemu zarządzania treścią. Zadajesz sobie sprawę, że istnieją także inne sposoby tworzenia stron lub ich elementów. Poznaj mało znane opcje eksportu dokumentów i systemy gotowych blogów, które wystarczy jedynie wypełnić treścią.

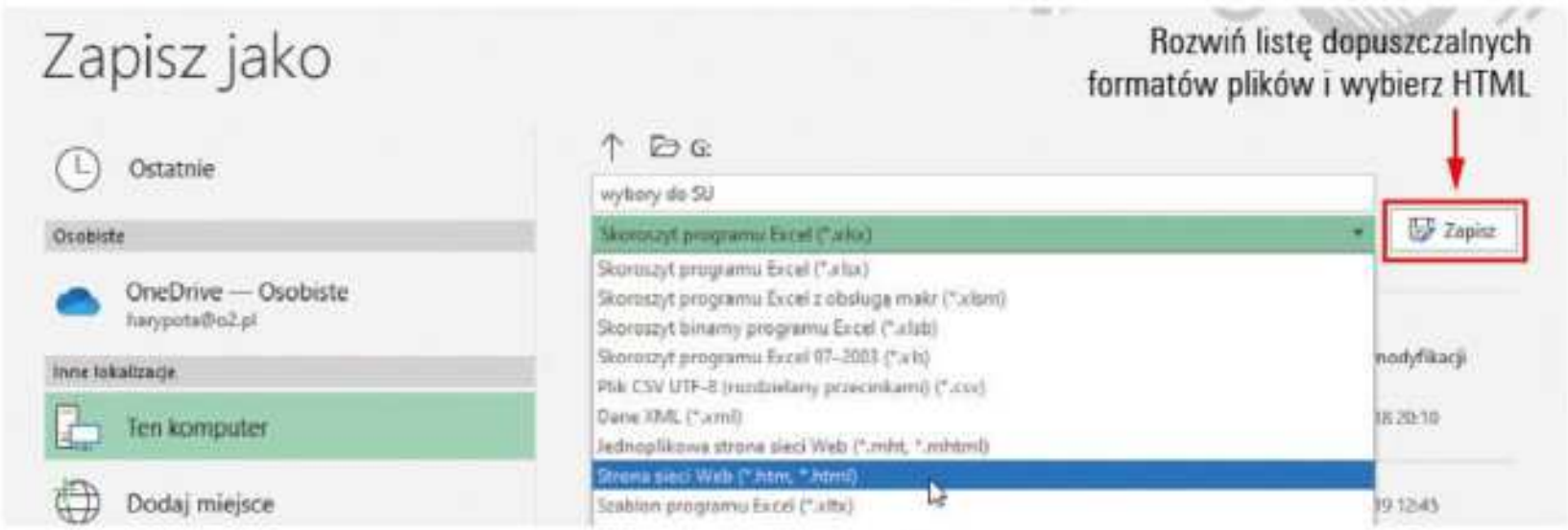
38.1. Zapisz jako, czyli jak eksportować dokumenty z arkusza do formatu HTML

Załóżmy, że na stronie internetowej chcesz umieścić wyniki szkolnych wyborów do samorządu uczniowskiego. Policzono głosy i zestawienie umieszczono w arkuszu kalkulacyjnym (rys. 38.1.). Wykonano wykresy i analizy. Ogłoszenie wyników postanowiono umieścić na stronie szkoły. Najprostszym sposobem byłoby umieszczenie zrzutu ekranowego okna arkusza, jednak nie jest to dobre rozwiązanie. Po pierwsze, należałoby dopasować rozdzielczość i wielkość zrzutu, a także zapewnić odpowiednie skalowanie wyświetlania ilustracji. Po drugie, uniemożliwia to skopiowanie danych ze strony przez redaktorów szkolnej gazety i innych zainteresowanych wykorzystaniem danych w swoich opracowaniach. Na szczęście arkusz ma opcję zapisania dokumentu w formacie HTML. Taki kod można wyświetlić jako podstronę lub wkleić do kodu artykułu strony.



Rys. 38.1. Przykładowy arkusz z wynikami wyborów

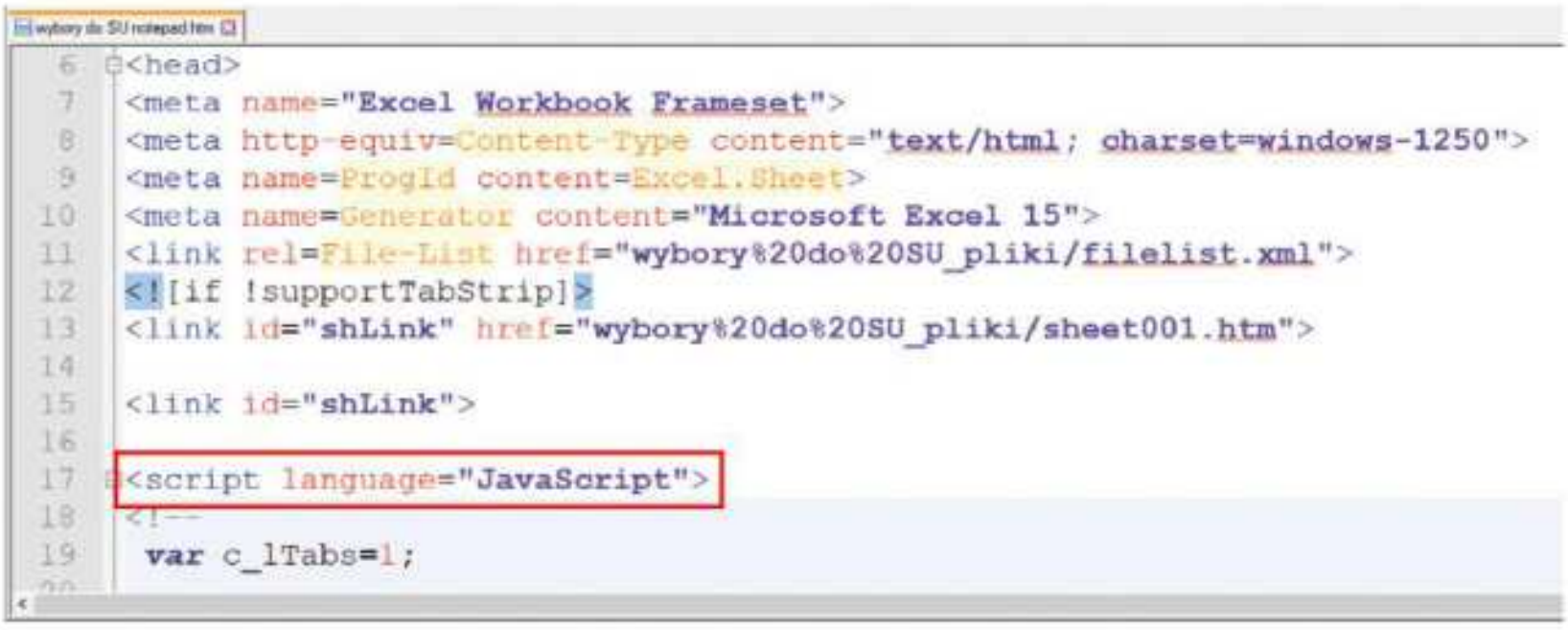
eksport arkusza do formatu HTML ▸ Gdy arkusz jest już gotowy, możesz go zapisać w formacie HTML (rys. 38.2.).



Rys. 38.2. Zapisywanie arkusza w formacie HTML

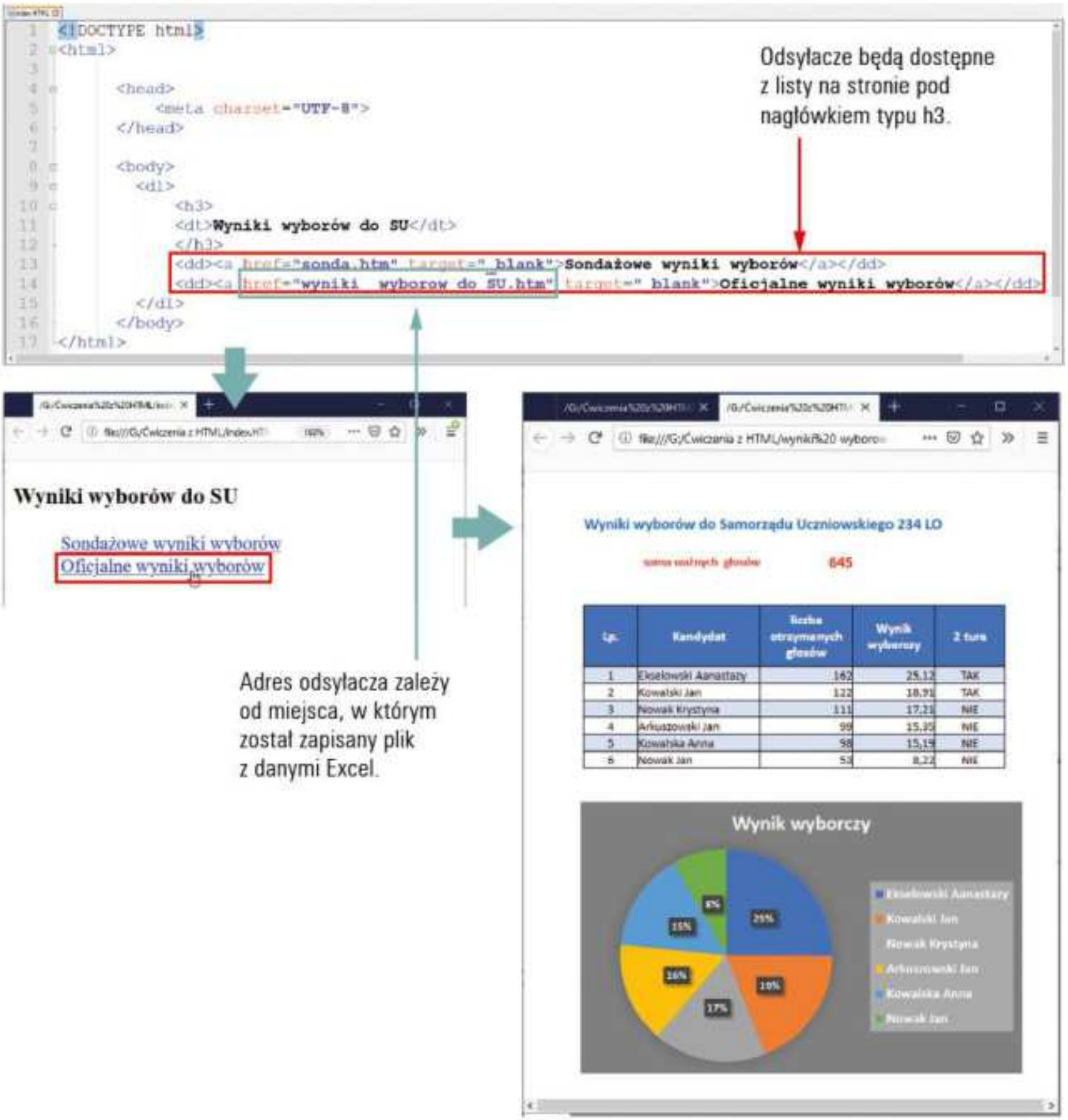
Teraz możesz sprawdzić, jak Excel poradził sobie z eksportem arkusza. Kliknij dwukrotnie ikonę powstałego pliku HTML. Powinien on się otworzyć w domyślnej przeglądarce. Otrzymana tabela daje się w prosty sposób skopiować i przenieść, np. do edytora tekstu. Wykres został wstawiony jako ilustracja. Można go skopiować za pomocą opcji **Kopiuj obraz** z menu kontekstowego (prawy przycisk myszy na rysunku).

Zauważ, że używając edytora, np. Notepad++, możesz podejrzeć kod wygenerowanego pliku HTML. Już pobieżna analiza wykaże, że zastosowano w nim skrypty w języku JavaScript (rys. 38.3.).



Rys. 38.3. Fragment kodu HTML z Excel

Załóżmy, że ze strony głównej do wyników wyborów mamy się dostać po kliknięciu odsyłacza. W takim przypadku należy umieścić plik HTML (wygenerowany przez Excel) w określonym folderze (rys. 38.4.). Dla uproszczenia ćwiczenia zapisano go w tym samym folderze co plik strony głównej **index.html**. Wykorzystano znany ci już z poprzednich rozdziałów kod z nagłówkiem i listą.



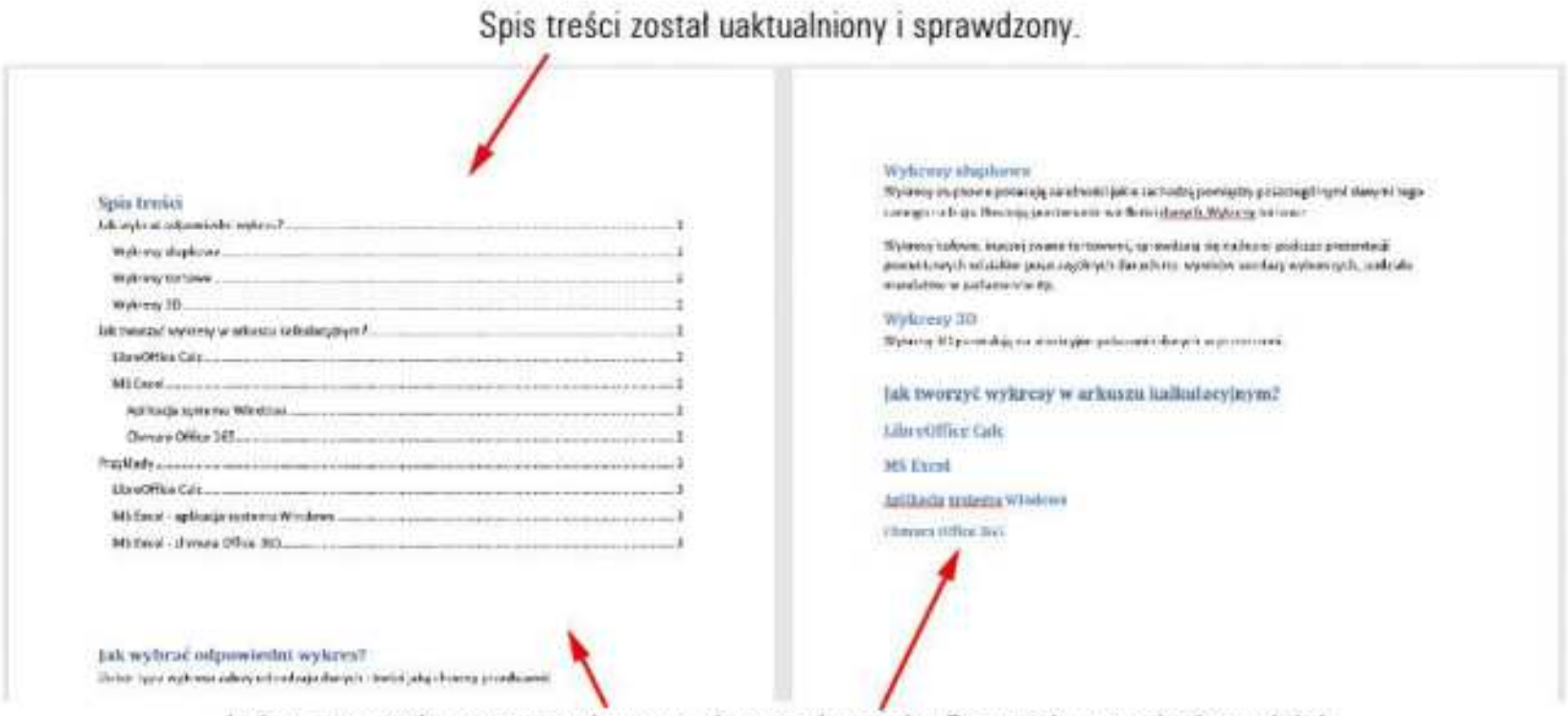
Lp.	Kandydat	Liczba otrzymanych głosów	Wynik wyborczy	2 tura
1	Ekselowski Aanastazy	162	25,12	TAK
2	Kowalski Jan	122	18,91	TAK
3	Nowak Krystyna	111	17,21	NIE
4	Arkuszowski Jan	99	15,35	NIE
5	Kowalska Anna	98	15,19	NIE
6	Nowak Jan	53	8,22	NIE

Rys. 38.4. Przykład odsyłacza do strony z danymi z Excel

38.2. Strona z edytora tekstowego, czyli jak zapisać dokument Word w formacie HTML

Domyślasz się, że schemat postępowania przypadku dokumentu Word będzie bardzo podobny do zapisywania arkusza Excel. Czy dokument HTML utworzony z dokumentu Word będzie wyłącznie sformatowanym tekstem wyświetlonym w przeglądarce? Sprawdźmy, czy zostaną zachowane odsyłacze spisu treści, spisu ilustracji i tabel. Wtedy działanie takiej strony byłoby podobne do list

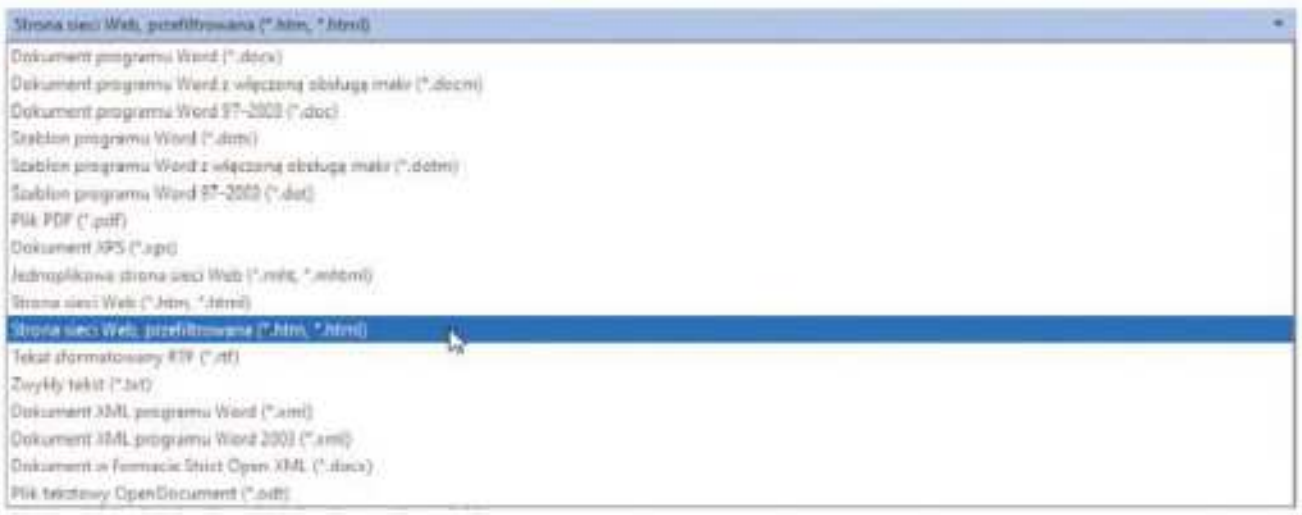
znanych ci z HTML. Będzie nam potrzebny dokument Word zawierający wymienione elementy. Możesz wykorzystać te, które zostały opisane w początkowych rozdziałach podręcznika (rozd. 2. i 3.; rys. 38.5.).



Rys. 38.5. Przykładowy dokument tekstowy edytora Word ze spisem treści

eksport z edytora tekstu do HTML

Zapiszmy dokument w formacie HTML. Tym razem masz do wyboru trzy opcje. Wybierz **Strona sieci Web, przefiltrowana (*.htm,*.html)**. Pomoc na stronach Microsoftu taki wybór argumentuje następująco: *Po zapisaniu dokumentu jako przefiltrowanej strony sieci Web program Word zachowuje tylko zawartość, instrukcje dotyczące stylu oraz inne informacje. Plik jest mały, bez dużego kodu.* Zastosujmy się do tej rady (rys. 38.6.).

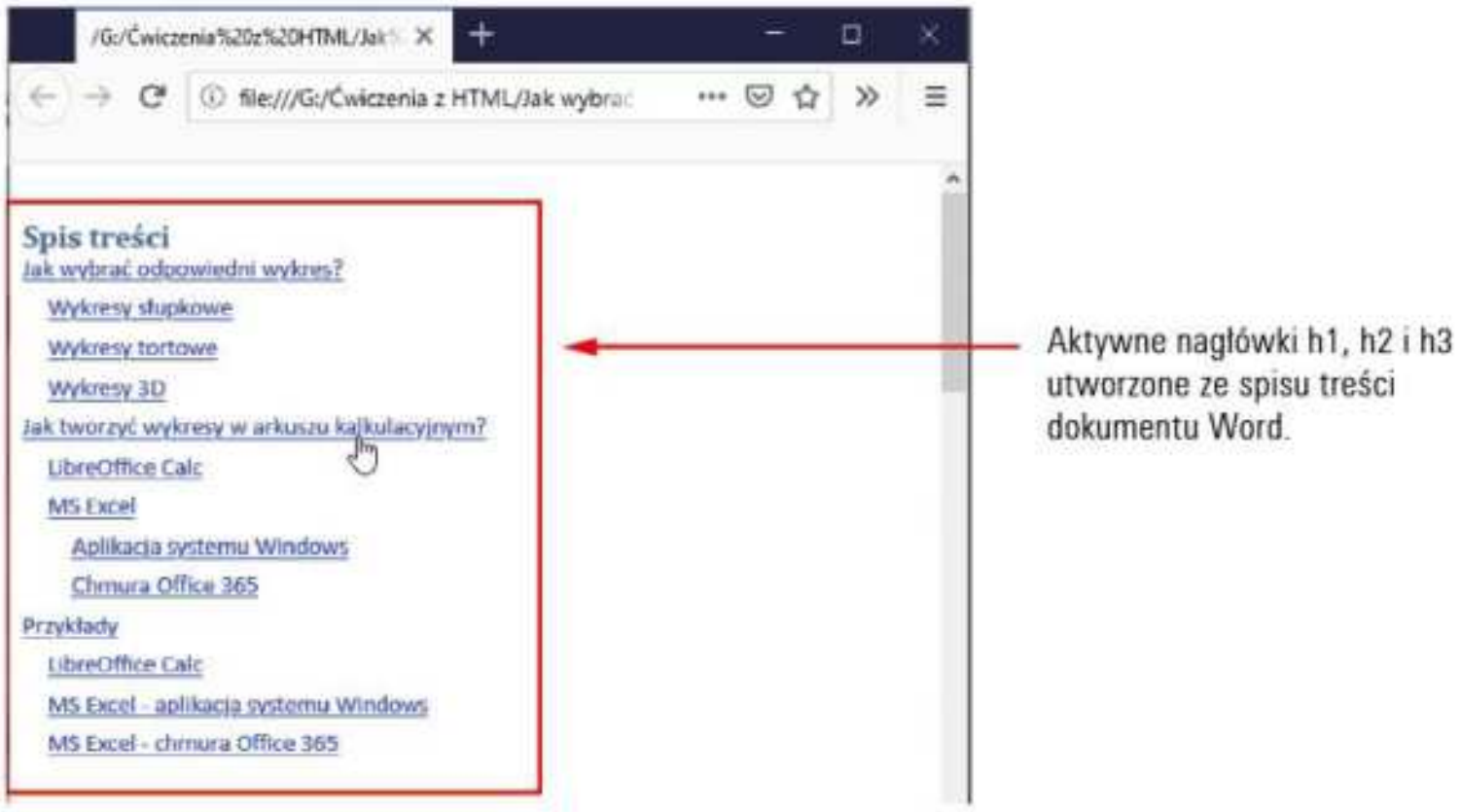


Rys. 38.6. Zapisywanie dokumentu Word w formacie HTML

Tym razem kod będzie dużo krótszy od tego otrzymanego z Excela. Jeśli wyedytujesz go za pomocą Notepad++ i przeanalizujesz, znajdziesz tam znane ci już znaczniki, np. spis treści został zapisany z użyciem znaczników nagłówków `h1`, `h2` i `h3`. Użyto także definicji klas.

Gdy plik zostanie już zapisany, uruchom go w przeglądarce i sprawdź, czy działają odsyłacze do poszczególnych rozdziałów. Podobnie jak w przypadku pliku

uzyskanego z arkusza, możesz użyć kodu z Worda jako osobnej strony lub podstrony (rys. 38.7.).



Rys. 38.7. Uruchomiony w przeglądarce przykładowy plik HTML uzyskany z dokumentu Word

Nasz przykład zawierał jedynie sformatowany tekst. W dokumencie Word mogą się znajdować odsyłacze, ilustracje, tabele, wzory itp. Czy i tym razem program tłumaczący dokument na kod HTML wykona wszystko prawidłowo? Utwórzmy dokument Word, w którym zawrzemy powyższe elementy (rys. 38.8.). Zapiszmy go w formacie HTML tak jak poprzedni (ze spisem treści).

To jest dokument

http://men.gov.pl

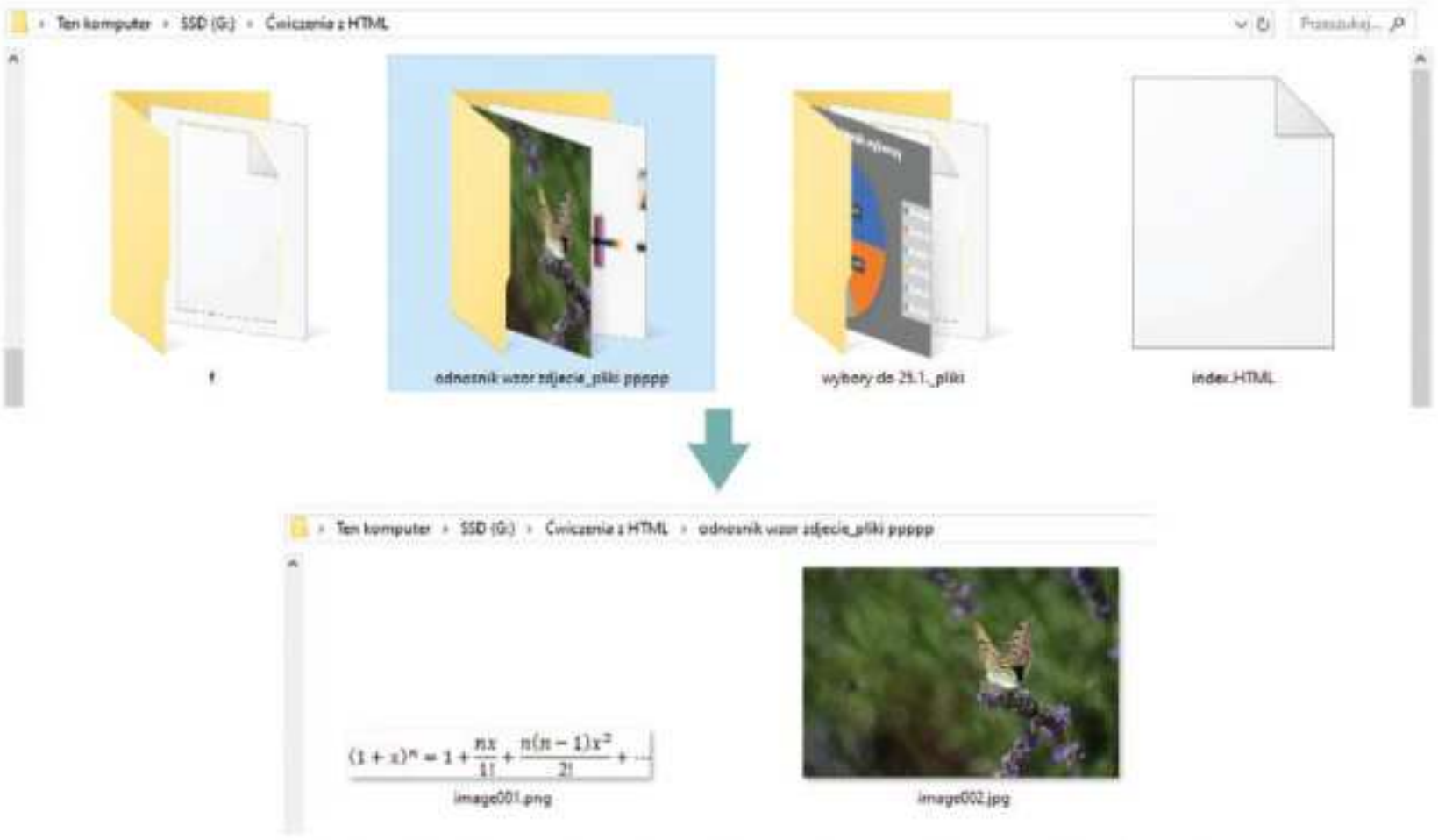
kolumna 1	kolumna 2	kolumna 3	kolumna 4	kolumna 5

$$(1+x)^n = 1 + \frac{nx}{1!} + \frac{n(n-1)x^2}{2!} + \cdots$$



Rys. 38.8. Przykładowy dokument zawierający kilka różnych elementów

Zapis przebiegł prawidłowo. Powstał plik o założonej nazwie. Jeśli jednak przypomnisz sobie wiadomości z rozdziałów o tworzeniu programów HTML z wstawianiem zdjęć i innych obiektów, dojdziesz do wniosku, że i w tym przypadku muszą być one zapisane w osobnych plikach. Przeglądając folder docelowy, znajdziesz folder o takiej samej nazwie jak plik HTML z dopiskiem **_pliki**. Domyślasz się, że tam właśnie trafiły wyodrębnione z dokumentu Word elementy – zdjęcie i wzór (rys. 38.9.). Wynika z tego, że tekst, tabela i odnośnik zostały zaprogramowane w kodzie strony. Aby strona prawidłowo wyświetlała elementy z folderu, musisz zadbać, by folder z elementami był w tej samej lokalizacji co jej plik HTML, a nazwy plików w nim zapisanych pozostały z takimi samymi nazwami.



Rys. 38.9. Folder z elementami strony utworzonej przez edytor Word

38.3. Blog, czyli nie zawsze trzeba tworzyć stronę od podstaw

strona jako blog ▸ Jeśli twoja strona ma być blogiem lub chcesz planowane treści (informacje z życia twojej miejscowości, galerie zdjęć itp.) umieszczać w formie bloga, nie musisz korzystać z CMS ani tym bardziej pisać programu strony od podstaw. Wykorzystaj usługi internetowe oferujące gotowe systemy blogów.

Jednym z przykładów jest *blogger.com*. Jest to jeden z programów dostępnych w chmurze Google. Jeśli masz konto w Gmail, YouTube lub innych usługach tej firmy, nie musisz nawet zakładać konta. Podobnie jak w przypadku CMS, możesz wybierać motywy, umieszczać na stronie dowolne elementy – zdjęcia, filmy, teksty itp.



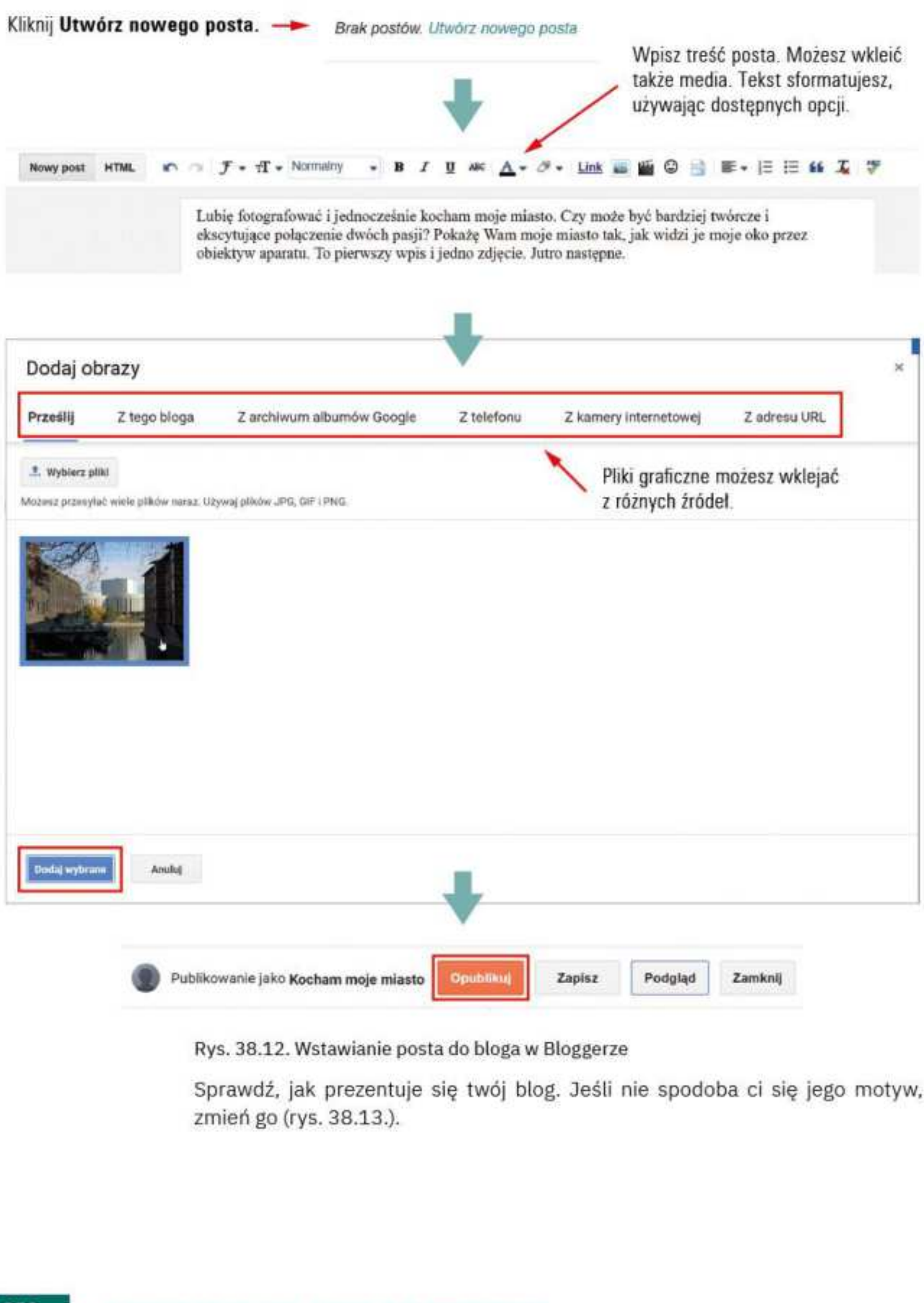
Rys. 38.10. Zakładanie konta w Bloggerze

Zapewne w trakcie zakładania konta brakowało ci możliwości wyboru motywu. A przecież to jeden z ważniejszych elementów strony. Zrobisz to z poziomu panelu użytkownika (rys. 38.11.).



Rys. 38.11. Tworzenie nowego bloga w Bloggerze

Czas na pierwszy wpis. W panelu użytkownika użyjesz łatwego w obsłudze edytora (rys. 38.12.).



Rys. 38.12. Wstawianie posta do bloga w Bloggerze

Sprawdź, jak prezentuje się twój blog. Jeśli nie spodoba ci się jego motyw, zmień go (rys. 38.13.).

W panelu użytkownika wybierz **Motyw** i spośród dostępnych skórek wybierz odpowiednią dla twojego bloga.



Rys. 38.13. Zmiana motywu w Bloggerze

Blogger jest jedną z wielu usług (programów sieciowych), które pomagają w tworzeniu blogów lub stron internetowych Z pewnością znajdziesz odpowiedni dla swoich stron. Pamiętaj jednak, że jeśli prowadzisz bloga, nie możesz zaniedbywać swoich czytelników. Swoje posty umieszczaj regularnie i staraj się, by były ciekawe. Panele użytkownika zazwyczaj mają wiele narzędzi do sprawdzania liczby odwiedzin.

wada darmowych blogów

Pewną **wadą** blogów tworzonych w darmowych usługach jest konieczność korzystania z domeny usługodawcy. Otrzymujemy za to możliwość dotarcia do wielu czytelników dzięki systemowi promocji i powiadomień.

ZADANIA DO WYKONANIA

1. Utwórz dokument Word z jedną ilustracją. Zapisz go w formacie HTML. Odnajdź folder, do którego w procesie zapisu trafiła ilustracja. Zmień nazwę pliku z ilustracją. Uruchom plik HTML w przeglądarce. Omów przyczynę zmiany na stronie.
2. Utwórz dokument Word zawierający ilustracje i ich spis. Zapisz plik w formacie HTML i uruchom w przeglądarce. Sprawdź, czy spis będzie się składał z odnośników do ilustracji oraz czy zapisanie pliku w HTML spowodowało także powstanie folderu z wyodrębnionymi ilustracjami.
3. Utwórz dokument Word zawierający tabele i ich spis. Zapisz plik w formacie HTML i uruchom w przeglądarce. Sprawdź, czy spis będzie się składał z odnośników do tabel oraz czy zapisanie pliku w HTML spowodowało także powstanie folderu z tabelami. Jeśli nie, wyjaśnij przyczynę.

4*. Edytuj plik HTML z zadania 1. Wskaż miejsce, w którym program odwołuje się do ilustracji. Przedstaw działanie tego fragmentu kodu.

5. Znajdź w internecie usługi oferujące zakładanie blogów lub stron internetowych bez konieczności instalowania CMS i znajomości języków programowania. Porównaj ich możliwości i łatwość obsługi. Wybierz najlepszy według ciebie. Uzasadnij swój wybór.

PODSUMOWANIE LEKCJI

s. 246 • **Arkusze kalkulacyjne** mają możliwość eksportu do formatu HTML, lecz na stronie nie będą aktywne.

s. 248 • **Edytory tekstów**, w tym także MS Word, umożliwiają zapisanie dokumentu w formacie strony internetowej (HTML). Mogą przy tym zachowywać formatowanie, układ elementów, grafikę, tabele, odnośniki, spisy treści itp.

s. 250 • Do budowy prostych stron internetowych można skorzystać z usług tworzenia **blogów** internetowych. Dzięki nim utworzysz stronę bez znajomości języków programowania lub CMS.

s. 253 • **Wadą blogów** tworzonych w darmowych usługach jest konieczność korzystania z domeny usługodawcy. Wpisy ukazują się na stronie zgodnie z projektem skórki.

39. Pierwsze skrypty, czyli poznajemy język PHP

NA TEJ LEKCJI:

- dowiesz się, jakim rodzajem języka jest język PHP;
- nauczysz się umieszczać skrypt PHP w dokumencie HTML;
- poznasz sposób wstawiania komentarzy do kodu PHP;
- dowiesz się, jak stworzyć pierwszy skrypt w języku PHP;
- nauczysz się umieszczać plik .php na lokalnym serwerze.

◂ PHP

PHP jest językiem programowania, który ma cechy języków Pearl i C. Jest to język interpretowalny, czyli nie ma potrzeby kompilacji skryptów czy programów w nim napisanych przed ich uruchomieniem, ponieważ wszystkie operacje wykonywane są przez serwer WWW. Gdy serwer WWW otrzyma dokument i żądanie rozpoznania go jako skryptu (programu) PHP, przetwarza go przez interpreter PHP. Interpreter przygotowuje odpowiedni kod HTML, który jest przesyłany przez serwer WWW do urządzenia klienckiego.

39.1. Podstawy języka, czyli pierwszy skrypt PHP

◂ umieszczanie skryptu PHP w dokumencie HTML

Dokumenty tworzone przy użyciu języka PHP to tak naprawdę dokumenty składające się z kompilacji kodu HTML i skrypty PHP. W dokumencie takim skrypt PHP jest wydzielony specjalnym znacznikiem – może być również na odwrót: kod HTML jest zawarty w skrypcie PHP.

Początek i koniec skryptu PHP musi być zaznaczony przez parę znaczników:

```
<?php ... ?>
<? ... ?>
<% ... %>
<script language ="php"> ... </script>
```

Pierwszy i ostatni typ są dostępne standardowo, natomiast drugi i trzeci wymagają włączenia odpowiednich konfiguracji środowiska PHP – nie są one jednak zalecane, ponieważ nie spełniają standardów HTML i XHTML.

Przy tworzeniu skryptów PHP polecam korzystać z komentarzy. Ułatwiają one wszystkim osobom pracę nad danym plikiem.

Komentarze w skrypcie PHP umieszczamy za pomocą następujących symboli:

/* ... */ – komentarz wielowierszowy,

// – komentarz jednowierszowy – wszystko od tego znaku do końca wiersza będzie komentarzem,

– to również jest komentarz jednowierszowy – wszystko od tego znaku do końca wiersza będzie komentarzem.

Napiszmy pierwszy skrypt z użyciem języka PHP.

Przykład

Stwórzmy skrypt, który będzie wyświetlał nasze imię i nazwisko (rys. 39.1.).

40_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Poznajmy się</title>

</head>
<body>
    <p>
        <?php
            //Wypisuję imię i nazwisko za pomocą funkcji "print"
            print ("Dzień dobry, nazywam się Jan Kowalski.");
        ?>
    </p>
</body>
</html>
```

Rys. 39.1. Kod HTML z wbudowanym skryptem PHP

print(), echo(), printif()

W skrypcie PHP użyliśmy funkcji `print()` w celu wypisania tekstu. Możemy użyć także funkcji `echo()` i `printif()`:

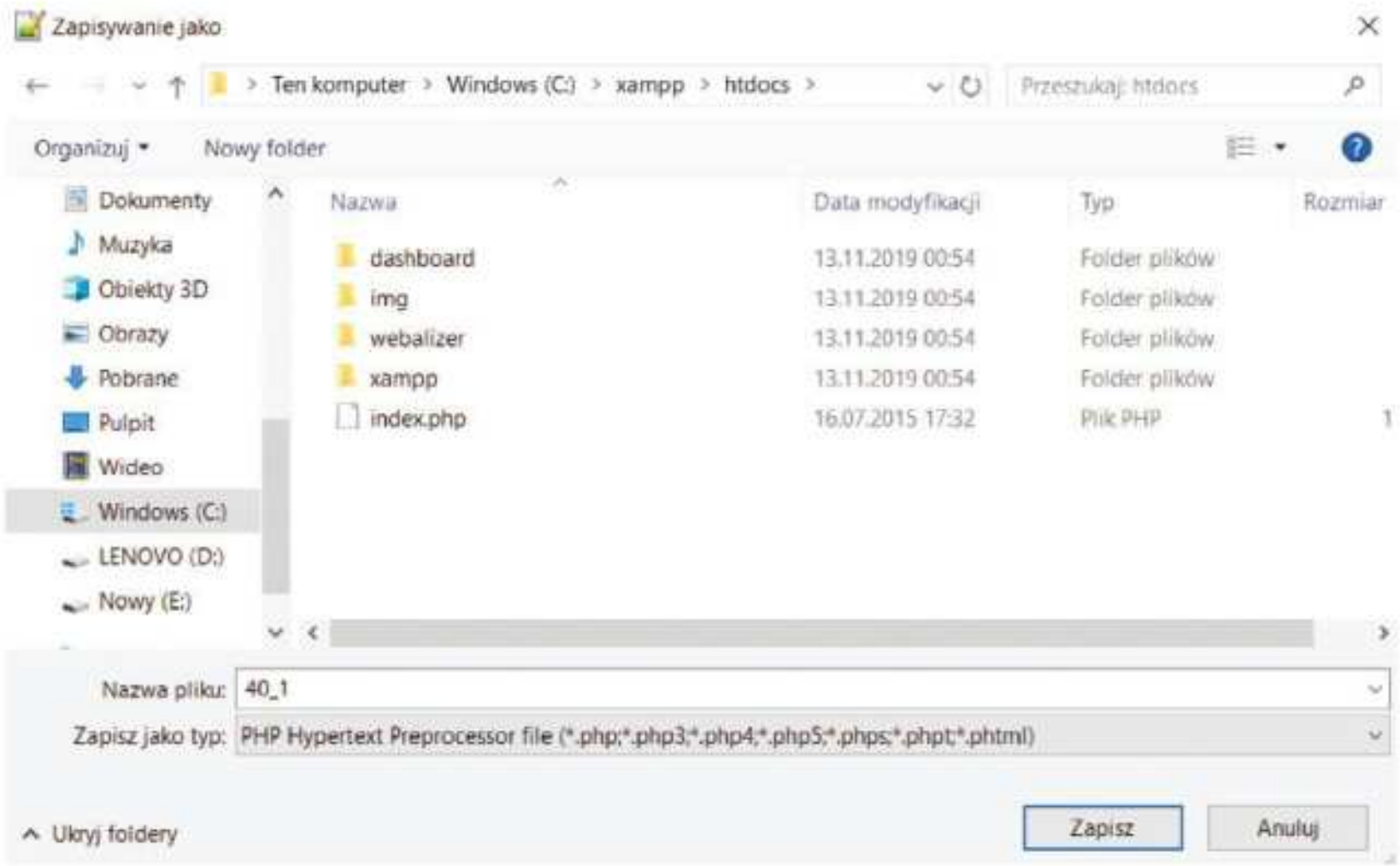
- `print()` – jest najprostszą funkcją wypisującą podaną wartość,
- `echo()` – wypisuje podaną wartość podobnie jak print,
- `printif()` – pozwala na wypisanie wartości po jej wcześniejszym sformatowaniu.

Instrukcje PHP są zawsze zakończone znakiem średnika `(;)`.

Po skończeniu naszego skryptu zapisujemy go na dysku z rozszerzeniem .php. Skrypty i programy pisane w języku PHP możemy zapisać również z rozszerzeniami: .php3, .php4, .php5 oraz .phtml.

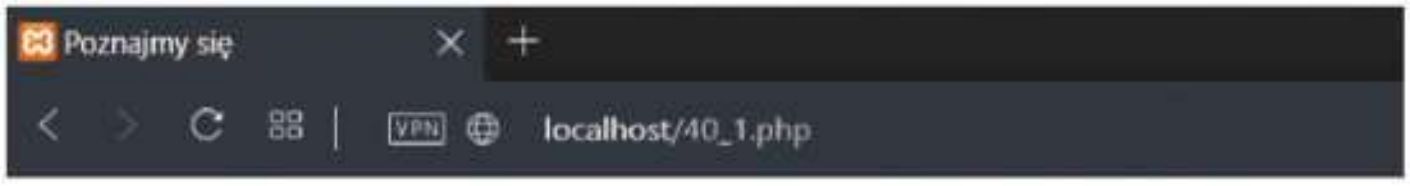
Aby otworzyć nasz plik w przeglądarce, musimy go umieścić na serwerze – albo na serwerze WWW w internecie, albo na lokalnym. Ponieważ korzystamy z lokalnego serwera Apache, więc na nim umieścimy nasz plik. Nazwijmy go **40_1.php.**

Należy odszukać na dysku folder **XAMPP**, a następnie wyszukać w nim folder **hcdocs** i to w nim umieścić plik (rys. 39.2).



Rys. 39.2. Umieszczanie pliku 40_1.php na lokalnym serwerze

Teraz w oknie przeglądarki wpisujemy: *http://localhost/40_1.php*. Powinniśmy zobaczyć efekt naszej pracy, nasz pierwszy skrypt napisany w PHP (rys. 39.3.).



Rys. 39.3. Wynik skryptu napisanego w języku PHP

ZADANIA DO WYKONANIA

1. Wyszukaj w internecie informacje na temat różnic pomiędzy funkcjami `echo()` i `print()`. Sporządź w zeszycie notatkę na ten temat.
2. Utwórz stronę internetową zawierającą skrypt w języku PHP, która będzie twoją wizytówką. Strona powinna zawierać twoje imię i nazwisko, adres zamieszkania, numer telefonu i adres e-mail.
3. Utwórz stronę internetową zawierającą skrypt w języku PHP, która będzie wizytówką twojej szkoły. Strona powinna zawierać nazwę szkoły oraz jej adres, numer telefonu i adres e-mail. W skrypcie umieść komentarze jednowierszowe, w których wyjaśnisz, jakie zadania pełnią poszczególne wiersze kodu skryptu.

PODSUMOWANIE LEKCJI

s. 255 ▸ • PHP jest interpretowalnym językiem programowania, który ma cechy języków Pearl i C. Oznacza to, że nie ma potrzeby kompilacji skryptów czy programów w nim napisanych przed ich uruchomieniem. Są one wykonywane przez serwer WWW.

s. 255 ▸ • Początek i koniec skryptu PHP musi być zaznaczony przez parę znaczników `<?php ... ?>`. Możemy także zastosować znaczniki `<?..?>` lub `<%...%>`, ale ich używanie wymaga zmian w konfiguracji środowiska PHP.

s. 256 ▸ • W skrypcie PHP używamy funkcji `print` w celu wypisania tekstu. Możemy użyć także funkcji `echo` i `printif`.

40. Stałe i zmienne, czyli typy danych w języku PHP

NA TEJ LEKCJI:

- dowiesz się, czym są zmienne i stałe w języku PHP;
- nauczysz się, z jakich typów danych korzystamy w języku PHP;
- poznasz operatory stosowane w języku PHP.

Jak w każdym języku programowania, również w PHP spotkamy się ze zmiennymi, stałymi oraz operatorami. Czym zatem są te elementy i do czego służą podczas tworzenia skryptów?

40.1. Zmienne w PHP

zmienne ▸ Jak każdy język programowania, tak samo język PHP posługuje się **zmiennymi**. Najprościej je opisać jako pudełka, które przechowują dane w trakcie wykonywania programu. Każda zmienna ma swoją unikatową nazwę, poprzez którą odwołujemy się do jej wartości.

Nazwy zmiennych mogą składać się z liter, cyfr i znaku podkreślenia, nie mogą jednak zaczynać się od cyfry. Każda nazwa zmiennej jest poprzedzona znakiem dolara – np. `$zmienna`.

Dobrym nawykiem jest nazywanie zmiennych jednolicie i adekwatnie. Ułatwi nam to pracę i przegląd programu, jeżeli wrócimy do niego po jakimś czasie.

Również podczas jego pisania będzie nam prościej, jeżeli będziemy używać nazw zmiennych, które informują nas, do czego dana zmienna się odnosi.

typy zmiennych

Język PHP nie wymaga od nas jawnej deklaracji danych i pozwala na zmianę ich typu podczas pisania programu. W języku PHP typ zmiennej dobierany jest automatycznie. Zależy on od rodzaju danej, jaką ma przechowywać. Jeśli zmienna przechowywała liczbę całkowitą, a w pewnym momencie zapiszemy pod jej nazwą tekst, wówczas typ zmiennej zmieni się samoczynnie.

Przykład

Utworzymy wizytówkę wyświetlającą imię i nazwisko, wykorzystując przy tym zmienne (rys. 40.1.).

41_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Poznajmy się drugi raz</title>

</head>
<body>
    <div>
    <p>
        <?php
            //Imię i nazwisko zapisujemy w dwóch odrębnych zmiennych
            $imie = "Jan";
            $nazwisko = "Kowalski";
            //Za pomocą funkcji print() wypisuję imię i nazwisko
            print ("Dzień dobry, nazywam się " . $imie . " " . $nazwisko. ".");
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 40.1. Skrypt PHP z użyciem zmiennych

Efekt tego skryptu jest identyczny jak ten, który napisaliśmy w poprzednim rozdziale, jednak tym razem imię i nazwisko nie zostało wypisane bezpośrednio za pomocą funkcji `print()`, lecz użyliśmy do tego wcześniej zadeklarowanych zmiennych. Przypisanie wartości do zmiennej dokonuje się za pomocą znaku równości, a kolejne linie kodu zostały oddzielone od siebie średnikiem.

interpolacja zmiennych

Łańcuch tekstowy w kodzie PHP może być zapisany między znakami cudzysłowu lub między apostrofami. Ich znaczenie jest różne. W łańcuchach zapisanych w cudzysłowach następuje tak zwana **interpolacja zmiennych**. Oznacza to, że nazwy zmiennych są zastępowane przez ich wartości. W łańcuchach, które są zapisane w apostrofach, taka zamiana nie następuje.

Przykład

```
$tekst = "Nazywam się $imie $nazwisko";
wyświetli nam: Nazywam się Jan Kowalski
$tekst = 'Nazywam się $imie $nazwisko';
wyświetli nam: Nazywam się $imie $nazwisko.
```

W 16. linii naszego kodu pomiędzy poszczególnymi łańcuchami pojawiły się kropki – są one operatorami łączącymi łańcuchy.

40.2. Definiowanie i użycie stałych

Zdarza nam się często, że w skrypcie musimy użyć danych, które mają stałą wartość, np. liczba ∏. Używając jej, będziemy mieli pewność, że jej wartość nie ulegnie zmianie. Takie dane nazywamy stałymi.

Stałe definiuje się za pomocą funkcji `define`:

```
define ("STAWKA_VAT" , 0.23);
```

Zaprezentowana funkcja posiada dwa argumenty – nazwę stałej i jej wartość. Nazwę stałej w języku PHP przyjęło się zapisywać dużymi literami – nie jest to oczywiście obowiązkowe.

Przykład

Obliczymy, ile zapłacimy za usługi, które są obłożone stawką podatkową 23%. Biuro rachunkowe chce wprowadzić konkurencyjne niższe ceny na rynku z tytułu świadczonych usług. W związku z tym usługę prowadzenia księgowości uproszczonej wycenili na 350,50 zł netto. W jakiej wysokości biuro musi zapłacić podatek VAT od swoich usług, jeżeli właściwą stawką VAT jest w tym przypadku 23%?

350,50 * 23% = 80,62

41_2.php

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <meta charset="utf-8" />
5      <title>Pole trójkąta</title>
6
7  </head>
8  <body>
9      <div>
10     <p>
11         <?php
12             define ("PODATEK_VAT" , 0.23);
13             print ("Podatek VAT za usługi wynosi: ");
14             printf ("%0.2f" , 350.50 * PODATEK_VAT);
15             print (' PLN');
16
17         ?>
18     </p>
19     </div>
20 </body>
21 </html>
22
```

Rys. 40.2. Skrypt PHP z użyciem stałych

W skrypcie użyliśmy funkcji `printf`, która pozwala na formatowanie wypisanego tekstu. Funkcja ta wymaga argumentu, który określa sposób formatowania. Po nim są wypisywane informacje do wyświetlenia oddzielone przecinkami.

W 15. wierszu naszego kodu określiliśmy format (%0.2f), który oznacza liczbę zmiennoprzecinkową (f). Jej część dziesiętna powinna zająć tyle znaków, ile jest niezbędnych do jej wyświetlenia (0), a część ułamkowa powinna zostać wyświetlona z dokładnością do dwóch (2) miejsc po przecinku.

40.3. Operatory

◂ operatory

Do wykonywania operacji matematycznych i logicznych na zmiennych i stałych będziemy się posługiwać **operatorami**. Większość operatorów w języku PHP jest całkowicie intuicyjna. Część z nich poznaliście, jeżeli programowaliście w języku Pearl (czyt. perl) lub C. Oto lista najważniejszych z nich:

`==` – operator równości, przy którym musimy użyć dwóch znaków równości, ponieważ jeden znak oznacza przypisanie,
`!=` – operator nierówności,
`!` – operator zaprzeczenia (negacji), który jest stosowany przed argumentem,
`++` – operator zwiększenia o 1, czyli inkrementacja,
`--` – operator zmniejszenia o 1, czyli dekrementacja.

Przypisanie jest również operatorem, który ma kilka pochodnych:

`+=` – przypisuje wynikowi sumę bieżącej wartości zmiennej docelowej i argumentu (`$x = $x + $y`),
`-=` – przypisuje wynikowi różnicę bieżącej wartości zmiennej docelowej i argumentu (`$x = $x - $y`),
`.=` – przypisuje wynikowi (będącemu łańcuchem) jego połączenie z łańcuchem (będącym argumentem).

Przykład

Utwórzmy skrypt, który wypisze dane teleadresowe szkoły (rys. 40.3.).

41_3.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Wizytówka szkoły</title>

</head>
<body>
    <div>
    <p>
        <?php
            //Kolejne informacje zostają zebrane przy użyciu operatora .=
            $tekst = 'Zespół Szkół Elektronicznych';
            $tekst .= '<br />';
            $tekst .= 'im. Wojska Polskiego';
            $tekst .= '<br />';
            $tekst .= 'w Bydgoszczy';
            $tekst .= '<br />';
            $tekst .= '85-092 Bydgoszcz';
            $tekst .= '<br />';
            $tekst .= 'ul. Karłowicza 20';
            $tekst .= '<br />';
            $tekst .= 'tel. +48523414444';
            print ($tekst);
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 40.3. Skrypt PHP z użyciem operatora .=

ZADANIA DO WYKONANIA

1. Napisz skrypt w języku PHP, który będzie obliczał obwód i pole trapezu. Użyj w nim zmiennych.
2. Napisz skrypt w języku PHP, który będzie obliczał pole i obwód koła. Zastosuj przynajmniej jedną stałą.
3. Napisz skrypt w języku PHP, który będzie wykonywał podstawowe działania arytmetyczne (dodawanie, odejmowanie, mnożenie i dzielenie). W skrypcie zadeklaruj dwie zmienne o wartościach –7 i 3.

PODSUMOWANIE LEKCJI

s. 258
- PHP posługuje się zmiennymi, które można opisać jako pudełka. Przechowują one dane w trakcie wykonywania programu. Każda zmienna ma swoją unikatową nazwę, poprzez którą odwołujemy się do jej wartości.

s. 259
- W języku PHP nie wymaga się deklaracji danych i pozwala na zmianę ich typu podczas pisania programu. Wyróżniamy następujące typy zmiennych: łańcuchy znakowe, liczby całkowite i liczby zmiennoprzecinkowe.

s. 261
- Do wykonywania operacji matematycznych i logicznych służą między innymi operatory arytmetyczne, przypisania, inkrementacji i porównania.

41. Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP

NA TEJ LEKCJI:

- dowiesz się, czym są instrukcje sterujące w języku PHP;
- dowiesz się, czym są instrukcje warunkowe w języku PHP;
- poznasz działanie instrukcji wyboru w języku PHP.

Instrukcje w języku PHP służą do sterowania kolejnością wykonywania poleceń, które mamy zapisane w skrypcie. Język PHP ma wiele instrukcji pozwalających na zmianę kolejności wykonywania poleceń w zależności od tego, czy zostaną spełnione określone warunki, czy też nie.

41.1. Instrukcja warunkowa

instrukcja warunkowa

Do tej pory w programach, które pisaliśmy, instrukcje były wykonywane jedna po drugiej. Nie przeszkadzało nam to, ponieważ programy były bardzo proste. Podczas pisania bardziej skomplikowanych skryptów będziemy jednak musieli zaprogramować różnego rodzaju działania w taki sposób, że będą one wykonywane w zależności od konkretnego warunku. W języku PHP taką możliwość daje **instrukcja warunkowa**. Działa ona w ten sposób, że sprawdza wartość warunku. Jeżeli jest on prawdziwy, jest wykonywany blok instrukcji, który występuje po nim. Jeżeli warunek nie jest prawdziwy, program przechodzi do sprawdzania wartości kolejnego warunku. Zapoznaj się z przykładem przedstawionym na rysunku 41.1.

42_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Która liczba jest większa?</title>

</head>
<body>
    <div>
    <p>
        <?php
            // W instrukcji warunkowej sprawdzasz czy $a jest większe od $b
            $a = 7;
            $b = 10;
            print ("<b>a</b> ma wartosc $a<br />");
            print ("<b>b</b> ma wartość $b<br />");
            if ($a>$b){
                print ("Większą liczbą jest a, która ma wartość $a. ");
            }
                elseif ($a==$b){
                print ("Liczby a i b są równe, obie mają wartość $a. ");
            } else {
                print ("Większą liczba jest b, która ma wartość $b. ");
            }
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 41.1. Skrypt PHP napisany z użyciem instrukcji warunkowej

Skrypt podany w przykładzie ma sprawdzić, która spośród dwóch podanych liczb jest większa, a następnie to wypisać, podając jej wartość.

41.2. Instrukcja wyboru

instrukcja wyboru

Jeżeli chcemy sprawdzić wartość zmiennej i w zależności od wyniku wykonać działanie, w skrypcie wykorzystamy **instrukcję wyboru**. Umożliwi ona rozdzielenie działania programu w zależności od wartości wyrażenia.

Wyrażenia wyboru służą do tego, aby określić, które z działań ma zostać podjęte (to, dla którego warunek odpowiada wartości). Jeżeli nie jest on spełniony dla żadnej wartości, brane są pod uwagę wartości z bloku `default`.

Zapoznaj się z przykładem (rys. 41.2.):

42_2.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Która liczba jest większa?</title>

</head>
<body>
    <div>
    <p>
        <?php
            $dzien = date("d");
            $miesiac = date("m");
            $rok = date("Y");
            switch ($miesiac){
                case '01': $miesiac = 'stycznia'; break;
                case '02': $miesiac = 'lutego'; break;
                case '03': $miesiac = 'marca'; break;
                case '04': $miesiac = 'kwietnia'; break;
                case '05': $miesiac = 'maja'; break;
                case '06': $miesiac = 'czerwca'; break;
                case '07': $miesiac = 'lipca'; break;
                case '08': $miesiac = 'sierpnia'; break;
                case '09': $miesiac = 'września'; break;
                case '10': $miesiac = 'października'; break;
                case '11': $miesiac = 'listopada'; break;
                case '12': $miesiac = 'grudnia'; break;
                default: $miesiac = 'niezidentyfikowany'; break;
            }
            print "$dzien $miesiac $rok";
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 41.2. Skrypt PHP napisany z użyciem instrukcji wyboru

ZADANIA DO WYKONANIA

1. Napisz skrypt w języku PHP, który wypisze liczby w kolejności od najmniejszej do największej. Wykorzystaj instrukcję warunkową.
2. Napisz skrypt w języku PHP, który obliczy wartość bezwzględną liczby.
3. Napisz skrypt w języku PHP, który wyliczy pierwiastki równania kwadratowego A, B, C.

PODSUMOWANIE LEKCJI

s. 263 • Instrukcja warunkowa na podstawie określonego warunku decyduje, który z fragmentów kodu zostanie wykonany.

s. 263 • Instrukcja wyboru umożliwia dokonanie wyboru, która instrukcja spośród wielu zostanie wykonana.

42. Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP

NA TEJ LEKCJI:

- dowiesz się, czym są pętle;
- poznasz sposób działania pętli `for` oraz pętli `while` i `do..while`.

Jedną z zalet sprzętu komputerowego jest to, że może on wykonywać wielokrotnie te same obliczenia, które mogłyby nam przysporzyć wielu trudności oraz zająć sporo czasu. Tworząc program komputerowy, często musimy wykonać pewną instrukcję czy wypisać coś na ekranie wielokrotnie po sobie.

42.1. Pętla for

pętla for

Pętla jest instrukcją programistyczną występującą w większości języków programowania. Instrukcja ta pozwala wykonać daną linijkę kodu wielokrotnie bez ponownego jej wpisywania. W języku PHP wykorzystujemy trzy rodzaje pętli: `for`, `while` i `do..while`.

Pętla `for` pozwala na deklarowanie zmiennych w obrębie jej samej, sprawdzanie prawdziwości warunków w trakcie działania oraz modyfikacje zmiennych po każdej iteracji (wielokrotnym stosowaniu tej samej procedury), co czyni ją najbardziej wszechstronną ze wszystkich pętli.

Pętla `for` ma postać:

```
for (początek: warunek_pętli: zwiększenie_licznika) {
blok instrukcji
}
```

W pierwszej części pętli (na początku) nadajemy zmiennej, która pętlą steruje, wartość początkową. Część druga (warunek_pętli) jest wyrażeniem, które określa warunek. Przy jego spełnieniu pętla ma być wykonana. W części trzeciej (zwiększenie_licznika) określamy sposób, w jaki zadeklarowana zmienna sterująca pętlą ma być zwiększona podczas każdego obiegu pętli.

Przykład

Wyświetlenie tabliczki mnożenia dla liczby 10 przy użyciu pętli `for` (rys. 42.1.).

42_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Mnożenie</title>

</head>
<body>
    <div>
    <p>
        <?php
            for ($a=1; $a<=10; ++$a){
                print "$a pomnożone przez 10 równa się " . $a*10 . "<br>";
            }
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 42.1. Pętla **for**

42.2. Pętla while

pętla while ▸ Wykorzystując pętlę `for`, musimy zadeklarować, ile razy ma zostać wykonana. To może jednak okazać się niemożliwe. Pomocna w takiej sytuacji będzie pętla typu `while` – wykonywana dopóty, dopóki warunek logiczny jest prawdziwy. Instrukcje, które zawrzemy w pętli `while`, mogą nie być wykonane w ogóle – w przypadku nieprawdziwości warunku już przy pierwszym jego wywołaniu.

Pętla `while` ma postać:

```
while (warunek_pętli) {

blok instrukcji

}
```

Przykład

Wyświetlenie tabliczki mnożenia dla liczby 10 przy użyciu pętli `while` (rys. 42.2.).

42_2.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Mnożenie</title>

</head>
<body>
    <div>
    <p>
        <?php
            $a=1;
            while ($a <=10){
                print "$a pomnożone przez 10 równa się " . $a * 10 . "<br>";
                ++$a;
            }
        ?>
    </p>
    </div>
</body>
</html>
```

Rys. 42.2. Pętla **while**

42.3. Pętla do..while

pętla do..while

Odmianą pętli `while` jest pętla `do..while`, której będziemy używać w sytuacji, gdy blok kodu ma być wykonany przynajmniej jeden raz, a jego działanie jest uzależnione od spełnienia określonych warunków.

Pętla `do..while` ma postać:

```
do {

instrukcje_do_wykonania;

}

while warunek_pętli;
```

Przykład

Wyświetlenie tabliczki mnożenia dla liczby 10 przy użyciu pętli `do..while` (rys. 42.3).

43_3.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Wyswietlanie ciągu liczb</title>
</head>
<body>
<?php
  $a=1;
  do {
      echo "$a pomnożone przez 10 równa się " . $a * 10 . "<br>";
  }
  while (++$a <=10);
?>
</body>
</html>
```

Rys. 42.3. Pętla **do...while**

ZADANIA DO WYKONANIA

1. Napisz skrypt w języku PHP, który wyświetli 1000 kolejnych liczb całkowitych, od 10 wzwyż. Wykorzystaj pętlę `for`.
2. Napisz skrypt w języku PHP, który za pomocą znaków * (gwiazdki) narysuje choinkę. Wykorzystaj pętlę `for`.
3. Napisz skrypt w języku PHP, który liczbę całkowitą wyświetli w odwrotnej kolejności cyfr. Wykorzystaj pętlę `while`.
4. Napisz skrypt w języku PHP, który liczbę całkowitą nieujemną wyświetli w notacji systemu dwójkowego. Wykorzystaj pętlę `while`.
5. Napisz skrypt w języku PHP, który wyświetli wszystkie kolejne liczby wybrane losowo, aż do chwili, gdy taką liczbą jest 3. Wykorzystaj pętlę `do..while`.
6. Napisz skrypt w języku PHP, który wyświetli sześć liczb losowanych bez powtórzeń z przedziału od 1 do 48. Wykorzystaj pętlę `do..while`.

PODSUMOWANIE LEKCJI

s. 265 ▸ • Pętla `for` jest najczęściej stosowana podczas programowania w języku PHP. Działa, dopóki warunek jest spełniony, oraz pozwala na deklarowanie zmiennych w obrębie jej samej.

s. 266 ▸ • Pętla `while` jest wykonywana dopóty, dopóki warunek logiczny jest prawdziwy. Instrukcje w tej pętli nie są wykonane ani razu, jeżeli warunek nie jest prawdziwy przy pierwszym jej wywołaniu.

s. 267 ▸ • Pętla `do..while` to odmiana pętli `while`. Jest wykonywana raz przed sprawdzeniem warunku logicznego.

43. Funkcje w języku PHP, czyli jak wzbogacić naszą stronę

NA TEJ LEKCJI:

- dowiesz się, czym są funkcje w języku PHP;
- poznasz rodzaje funkcji;
- dowiesz się, jak konstruować własne funkcje w języku PHP.

W wcześniejszych rozdziałach dotyczących programowania w języku PHP korzystaliśmy wielokrotnie z wbudowanych funkcji `print()` i `echo()`. W tym rozdziale dowiemy się, w jaki sposób użytkownik może tworzyć własne funkcje, które następnie z powodzeniem wykorzysta w tworzeniu kodu PHP.

43.1. Funkcje w PHP

funkcje ▸ **Funkcja** to fragment kodu, który można wywołać, podając jej unikatową nazwę. W języku PHP funkcje poznamy po nawiasach występujących bezpośrednio po nazwie. Stosuje się je, aby uprościć działanie kodu oraz zwiększyć jego czytelność. Funkcje dają nam możliwość zdefiniowania operacji, które wielokrotnie wykorzystamy w programie. Znajomość funkcji jest fundamentem programowania, dlatego w tym rozdziale przyjrzymy się im dokładniej.

W języku PHP mamy dwa rodzaje funkcji – wbudowane oraz tworzone przez użytkownika.

43.2. Funkcje własne

funkcja własna ▸ **Funkcja własna** jest tworzona przez użytkownika na jego potrzeby. Musi posiadać unikalną nazwę, która odróżni ją od innych funkcji. Jej budowanie rozpoczynamy od słowa `function`. Określamy również, jakie argumenty przyjmie i co będzie właściwie robić.

```
function nawa_funkcji (argument_funkcji1, argument_funkcji2,
…, argument_funkcjin) {
instrukcje
}
```

Wywołanie funkcji jest identyczne jak wywołanie funkcji wbudowanej w PHP, czyli poprzez podanie jej nazwy (rys. 43.1.).

43_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Funkcje</title>
</head>
<body>
<?php
  function bold($tekst) {
      return "<b>".$tekst."</b>";
  }
  print("Tekst niepogrubiony. <br />");
  print bold("Tekst z funkcją, która pogrubia czcionkę.");
?>
</body>
</html>
```

Rys. 43.1. Użycie funkcji zdefiniowanej przez użytkownika

Kod funkcji jest dowolnym poprawnym kodem PHP. Możemy w nim umieścić każdą rzecz, z tworzeniem funkcji włącznie. Aby funkcja zwróciła żądaną wartość, musimy skorzystać z komendy `return`. Po `return` podajemy wyrażenie generujące wartość, która ma zostać zwrócona.

43.3. Funkcje wbudowane

funkcja wbudowana

Funkcje wbudowane są zawsze gotowe do wywołania. Pozwalają na wykonanie typowych zadań. Są to na przykład funkcje daty i czasu.

Funkcja `date()` zwraca nam informacje na temat bieżącej daty i czasu. Nie ma ona żadnych argumentów, a informacje są zwracane w formie liczb (rys. 43.2.).

Przykład

43_2.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Funkcje</title>
</head>
<body>
<?php
  $data=date("l, Y-m-d");
  $czas=date("H:i");
  echo "Dzisiaj jest - $data, godz. $czas";
?>
</body>
</html>
```

Rys. 43.2. Funkcja daty i czasu

Funkcja `date()` przekształca otrzymane argumenty w ciąg znaków. Wszystko, co nie jest podane jako format daty (w naszym przypadku są to separatory: „.", „–" oraz „:"), pozostaje niezmienione.

W przykładzie l – oznacza dzień tygodnia podany słownie w języku angielskim, Y – rok zapisany w formacie YYYY, m – miesiąc w formacie mm, d – dzień w formacie dd, H – godzinę, a i – minutę.

Więcej informacji dotyczących formatów dla funkcji `date()` znajdziecie pod adresem: *https://www.php.net/manual/en/function.date.php.*

W języku PHP do kodu naszego programu możemy włączyć zawartość innego kodu znajdującego się w plikach zewnętrznych. W tym celu musimy posłużyć się funkcjami:

- `include()` – funkcja, która wstawi całą zawartość pliku oraz dołączy cały plik . Jako argument musimy podać adres dołączanego pliku. Jeśli okaże się, że plik nie istnieje, pojawi się ostrzeżenie o jego braku, ale program będzie wykonywany dalej;
- `require()` – pozwala nam wstawić również zawartość pliku lub dołączyć go w całości. Również w tym przypadku jako argument podajemy adres dołączanego pliku. Jeśli okaże się, że plik nie istnieje, pojawi się ostrzeżenie o jego braku, a wykonywanie programu zostanie przerwane.

Zapoznaj się z kodem php, który wykorzystuje powyższe funkcje (rys. 43.3. i 43.4.).

43_3.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Funkcje</title>
</head>
<body>
<?php
  function tekst_naglowka() {
      print ("Witam na mojej stronie. <br />");
  }
?>
</body>
</html>
```

Rys. 43.3. Kod php z tekstem, który będzie wykorzystany w innym pliku za pomocą funkcji **require**

43_4.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Funkcje</title>
</head>
<body>
<?php
  require('43_3.php');
  tekst_naglowka();
?>
<br />
<p> Powyższy tekst został wstawiony za pomocą funkcji require().</p>
</body>
</html>
```

Rys. 43.4. Zastosowanie funkcji **require**

ZADANIA DO WYKONANIA

1. Napisz skrypt w języku PHP, który obliczy silnię liczby i sprawdzi działanie. Zastosuj funkcję.
2. Napisz skrypt w języku PHP, który oblicza największy wspólny dzielnik i sprawdzi działanie. Zastosuj funkcję.
3. Napisz skrypt w języku PHP, który zwraca sumę liczb z zakresu liczb wprowadzonego przez użytkownika. Sprawdź najpierw, czy liczba wprowadzona jako pierwsza nie jest większa od liczby wprowadzonej jako druga.
4. Utwórz plik z funkcją, która wypisuje nagłówek strony, następnie włącz go instrukcją `include()` do kilku dokumentów i wypisuj ten nagłówek za pomocą tej funkcji.

PODSUMOWANIE LEKCJI

- **Funkcja** jest fragmentem kodu, który wywołujemy, podając jej unikatową nazwę. Stosujemy ją, aby uprościć działanie naszego programu oraz zwiększyć jego czytelność. ◂ s. 268
- **Funkcja własna** jest tworzona przez użytkownika na jego potrzeby. Musi mieć unikatową nazwę, która odróżni ją od innych funkcji. Wywołujemy ją poprzez podanie jej nazwy. ◂ s. 268
- **Funkcje wbudowane** są gotowe do wywołania. Pozwalają na wykonanie typowych zadań. ◂ s. 269
- W języku PHP możemy włączyć do kodu programu zawartość innego kodu znajdującego się w zewnętrznych plikach. W tym celu musimy posłużyć się funkcjami `include()` i `require()`. ◂ s. 269

44. Tablice, czyli jak sortować dane w języku PHP

NA TEJ LEKCJI:

- poznasz tablice w języku PHP;
- dowiesz się, do czego służy funkcja array();
- nauczysz się sortować tablicę jedno- i wielowymiarową.

Do tej pory pracowaliśmy z pojedynczymi zmiennymi, na przykład `$x=1`, `$ciag='Witaj świecie!'`. Bardzo często jednak będziemy musieli korzystać z dużej lub bardzo dużej liczby danych. Wypisywanie w kodzie setek zmiennych w postaci `$zmienna1`, `$zmienna2` jest praktycznie niewykonalne. Właśnie dlatego stosujemy tablice.

44.1. Tablice w PHP

Tablice, a raczej tablicowy typ danych daje nam możliwość grupowania elementów, które chcemy wykorzystać podczas tworzenia kodu PHP. ◂ tablice

W języku C++ dane przechowywane w tablicach muszą być jednego typu, ale w języku PHP nie ma takich ograniczeń.

tablica jednowymiarowa

Elementy tworzące tablicę są ponumerowane, każdy ma swój indeks (liczbę całkowitą nieujemną) lub klucz, na przykład `$tablica[5]` jest elementem tablicy `$tablica` o indeksie równym 5, natomiast `$tablica['nazwa']` to element tablicy `$tablica` o kluczu `nazwa`. Tak więc tablice w języku PHP dzielimy na indeksowane numerycznie i asocjacyjne, czyli indeksowane kluczami.

tablica wielowymiarowa

Jeżeli elementami tablicy są inne tablice, to mamy do czynienia z tablicą wielowymiarową. Jeżeli chcemy w tablicy umieścić dane znajdujące się w tabeli, to zastosujemy tablicę dwuwymiarową – każdy wiersz tabeli (rekord) będzie się składał z kolumn i pól. Tablicę `$t[$wiersz][$kolumna]` określimy dwoma indeksami – numerem wiersza i numerem kolumny.

44.1.1. Inicjowanie tablicy

array()

Zmienną, która jest tablicą, deklarujemy przez zastosowanie funkcji `array()` i wybór zmiennej typu tablicowego – `$tablica`:

```
$tablica=array();
```

Podczas inicjowania tablicy możemy jej elementom przypisać konkretne wartości, na przykład `$t=array(1, 2, 3, 5, 7)`. Utworzyliśmy w ten sposób tablicę `$t` składającą się z pięciu elementów – `$t[0], $t[1], $t[2], $t[3], $t[4]`. W nawiasach kwadratowych podano indeksy naszej tablicy, które zostały przydzielone automatycznie, poczynając od 0. Jest to tablica indeksowana numerycznie.

Przykład

Inicjowanie tablicy poprzez funkcję `array()` (rys. 44.1.).

45_1.php

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Tablice</title>
</head>
<body>
<?php
  $t=array(1, 2, 3, 5, 7);
  echo '$t[0] przechowuje wartość '.$t[0].'<br />';
  echo '$t[1] przechowuje wartość '.$t[1].'<br />';
  echo '$t[2] przechowuje wartość '.$t[2].'<br />';
  echo '$t[3] przechowuje wartość '.$t[3].'<br />';
  echo '$t[4] przechowuje wartość '.$t[4].'<br />';
  print_r($t);
?>
</body>
</html>
```

Rys. 44.1. Inicjowanie tablicy

W przypadku tablic asocjacyjnych nie operujemy indeksami, ale kluczami, czyli łańcuchami znaków. Sposób pokazany w poprzednim przykładzie możemy zastosować do zainicjowania tablicy asocjacyjnej `$ta=array('pierwsza' => 1, 'druga' => 2, 'trzecia' => 3, 'czwarta' => 5, 'piąta'=>7);`, przy czym `pierwsza`, `druga`, `trzecia`, `czwarta` oraz `piąta` są kluczami elementów tablicy (rys. 44.2.).

```php
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Tablica asocjacyjna</title>
</head>
<body>
<?php
   $ta=array('pierwsza' => 1, 'druga' => 2, 'trzecia' => 3, 'czwarta' => 5, 'piąta'=>7);
   echo '$ta['."'pierwsza'".'] przechowuje wartość '.$ta['pierwsza'].'<br />';
   echo '$ta['."'druga'".'] przechowuje wartość '.$ta['druga'].'<br />';
   echo '$ta['."'trzecia'".'] przechowuje wartość '.$ta['trzecia'].'<br />';
   echo '$ta['."'czwarta'".'] przechowuje wartość '.$ta['czwarta'].'<br />';
   echo '$ta['."'piąta'".'] przechowuje wartość '.$ta['piąta'].'<br />';
   print_r($ta);
?>
</body>
</html>
```

Rys. 44.2. Inicjowanie tablicy asocjacyjnej

44.2. Sortowanie tablic

Funkcja `sort()` służy do sortowania tablic jednowymiarowych. Przypisuje ona nowe klucze elementom po sortowaniu. Wszystkie istniejące wcześniej klucze zostaną usunięte.

◂ sort()

Przykład

Funkcja sortująca tablicę jednowymiarową (rys. 44.3.).

```php
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Sortowanie tablicy jednowymiarowej</title>
</head>
<body>



<?php
   function pokaz_tablice($t)
{
   echo '<pre>'."\n";
   print_r($t);
   echo "\n".'</pre>';
}

   $t1=array('polonez','fiat','syrenka');
   $t2=array('k1'=>'polonez','k2'=>'fiat','k3'=>'syrenka');
   $t3=array(2,12,9,8);
   $t4=array(0,12,9,8);

   sort($t1);
   sort($t2);
   sort($t3);
   sort($t4,SORT_STRING);

   pokaz_tablice($t1);
   pokaz_tablice($t2);
   pokaz_tablice($t3);
   pokaz_tablice($t4);
?>
</body>
</html>
```

Rys. 44.3. Sortowanie tablicy jednowymiarowej

Sortowanie tablic wielowymiarowych umożliwiają nam funkcje `usort()`, `uasort()`.

Funkcja `usort()` umożliwia sortowanie tablicy wielowymiarowej za pomocą zdefiniowanej przez użytkownika funkcji porównującej. Funkcja `usort()` tworzy

nowe indeksy, więc nie nadaje się do tablic asocjacyjnych. Do tablic asocjacyjnych stosujemy funkcje `uasort()` i `uksort()`, przy czym nie ma funkcji sortujących malejąco – porządek sortowania określamy, definiując funkcję porównującą.

Przykład

Funkcja sortująca tablicę wielowymiarową (rys. 44.4.).

```
45_2.php
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Sortowanie tablicy wielowymiarowej</title>
</head>
<body>


<?php
 function pokaz_tablice($t)
{
  echo '<pre>'."\n";
  print_r($t);
  echo "\n".'</pre>';
}
  function porownaj($zbiorkom1, $zbiorkom2)
{
  return strcmp($zbiorkom1[0], $zbiorkom2[0]);
}
  $pojazdy=array(
                  array('autobus','Autosan','2'),
                  array('autobus','Solaris','6'),
                  array('tramwaj','Pesa',1),
                  array('tramwaj','Bombardier',4),
                  array('trolejbus','Skoda',5)
                );
  usort($pojazdy,'porownaj');
  pokaz_tablice($pojazdy);
?>
</body>
</html>
```

Rys. 44.4. Sortowanie tablicy wielowymiarowej

ZADANIA DO WYKONANIA

1. Napisz program, który zapełni tablicę piętnastoma liczbami całkowitymi nieujemnymi z zakresu od 1 do 20.
2. Rozbuduj program z zadania 1. tak, aby sprawdzał, ile razy jest wylosowana dana liczba.
3. Napisz program, który zapełni tablicę dziesięcioma liczbami z zakresu od 1 do 100, a następnie ułoży je malejąco.

PODSUMOWANIE LEKCJI

s. 271 – 272

- Tablice, a raczej tablicowy typ danych, daje nam możliwość grupowania elementów, które chcemy wykorzystać podczas tworzenia kodu PHP. Dane w tablicach w języku PHP nie muszą być tego samego typu. Rozróżniamy tablice jednowymiarowe i wielowymiarowe.

s. 272

- Zmienną, która jest tablicą, deklarujemy, stosując funkcję `array()`. Podczas inicjowania możemy przypisać jej elementom konkretne wartości, np. `$t=array(1, 2, 3)`.

s. 273

- Funkcja `sort()` służy do sortowania tablic jednowymiarowych. Do sortowania tablic wielowymiarowych użyjemy funkcji `usort()`, aby posortować tablicę indeksowaną, oraz `uasort()`, aby sortować tablicę asocjacyjną.
  Funkcje `sort()` i `usort()` tworzą nowe indeksy.

45. Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP

NA TEJ LEKCJI:

- dowiesz się, czym są formularze;
- poznasz różnicę między przesyłaniem danych metodami POST i GET.

Formularze na stronach internetowych pomagają nam w gromadzeniu i przetwarzaniu danych wprowadzanych przez użytkownika. Stosujemy je podczas logowania, wypełniania ankiet czy zamawiania towarów w sklepach internetowych. Formularz jest zbudowany ze znaczników języka HTML, ale aby spełniał swoją funkcję, musi być połączony ze skryptem napisanym w języku PHP.

45.1. Metoda POST

Aby obsługiwać formularze za pomocą programu napisanego w języku PHP, potrzeba formularza, do którego będziemy wprowadzać dane, oraz skryptu, w którym umieścimy dane użytkownika.

O tym, w jaki sposób dane z formularza są przesyłane dalej, decyduje metoda zdefiniowana w atrybucie `method` znacznika `form` – w przypadku przykładowego formularza jest to metoda `POST` protokołu HTTP. ◂ POST

Przykład

46_1_1.php

```
    <meta charset="utf-8" />
    <title>Formularz</title>
</head>
<body>
<form action="" method="post">
  <div>
    Użytkownik:<br />
        <input name="login" value="" /><br />
    Hasło:<br />
        <input type="password" name="pass" value="" /><br /><br />
        <input type="submit" value="Zaloguj" name="zaloguj" />
  </div>
</form>
<?php
    if (!empty($_POST['login'])){
    if (!empty($_POST['pass'])){
    if (($_POST['login'] == "uczen") && ($_POST['pass'] == "klBsA2"))
{
print "Logowanie poprawne<br /><br />Witamy na naszej stronie :)";
}else print "Nazwa użytkownika lub hasła jest nieprawidłowa.";

}else print "Nie wpisałeś hasła!";

}else print "Żeby wejść do serwisu musisz wpisać nazwę użytkownika i hasło!";

?>
</body>
</html>
```

Rys. 45.1. Formularz z metodą **POST**

45.2. Metoda GET

Dla przykładu przedstawionego na rysunku 45.1. można również użyć metody `GET`, ale ma ona swoje ograniczenia: dane z formularza są przesyłane za pomocą

adresu URL, czyli w sposób widoczny dla użytkownika, a długość adresu URL jest ograniczona do 255 znaków.

Jeśli stosujemy metodę POST, wprowadzone dane przesyłane przez formularz nie są widoczne dla użytkownika i nie ma takiego ograniczenia co do długości adresu, jak w przypadku zastosowania metody GET.

GET ▸ Metodę GET stosuje się dla pojedynczych danych. Ponieważ są one widoczne w pasku adresu przeglądarki, nie jest to metoda bezpieczna.

Skrypt z wykorzystaniem tej metody wyglądałby tak jak na rysunku 45.2.

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>Formularz</title>
</head>
<body>
<form action="" method="get">
  <div>
    Użytkownik:<br />
        <input name="login" value="" /><br />
    Hasło:<br />
        <input type="password" name="pass" value="" /><br /><br />
        <input type="submit" value="Zaloguj" name="zaloguj" />
  </div>
</form>
<?php
    if (!empty($_GET['login'])){
    if (!empty($_GET['pass'])){
    if (($_GET['login'] == "uczen") && ($_GET['pass'] == "kl@sA2"))
{
print "Logowanie poprawne<br /><br />Witamy na naszej stronie :)";
}else print "Nazwa użytkownika lub hasła jest nieprawidłowa.";

}else print "Nie wpisałeś hasła!";

}else print "Żeby wejść do serwisu musisz wpisać nazwę użytkownika i hasło!";

?>
</body>
</html>
```

Rys. 45.2. Formularz z metodą **GET**

ZADANIA DO WYKONANIA

1. Wymień wady i zalety przesyłania danych z pomocą formularza metodami POST i GET.
2. Utwórz witrynę internetową z wykorzystaniem języka PHP zawierającą formularz, dzięki któremu będzie można zamówić ulubioną pizzę. W formularzu uwzględnij następujące pola: **imię**, **nazwisko**, **numer telefonu** i **pizza** (pole **pizza** to menu rozwijane). Dane z formularza mają być przesyłane do skryptu metodą POST.
3. Utwórz witrynę internetową z wykorzystaniem języka PHP zawierającą formularz. W formularzu wprowadź pola: imię, nazwisko, klasa, szkoła. Dane z formularza mają być przesyłane do skryptu metodą GET.

PODSUMOWANIE LEKCJI

s. 275 ▸ • Metodę POST stosuje się do przesyłania dowolnej liczby danych w sposób niewidoczny dla użytkownika.

s. 276 ▸ • Metodę GET stosuje się dla pojedynczych danych, jednak są one widoczne w pasku adresu przeglądarki. Nie jest to metoda bezpieczna.

PODSUMOWANIE PRZED SPRAWDZIANEM

29. HTML, czyli przeglądarka interpretuje język programowania stron

- **HTML** (ang. *hypertext markup language*) to hipertekstowy język znaczników. Nie jest językiem programowania, ponieważ nie układa się w nim programów komputerowych. ◂ s. 188
- Rozróżniamy dwa rodzaje znaczników: **znaczniki otwarcia** `< >` i **znaczniki zamknięcia** `</ >`. ◂ s. 188
- **CSS** (*Cascading Style Sheets*) to język opisujący reguły, według których przeglądarka wyświetlania strony WWW. ◂ s. 189

30. Budujemy stronę, czyli tabele, listy i inne elementy dobrej strony

- **Lista** w HTML porządkuje treści. Istnieją trzy różne listy: **uporządkowana, nieuporządkowana** i **lista definicji**. ◂ s. 194
- **Tabele**, podobnie jak listy, pomagają porządkować dane. ◂ s. 196
- Użycie **znacznika nagłówków** pozwala wyróżnić tekst z nagłówkiem, np. listy.
- **Odnośniki** tworzy się za pomocą znacznika `<a> </a>`. Można także używać atrybutów: `href=` do podania adresu odnośnika i `target="blank"` do wyświetlenia odnośnika w nowej karcie przeglądarki. ◂ s. 197

31. Tabele i grafika, czyli kolejne składowe stron internetowych w HTML

- Rozmiar zdjęcia sprawdzamy za pomocą opcji **Właściwości** z menu rozwiniętego po kliknięciu miniatury prawym klawiszem myszy. ◂ s. 200
- W HTML do wstawiania zdjęcia (grafiki) na stronę używa się **samozamykającego się znacznika** `<img>`. Należy pamiętać, by prawidłowo podać miejsce źródłowe grafiki wyświetlanej na stronie HTML. ◂ s. 201
- Zmniejszając zdjęcia na stronie, należy zachować ich proporcje. ◂ s. 202
- Zdjęcie może być **odnośnikiem** do miejsca na stronie, podstronie lub do innej strony w sieci. ◂ s. 204

32. Składnia stylów, czyli jak CSS pomaga w programowaniu strony

- **Elementy blokowe, sekcje, identyfikatory i klasy** mogą być wykorzystane do formatowania strony HTML za pomocą stylów CSS. ◂ s. 205
- Dzięki **selektorom** możliwe jest odniesienie formatowania stylami z pliku CSS do konkretnych elementów strony HTML. ◂ s. 205
- **Rozszerzeniem** nazwy pliku ze stylami CSS dla kodu HTML powinno być .css. ◂ s. 206
- W kodzie HTML w nagłówku `<head></head>` **należy umieścić odniesienie do konkretnego pliku CSS** formatującego stylami daną stronę, np.: ◂ s. 206

```
<link rel="stylesheet" href="nazwa_pliku_css.css" type="text/css">
```

- **Nie można nadawać takich samych nazw** identyfikatorom id w tym samym kodzie HTML. ◂ s. 206
- W pliku CSS odwołanie do części objętej selektorem `id` dokonuje się za pomocą `#`. W przypadku użycia **klasy** używa się **kropki**. Dla elementów zawartych **pomiędzy znacznikami** używa się ich nazwy bez nawiasów, np. **p**, **dd**. ◂ s. 206

33. Pliki stylów, czyli CSS w akcji

s. 211
- **Listę z odsyłaczami** można zmienić w **aktywne przyciski** za pomocą formatowania w CSS.

s. 214
- **Pseudoklasa** przypomina identyfikator, jednak przeglądarka stosuje go bez konieczności określania danych elementów za pomocą stylów.

s. 214
- Pseudoklasa `hover` przypisana w CSS do atrybutu modyfikuje zachowanie się obiektu wchodzącego w interakcję, np. z kursorem.

s. 217
- **Walidacja** to proces sprawdzania poprawności kodu.

34. Treści mogą się zmieniać, czyli elementy dynamiczne na stronie internetowej

s. 219
- **Dynamiczna strona** internetowa wykorzystuje działanie aplikacji internetowej.

s. 219
- **Aplikacja internetowa** modyfikuje stronę, wyświetlając na niej elementy zgodne z parametrami podanymi przez użytkownika, np. pogodę w danym mieście, zgodnie z danymi z bazy danych itp.

s. 222
- W sieci istnieją usługi oferujące aktualne informacje, np. prognozy pogody, notowania walut itp. w postaci **widżetów**.

s. 222
- **Widżet** na stronie internetowej pełni często funkcję **gadżetu** wyświetlającego informacje, rysunki itp. Jest także aktywnym odsyłaczem.

s. 223
- **Umieszczenie widżetu na stronie** następuje poprzez wklejenie kodu udostępnionego przez właściciela widżetu do swojej strony HTML.

35. Widoczna w internecie, czyli jak opublikować stronę

s. 225
- Dla ułatwienia korzystania z sieci WWW zastosowano łatwe do zapamiętania adresy – **domeny**.

s. 225
- Każdej domenie może być przypisany adres IP. Dane te są zapamiętane na serwerze **DNS**.

s. 225
- Wpisany do przeglądarki adres (domena) zostaje wysłany do DNS, który odczytuje z bazy adres IP serwera strony i zwraca go przeglądarce. Przeglądarka wczytuje stronę spod danego IP.

s. 227
- Do wysyłania plików na serwer służy protokół FTP.

s. 228
- Menadżer plików daje możliwość korzystania z FTP, np. Eksplorator plików z Windows.

36. CMS, czyli system zarządzania treścią strony internetowej

s. 231
- **CMS** jest oprogramowaniem pozwalającym na łatwe i szybkie tworzenie stron internetowych oraz zarządzanie jej treściami.

s. 231
- Istnieją **CMS wykorzystywane przez webmasterów** dostępne w internecie. Możesz jednak **samodzielnie utworzyć** stronę z panelem administracyjnym pozwalającym na zarządzanie jej treścią. To też będzie CMS.

s. 231
- **Layout** jest elementem strony decydującym o jej wyglądzie.

s. 231
- **Skórka** w CMS jest gotowym do użycia **layoutem**. Można ją pobrać z sieci, kupić lub wykonać samodzielnie.

s. 232
- **WordPress**, **Drupal**, **Joomla** to przykłady często stosowanych systemów zarządzania treścią.

s. 233
- Do funkcjonowania CMS jest potrzebne konto w **bazie danych serwera**.

s. 233
- CMS lub dowolne projekty stron internetowych można testować w środowisku naśladującym działanie serwera WWW, np. **Laragon** lub **Xampp**.

37. Panel i skórki, czyli tworzymy stronę w CMS

- Skórka, inaczej zwana także szablonem, może być modyfikowana przez użytkowników. Możesz użyć darmowych szablonów z kolekcji *wordpress.org* lub dostępnych w sieci pod warunkiem, że są udostępnione bez opłat. ◂ s. 236
- Dostosowanie skórki do treści twojej strony polega na modyfikacji z użyciem narzędzi Kokpitu, czyli panelu zarządzania stroną. ◂ s. 238
- Wpisy umieszczane na stronie mogą zawierać informacje w różnych formach np. zdjęć, filmów itp. Wpisy ukazują się na stronie zgodnie z projektem skórki. ◂ s. 240
- Administrator może dodawać nowych użytkowników i nadawać im role (uprawnienia). ◂ s. 243

38. Szybko i łatwo, czyli programy do tworzenia stron internetowych

- **Arkusze kalkulacyjne** mają możliwość eksportu do formatu HTML, lecz na stronie nie będą aktywne. ◂ s. 246
- **Edytory tekstów**, w tym także MS Word, umożliwiają zapisanie dokumentu w formacie strony internetowej (HTML). Mogą przy tym zachowywać formatowanie, układ elementów, grafikę, tabele, odnośniki, spisy treści itp. ◂ s. 248
- Do budowy prostych stron internetowych można skorzystać z usług tworzenia **blogów** internetowych. Dzięki nim utworzysz stronę bez znajomości języków programowania lub CMS. ◂ s. 250
- **Wadą blogów** tworzonych w darmowych usługach jest konieczność korzystania z domeny usługodawcy. Wpisy ukazują się na stronie zgodnie z projektem skórki. ◂ s. 253

39. Pierwsze skrypty, czyli poznajemy język PHP

- PHP jest interpretowalnym językiem programowania, który ma cechy języków Pearl i C. Oznacza to, że nie ma potrzeby kompilacji skryptów czy programów w nim napisanych przed ich uruchomieniem. Są one wykonywane przez serwer WWW. ◂ s. 255
- Początek i koniec skryptu PHP musi być zaznaczony przez parę znaczników `<?php ... ?>`. Możemy także zastosować znaczniki `<?..?>` lub `<%...%>`, ale ich używanie wymaga zmian w konfiguracji środowiska PHP. ◂ s. 255
- W skrypcie PHP używamy funkcji `print` w celu wypisania tekstu. Możemy użyć także funkcji `echo` i `printif`. ◂ s. 256

40. Stałe i zmienne, czyli typy danych w języku PHP

- PHP posługuje się zmiennymi, które można opisać jako pudełka. Przechowują one dane w trakcie wykonywania programu. Każda zmienna ma swoją unikatową nazwę, poprzez którą odwołujemy się do jej wartości. ◂ s. 258
- W języku PHP nie wymaga się deklaracji danych i pozwala na zmianę ich typu podczas pisania programu. Wyróżniamy następujące typy zmiennych: łańcuchy znakowe, liczby całkowite i liczby zmiennoprzecinkowe. ◂ s. 259
- Do wykonywania operacji matematycznych i logicznych służą między innymi operatory arytmetyczne, przypisania, inkrementacji i porównania. ◂ s. 261

41. Instrukcje sterujące, czyli praktyczne wykorzystanie języka PHP

s. 263 ▸
- Instrukcja warunkowa na podstawie określonego warunku decyduje, który z fragmentów kodu zostanie wykonany.

s. 263 ▸
- Instrukcja wyboru umożliwia dokonanie wyboru, która instrukcja spośród wielu zostanie wykonana.

42. Pętle, czyli jeszcze więcej możliwości wykorzystania języka PHP

s. 265 ▸
- Pętla `for` jest najczęściej stosowana podczas programowania w języku PHP. Działa, dopóki warunek jest spełniony, oraz pozwala na deklarowanie zmiennych w obrębie jej samej.

s. 266 ▸
- Pętla `while` jest wykonywana dopóty, dopóki warunek logiczny jest prawdziwy. Instrukcje w tej pętli nie są wykonane ani razu, jeżeli warunek nie jest prawdziwy przy pierwszym jej wywołaniu.

s. 267 ▸
- Pętla `do..while` to odmiana pętli `while` i różni się tym, że polecenia zawarte w pętli `do..while` będą przynajmniej raz wykonane.

43. Funkcje w języku PHP, czyli jak wzbogacić naszą stronę

s. 268 ▸
- **Funkcja** jest fragmentem kodu, który wywołujemy, podając jej unikatową nazwę. Stosujemy ją, aby uprościć działanie naszego programu oraz zwiększyć jego czytelność.

s. 268 ▸
- **Funkcja własna** jest tworzona przez użytkownika na jego potrzeby. Musi mieć unikatową nazwę, która odróżni ją od innych funkcji. Wywołujemy ją poprzez podanie jej nazwy.

s. 269 ▸
- **Funkcje wbudowane** są gotowe do wywołania. Pozwalają na wykonanie typowych zadań.

s. 269 ▸
- W języku PHP możemy włączyć do kodu programu zawartość innego kodu znajdującego się w zewnętrznych plikach. W tym celu musimy posłużyć się funkcjami `include()` i `require()`.

44. Tablice, czyli jak odzyskać dane w języku PHP

s. 271 – 272 ▸
- Tablice, a raczej tablicowy typ danych, daje nam możliwość grupowania elementów, które chcemy wykorzystać podczas tworzenia kodu PHP. Dane w tablicach w języku PHP nie muszą być tego samego typu. Rozróżniamy tablice jednowymiarowe i wielowymiarowe.

s. 272 ▸
- Zmienną, która jest tablicą, deklarujemy, stosując funkcję `array()`. Podczas inicjowania możemy przypisać jej elementom konkretne wartości, np. `$t=array(1, 2, 3)`.

s. 273 ▸
- Funkcja `sort()` służy do sortowania tablic jednowymiarowych. Do sortowania tablic wielowymiarowych użyjemy funkcji `usort()`, aby posortować tablicę indeksowaną, oraz `uasort()`, aby sortować tablicę asocjacyjną.

s. 273 ▸
- Funkcje `sort()` i `usort()` tworzą nowe indeksy.

45. Pobieranie i przetwarzanie danych, czyli obsługa formularza w języku PHP

s. 275 ▸
- Metodę `POST` stosuje się do przesyłania dowolnej liczby danych w sposób niewidoczny dla użytkownika.

s. 276 ▸
- Metodę `GET` stosuje się dla pojedynczych danych, jednak są one widoczne w pasku adresu przeglądarki. Nie jest to metoda bezpieczna.

VI. Algorytmy i programowanie

- Powtarzanie pętli i wywoływanie siebie, czyli iteracja i rekurencja w algorytmach
- Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa
- Przez wstawianie, czyli jeszcze o porządkowaniu liczb
- Komputer porządkuje liczby, czyli układamy programy sortujące
- Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu
- Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania
- Przecinając oś, czyli wyznaczanie miejsc zerowych funkcji metodą połowienia
- Prawie robi różnicę, czyli obliczanie przybliżonej wartości pierwiastka kwadratowego
- Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym
- Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak inaczej obliczyć potęgę o wykładniku naturalnym
- Suma binarna, czyli dodawanie w systemie binarnym
- Różnica binarna, czyli odejmowanie w systemie binarnym
- Iloczyn binarny, czyli mnożenie w systemie binarnym
- Iloraz binarny, czyli dzielenie w systemie binarnym
- Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby pi

46. Powtarzanie pętli i wywoływanie siebie, czyli iteracja i rekurencja w algorytmach

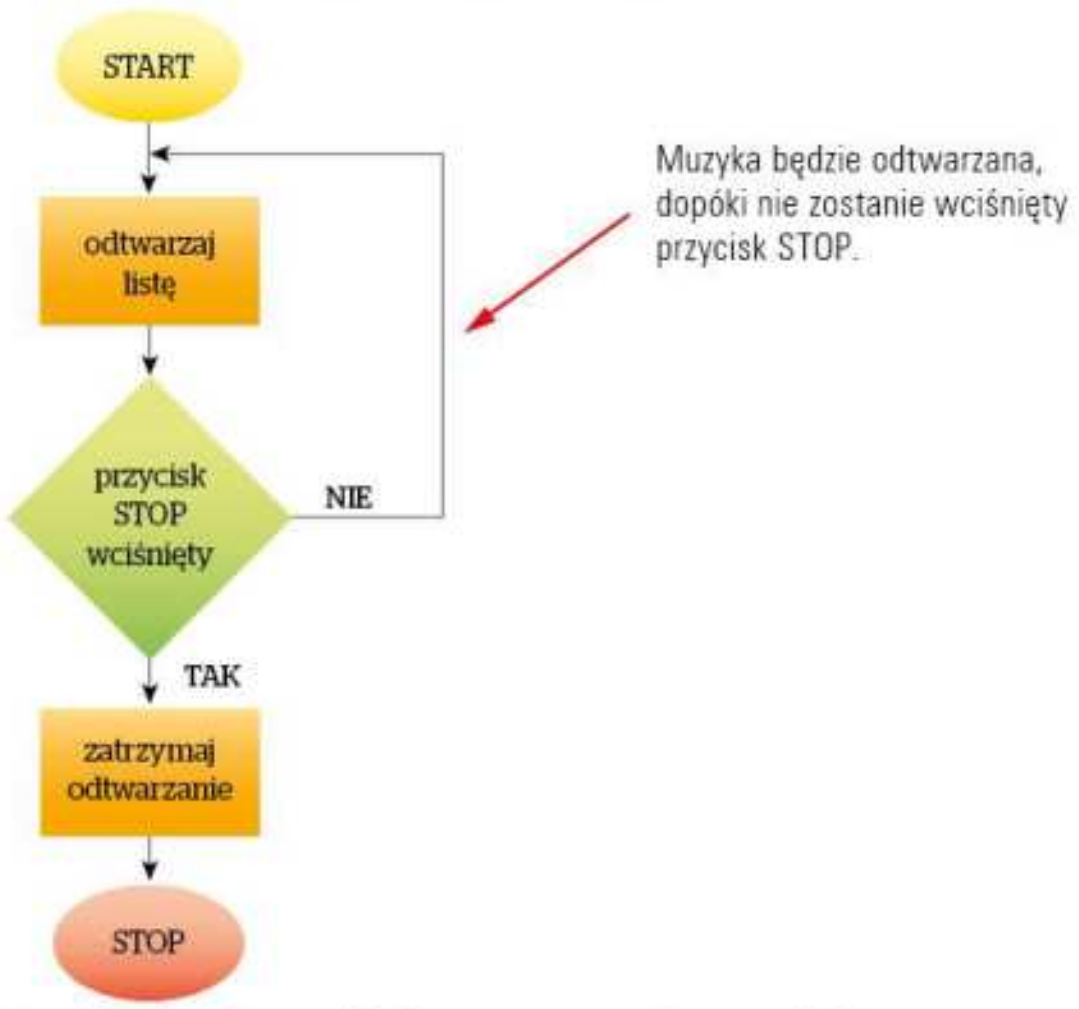
NA TEJ LEKCJI:

- poznasz różnicę pomiędzy iteracją a rekurencją w programowaniu;
- poznasz przykłady procesów rekurencyjnych i iteracyjnych.

Czy można znaleźć odpowiedź na pytanie, która z metod – iteracyjna czy rekurencyjna – jest lepsza? Jak to zwykle bywa w informatyce, nie można dać jednoznacznej odpowiedzi. Aby dobrać optymalne rozwiązanie, należy przeanalizować dany problem. Nie jest to łatwe. W zrozumieniu istoty obu rozwiązań pomoże wam lektura rozdziału.

46.1. Wiele razy ta sama sekwencja, czyli czym jest iteracja

Wyobraź sobie, że słuchasz utworów muzycznych przypisanych w telefonie do jednej listy. Chcesz, aby była ona odtwarzana, dopóki nie wciśniesz przycisku STOP. Ułóżmy algorytm takiej sytuacji (rys. 46.1.).



Rys. 46.1. Przykład algorytmu realizującego powtarzanie czynności do momentu spełnienia warunku.

W tym przypadku proces będzie się powtarzał, dopóki przycisk STOP nie zostanie wciśnięty. Nie można więc określić, jak długo ani ile razy usłyszysz ten sam utwór. Można za to użyć w programie pętli `while`, która będzie powtarzać kod, dopóki warunek jest spełniony. W naszym przypadku konieczne będzie zastosowanie warunku odwrotnego (np. przez użycie negacji !) niż w algorytmie. Odtwarzanie ma bowiem nastąpić, gdy przycisk **nie jest wciśnięty**, a więc gdy warunek z algorytmu ma wartość **false** (nie jest spełniony).

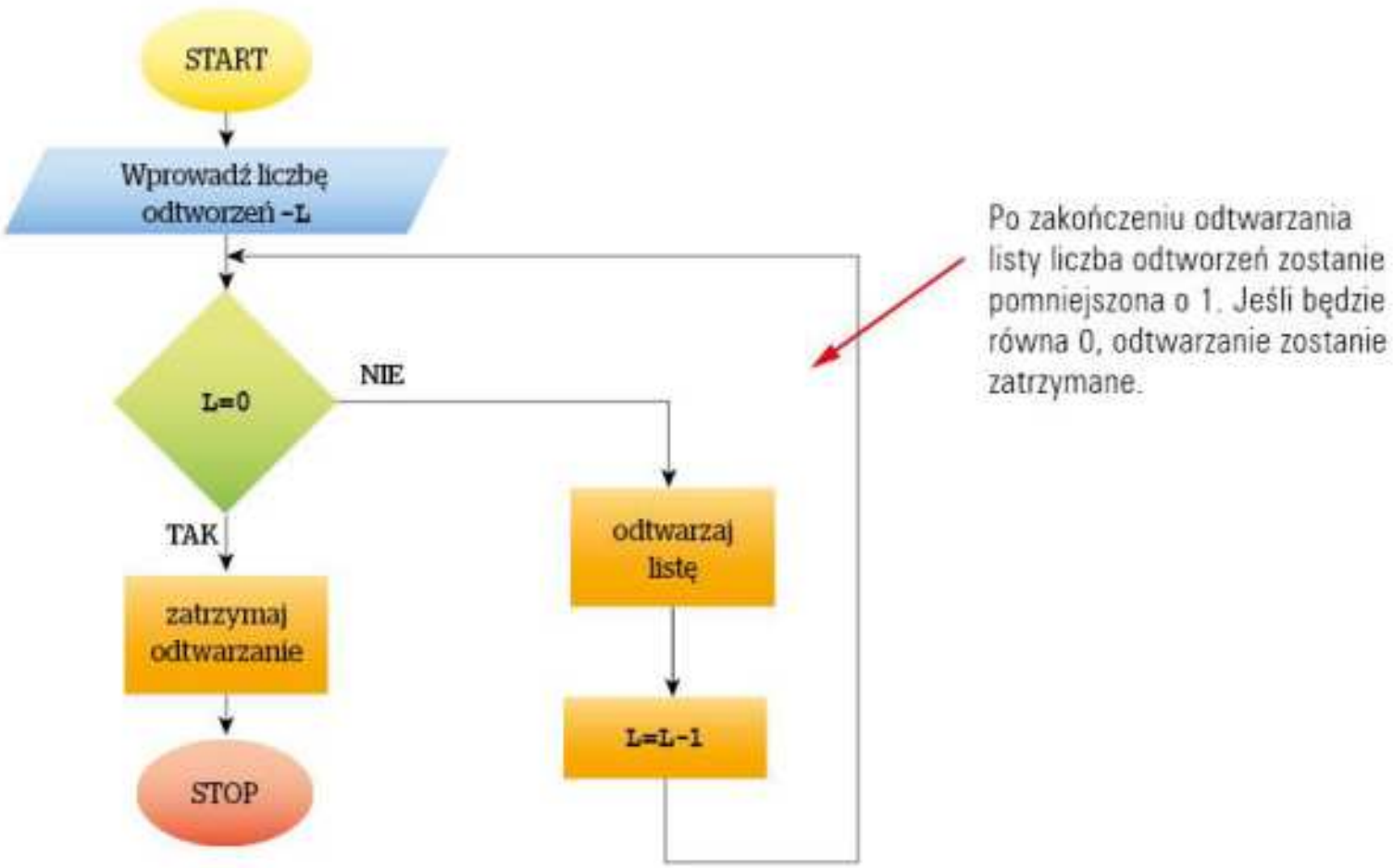
```
while (!(przycisk wciśnięty))
{odtwarzaj listę utworów}
```

Przypomnij sobie, że oprócz pętli `while` znasz także pętlę `do..while`. Jej działanie jest podobne do `while`, z tym że kod zostanie wykonany przynajmniej raz, ponieważ **badanie warunku odbywa się na końcu pętli**.

```
do
{
//kod wykonywany dopóki warunek jest spełniony
}
while (warunek);
```

Gdybyśmy w naszym przykładzie użyli `do..while`, przycisk zadziałałby dopiero po jednokrotnym odtworzeniu listy.

Algorytm można zmodyfikować i wprowadzić do aplikacji odtwarzającej muzykę opcję liczby odtworzeń (rys. 46.2.).



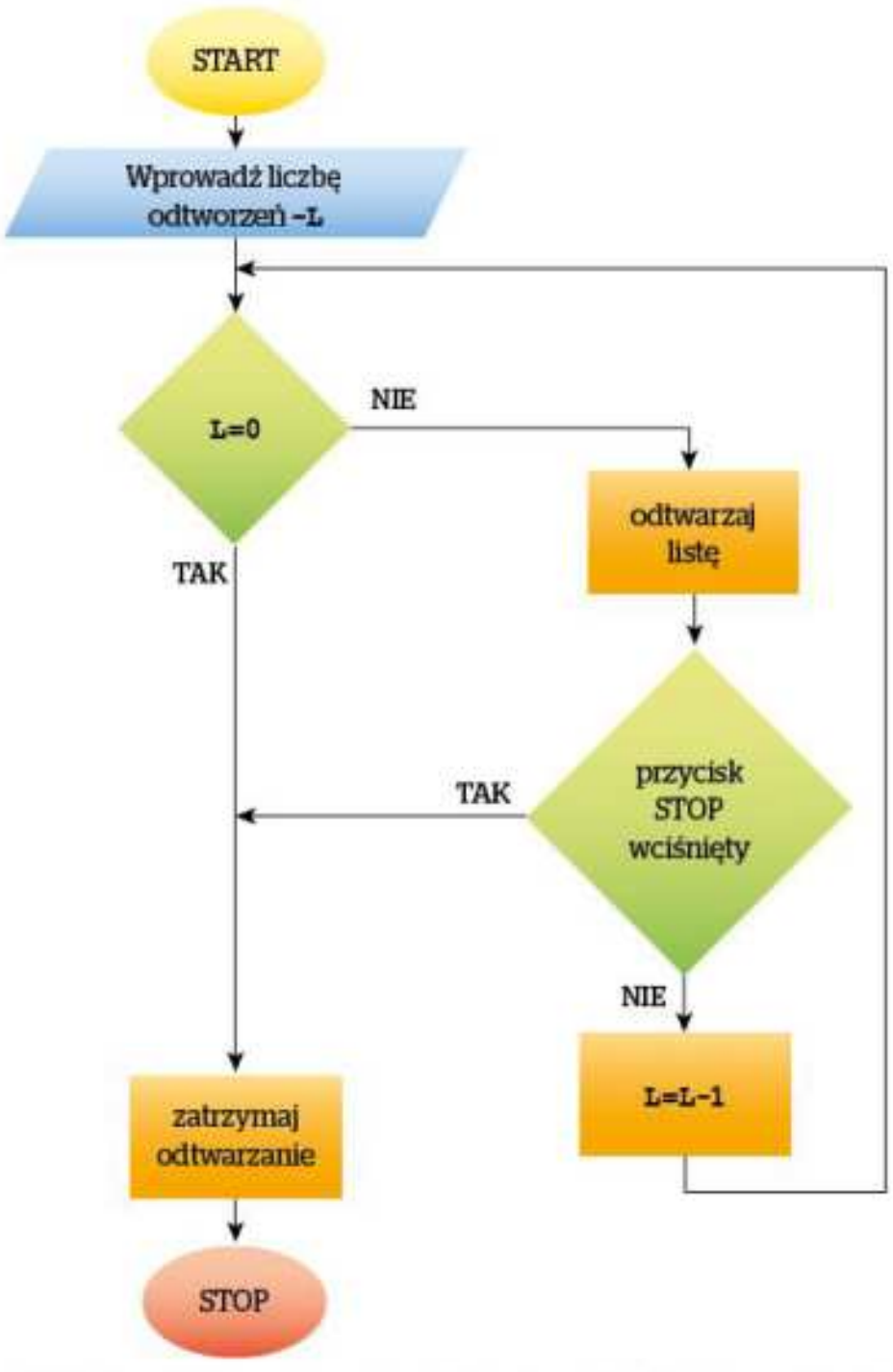
Rys. 46.2. Przykład algorytmu odtwarzania listy muzycznej *L* razy

W tym przypadku można użyć pętli `for`, która będzie wykonywana *L* razy.

Przypomnienie: W pętli `for` nadaje się wartość początkową zmiennej (można jednocześnie zadeklarować jej typ), która występuje w warunku, i podaje się, z jakim krokiem ma się zmieniać, np. `--`, co oznacza, że zmienna będzie po każdym przejściu pętli pomniejszana o 1. W naszym przypadku pętla zostanie wykonana *L* razy, czyli do osiągnięcia wartości 0 przez zmienną *i*.

```
for (int i=L; i==0; i--)
{
odtwarzaj listę utworów
}
```

Można także połączyć oba algorytmy tak, by odtwarzanie było zakończone po *L* odtworzeniach lub po wciśnięciu przycisku STOP (rys. 46.3.). W naszym przykładzie przycisk STOP zadziała także w trakcie odtwarzania listy.



Rys. 46.3. Przykładowy algorytm działania aplikacji odtwarzającej listę muzyczną

Co jest wspólną cechą tych algorytmów? **Powtarzanie**. Odtwarzanie listy utworów muzycznych jest wykonywane do momentu zajścia pewnych okoliczności – wciśnięcia przycisku STOP lub wykonania określonej liczby odtworzeń. Taki właśnie proces nazywamy **iteracją**.

Iteracja (łac. *iteratio* – powtarzanie) to czynność wykonywana określoną liczbę razy lub do spełnienia danego warunku.

◀ iteracja

46.2. Wywołanie do siebie, czyli rekurencja w informatyce

W internecie można znaleźć pewną sentencję nieznanego autora podpowiadającą, jak zrozumieć pojęcie rekurencji:

Żeby zrozumieć rekurencję, musisz najpierw zrozumieć rekurencję.

Z pozoru to zdanie nie ma sensu, każe bowiem wyjaśnić zjawisko, korzystając z wyjaśnienia tego zjawiska. Jednak w świecie informatyki nie jest ono pozbawione logiki, gdyż **rekurencja** to wywoływanie funkcji przez samą siebie. Oznacza to, że wewnątrz ciała (treści) funkcji może się znajdować odwołanie do niej samej.

Z punktu widzenia informatyki **rekurencją** nazywamy odwoływanie się funkcji lub definicji do samej siebie.

◀ rekurencja

Przykładem zagadnienia, które można programowo rozwiązać metodą rekurencyjną, jest obliczanie sumy pewnej liczby kolejnych liczb naturalnych (rys. 46.4.).

```
1 #include <cstdlib>
2 #include <iostream>
3 using namespace std;
4
5 long sumaln(int n)
6 {
7     if(n<1) return 0;
8     return n+sumaln(n-1);
9 }
10
11 int main()
12 {
13     int n;
14     cout<<"Program dodaje n kolejnych liczb naturalnych"<<endl;
15     cout<<"Ile liczb dodac? ";
16     cin>>n;
17     cout<<"Suma "<<n<<" kolejnych liczb naturalnych wynosi "<<sumaln(n)<<endl;
18     system("pause");
19     return 0;
20 }
21
```

Funkcja, która w trakcie obliczeń wywołuje samą siebie.

```
Program dodaje n kolejnych liczb naturalnych
Ile liczb dodac? 5
Suma 5 kolejnych liczb naturalnych wynosi 15
Press any key to continue . . .
```

Rys. 46.4. Przykład programu wykorzystującego rekurencję

Program główny to jedynie wprowadzanie danej i wyprowadzenie wyniku poprzez wywołanie funkcji. O wiele ciekawsza jest funkcja `sumaln`, ponieważ w swoim ciele wywołuje samą siebie. Dzięki temu dwie niepozorne linie dokonują obliczeń dla dowolnej liczby liczb ograniczonej wielkością miejsca zarezerwowanego dla wyniku działania funkcji (w tym przypadku `long`, czyli 32 bity).

```
long sumaln(int n)
{
      if(n<1) return 0;
      return n+sumaln(n-1);
}
```

Jak to się dzieje? Otóż funkcja zostaje wywołana po wczytaniu konkretnej liczby *n*. Gdy *n<1*, zwraca wartość *0* i kończy pracę. Kiedy osiągnie *0* (pomijając wprowadzenie przez użytkownika wartości *n=0*)? Wtedy, gdy doda już do siebie wszystkie kolejne liczby, począwszy od *n*-tej, aż do *1*. A jak je dodaje? W pierwszym kroku do *n* dodaje wynik funkcji `sumaln` dla argumentu *n-1*, czyli sumę wszystkich liczb naturalnych od *1* do *n-1*. W drugim kroku (wewnątrz rekurencyjnego wywołania funkcji `sumaln`) liczona jest wartość sumy *n-1* liczb naturalnych poprzez zsumowanie liczby *n-1* z sumą liczb naturalnych od *1* do *n-2*, czyli kolejnym rekurencyjnym wywołaniem funkcji `sumaln`. Tak dzieje się do momentu, gdy *n=0*. Wtedy zwracana jest wartość *0* bez wywoływania kolejnej rekurencji. Komputer składa to w odwrotnej kolejności, czyli do *1* dodaje wynik wywołania funkcji `sumaln(0)`, która przyjmuje wartość *0*. W ten sposób wróciliśmy z ostatniej rekurencji. To stanowi wynik funkcji `sumaln(1)`. Wynik ten jest dodany do liczby *2* i staje się wynikiem kolejnej rekurencji *(2+1=3)*. Wracając z następnej rekurencji, otrzymujemy *3+3=6*. Następnie *4+6=10*.

Prześledźmy to na przykładzie. Załóżmy, że wprowadzono liczbę 5. Jak widać, program podał **wynik 15**.

```
sumaln(5)=5+sumaln(4)=5+(4+sumaln(3))=5+(4+(3+sumaln(2)))=5+(4+(3+(2+(1+0))))=
5+(4+(3+(2+1)))=5+(4+(3+3))=5+(4+6)=5+10=15
```

stosowanie rekurencji i iteracji

W wielu przypadkach problemy informatyczne można rozwiązać, stosując zarówno **rekurencję**, jak i **iterację**. Programy oparte na iteracji zazwyczaj szybciej prowadzą do wyniku z powodu mniejszej liczby wywołań funkcji (mogą także nie zawierać funkcji). Programy z zastosowaniem rekurencji są zazwyczaj prostsze w budowie, a kod jest krótszy. Wybór metody należy dostosować do rozwiązywanego problemu lub algorytmu. **Iteracji** używamy głównie do przeprowadzania obliczeń na danych tablicowych lub gdy musimy wykonać pewną liczbę powtórzeń podobnej czynności. Wyznacznikiem dla użycia **iteracji** jest to, że dane pomiędzy cyklami nie muszą na siebie wpływać. **Rekurencji** używamy, gdy w kolejnych wywołaniach rekurencji używamy wyniku poprzedniego wywołania, na przykład przy liczeniu silni lub w omawianym przykładzie dodawania kolejnych liczb.

ZADANIA DO WYKONANIA

1. Wykonaj prezentację, w której przedstawisz definicje rekurencji i iteracji w informatyce. Dla każdej z nich podaj jeden przykład problemu i jego rozwiązanie.
2. Przedstaw algorytm z rysunku 46.3. w zapisie pseudokodu. Sprawdź jego działanie, opisując reakcję na polecenia użytkownika.
3. Wykonaj prezentację, w której opiszesz różnice pomiędzy poznanymi rodzajami pętli występujących w języku C++ – **while**, **do..while** i **for**. Podaj przykład problemu, który można rozwiązać przy użyciu każdej z nich.

4. Zmodyfikuj program z rysunku 46.4., aby po podaniu wyniku wyświetlił pytanie: *Czy chcesz powtórzyć obliczenia dla innej liczby liczb?* i reagował na odpowiedź t – tak lub n – nie. W przypadku wybrania t program powinien ponownie zapytać o liczbę liczb i wykonać obliczenia. Natomiast jeśli użytkownik wybierze n, program zakończy działanie i wyświetli zwrot grzecznościowy.

5*. Ułóż program obliczający silnię metodą rekurencyjną. Zastanów się, jakiego typu zmiennej użyjesz do przechowywania wyniku. Uzasadnij decyzję.

PODSUMOWANIE LEKCJI

- **Iteracja** (łac. *iteratio* – powtarzanie) to czynność wykonywana określoną liczbę razy lub do spełnienia danego warunku. W informatyce jest często stosowaną metodą układania programu. ◂ s. 285
- Z punktu widzenia informatyki **rekurencją** nazywamy odwoływanie się funkcji lub definicji do samej siebie. ◂ s. 285
- Ten sam problem programistyczny można często rozwiązać, stosując rekurencję lub iterację. Wybór metody należy dostosować do rozwiązywanego problemu lub algorytmu. ◂ s. 286

47. Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa

NA TEJ LEKCJI:

- dowiesz się, na czym polega sortowanie bąbelkowe;
- ułożysz algorytm realizujący sortowanie bąbelkowe;
- napiszesz program realizujący algorytm sortowania bąbelkowego.

Jednym z bardzo ważnych zagadnień informatycznych jest sortowanie. Porządkowanie zbiorów danych jest ważne nie tylko w bazach danych, lecz także w rozwiązywaniu innych problemów informatycznych. W tym rozdziale poznasz elementarną metodę. Sortowanie bąbelkowe jest skuteczne przy porządkowaniu niewielkich zbiorów, dlatego też stosuje się je rzadko. Stanowi jednak przykład klasycznego rozwiązania tego problemu.

47.1. Opis metody, czyli na czym polega sortowanie bąbelkowe

Napisz w zeszycie pięć różnych liczb z zakresu od 0 do 100 w przypadkowej kolejności, np.:

10	15	1	98	33

W programie komputerowym liczby te będą umieszczone w tablicy jednowymiarowej.

Bez problemu wskażesz największą liczbę w tym zbiorze. Spośród pozostałych także szybko odnajdziesz największą. Ale komputer nie działa tak jak nasz mózg. Potrzebny jest niezawodny algorytm. Istnieje sporo różnych rozwiązań tego problemu. Nasze rozważania zaczniemy od metody bąbelkowej. Skąd taka nazwa? Wyobraź sobie wysokie naczynie z wodą i wędrujące ku górze bąbelki gazu. Który wygra ten wyścig? No i masz gotową odpowiedź.

Wróćmy do naszych liczb. Ponieważ komputer nie potrafi objąć wzrokiem wszystkich i wyłuskać największej tak jak ty, musi zrobić to zgodnie z pewnym algorytmem w pewnej skończonej liczbie kroków. W ilu krokach? Policzymy to wspólnie po zakończeniu sortowania metodą bąbelkową.

Założenie: po zakończeniu sortowania liczby będą ułożone od najmniejszej z lewej strony do największej z prawej.

Krok 1.

Porównanie 1.

Weźmy dwie pierwsze liczby 10 i 15. 10<15, czyli dla tej pary kolejność jest zgodna z założeniem. Nie zmieniamy jej.

Porównanie 2.

Kolejna para to 15 i 1. 1<15, a więc zgodnie z założeniem zamieniamy je miejscami.

10	15 ↔	1	98	33

10	1	15	98	33

Porównanie 3.

Porównajmy następną parę 15<98. Nic się więc nie zmieni.

Porównanie 4.

98>33, dlatego zamieniamy je miejscami.

10	1	15	98 ↔	33

10	1	15	33	98

Teraz można już powiedzieć, że 98 jest największą liczbą w zbiorze. Zauważyłeś/zauważyłaś zapewne, że gdyby 98 znajdowała się pierwotnie na pierwszej pozycji z lewej, w każdej parze byłaby liczbą większą i przesuwałaby się w prawo (zamieniała miejsce w każdej parze).

Krok 2.

Porównanie 1.

Wracamy do pierwszej pary z lewej strony. 10>1, a więc obie liczby zamieniają się miejscami.

1	10	15	33	98

Porównanie 2.

10<15, zatem nie zmieniamy kolejności w tej parze.

Porównanie 3.

15<33, kolejność pozostaje bez zmian. Nie ma sensu porównywać 33 z 98, ponieważ wiadomo, że 98 jest liczbą największą w zbiorze.

Uwaga!

Widzisz już, że liczby zostały posortowane. Niestety, komputer realizujący algorytm o tym nie wie i musi przeprowadzić działania do końca.

Krok 3.

Porównanie 1.

1<10, nic się nie zmienia.

Porównanie 2.

10<15, kolejność pozostaje bez zmian. Wiemy za to, że 33 jest już na swoim miejscu i nie porównujemy jej z liczbą 15.

Krok 4.

Porównanie 1.

1<10, nie zmieniamy więc kolejności. 15 jest już na swoim miejscu, dlatego możemy zakończyć działanie algorytmu.

Ostatecznie otrzymujemy posortowany zbiór liczb.

1	10	15	33	98

Do posortowania pięciu liczb potrzebowaliśmy **czterech przejść algorytmu**. W każdym kolejnym wykonaliśmy o jedno porównanie mniej. W praktyce, co sprawdzisz, układając program, funkcja sortowania będzie zawsze w tej metodzie uruchamiana *n-1* razy.

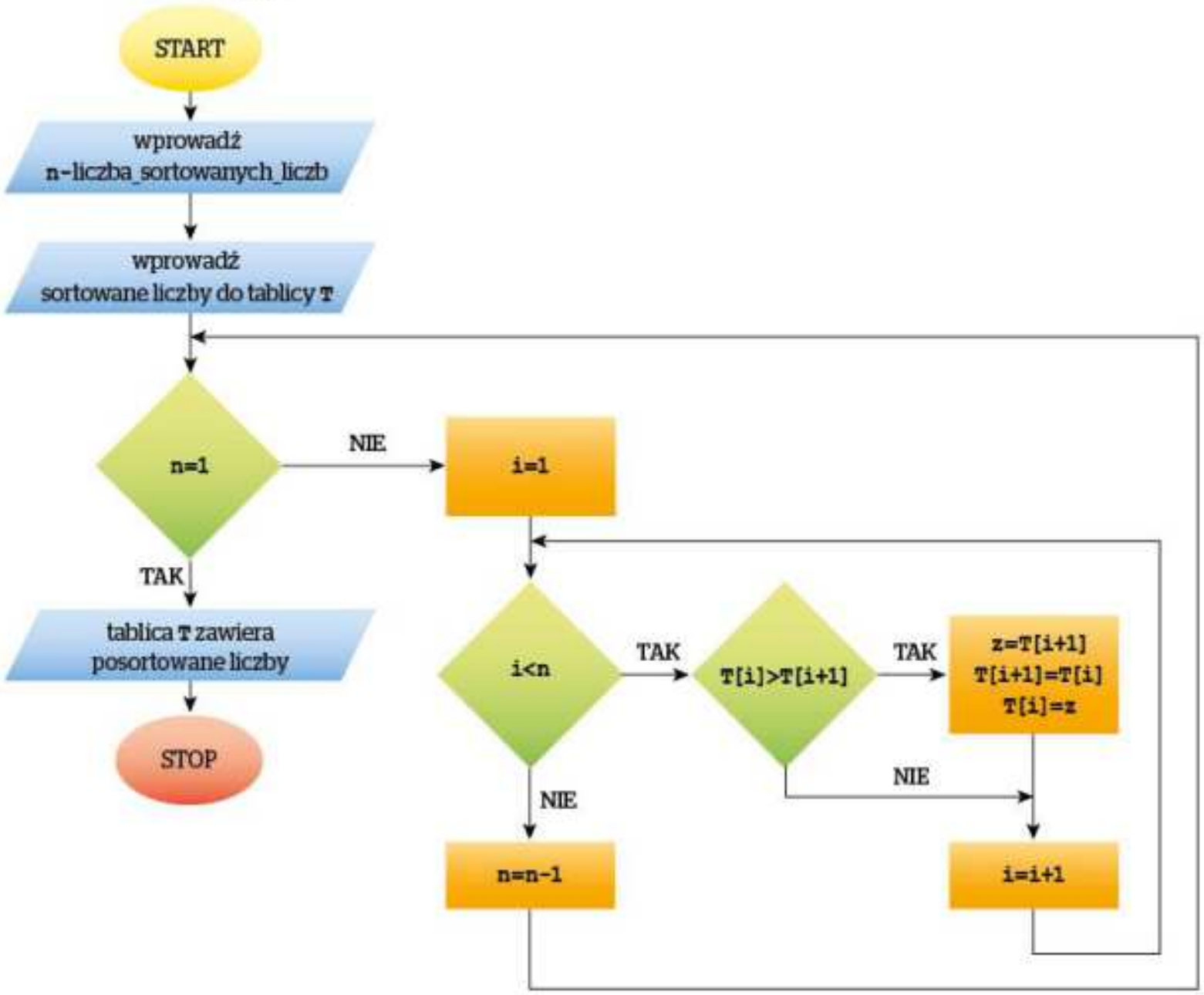
Zdefiniujmy tę metodę:

sortowanie bąbelkowe

Sortowanie bąbelkowe polega na porównywaniu kolejno dwóch sąsiednich (występujących obok siebie w tablicy) liczb i zamianie ich miejscami, w przypadku, gdy nie są uporządkowane, czyli liczba o niższym indeksie jest większa. Operacje zamiany wykonujemy tak długo, aż zbiór zostanie posortowany.

47.2. Sortowanie bąbelkowe w praktyce, czyli układamy algorytm

Na podstawie definicji, opisu i doświadczenia na liczbach z poprzedniego podrozdziału można ułożyć algorytm dla metody sortowania bąbelkowego (rys. 47.1.).



Rys. 47.1. Przykładowy algorytm sortowania bąbelkowego

W powyższym algorytmie dane liczby znajdują się w jednowymiarowej tablicy *T* o rozmiarze *n*, równym liczbie liczb podlegających sortowaniu. Wprowadzono także dwie zmienne pomocnicze. Za pomocą *i* indeksuje się elementy tablicy. Zmienna *z* pozwala przechować liczbę w procesie zamiany kolejności w danej parze.

Zweryfikujmy poprawność działania algorytmu (tab. 47.1.).

Lp.	Warunki	Sortowanie wartości zmiennych	Stan tablicy T				
1.	Start	n=5, wprowadzenie danych do **T**	3	10	2	1	4
2.	n≠1	n=5, i=1					
3.	i<n	T[1]<T[2], n=5, i=2	3	10	2	1	4
4.	i<n	T[2]>T[3] – zamiana n=5, i=3	3	2	10	1	4
5.	i<n	T[3]>T[4] – zamiana n=5, i=4	3	2	1	10	4
6.	i<n	T[4]>T[5] – zamiana n=5, i=5	3	2	1	4	10
7.	i=n	n=4, i=5					
8.	n≠1	n=4, i=1					
9.	i<n	T[1]>T[2] – zamiana n=4, i=2	2	3	1	4	10
10.	i<n	T[2]>T[3] – zamiana n=4, i=3	2	1	3	4	10
11.	i<n	T[3]<T[4], n=4, i=4	2	1	3	4	10
12.	i=n	n=3, i=4					
13.	n≠1	n=3, i=1					
14.	i<n	T[1]>T[2] – zamiana n=3, i=2	1	2	3	4	10
15.	i<n	T[2]<T[3], n=3, i=3	1	2	3	4	10
16.	i=n	n=2, i=3					
17.	n≠1	n=2, i=1					
18.	i<n	T[1]<T[2], n=2, i=2	1	2	3	4	10
19.	i=n	n=1, i=2					
20.	n=1		1	2	3	4	10
21.	Stop						

Tab. 47.1. Weryfikacja działania algorytmu dla przykładowych liczb

Algorytm spełnił swoje zadanie i w przewidywanej liczbie kroków posortował przykładowe liczby. Zauważ, że gdy większa liczba z pary znajdowała się po prawej stronie, ułożenie elementów się nie zmieniało. Jest to efekt niespełnienia warunku *T[i]>T[i+1]*.

sortowanie liter

Jak sortować litery? Przypomnij sobie, jak są one przechowywane w pamięci komputera. Każdemu znakowi przyporządkowuje się liczbę – jego kod ASCII. A skoro litery dla programu komputerowego są liczbami, to ich sortowanie przebiega zgodnie z algorytmami sortowania liczb. Jeśli przyjrzysz się uważnie tabeli kodów ASCII, z pewnością dojdziesz do wniosku, że małe litery mają większe liczby w ASCII od dużych i zgodnie z tym zostaną posortowane (tab. 47.2.). Kody pierwszych liter alfabetu są mniejsze od ostatnich, co pozwala na ich układanie według alfabetu. Znakom typu przecinek, nawias itp. w ASCII odpowiadają liczby mniejsze od kodów liter.

Litery przed sortowaniem:

Litera	a	A	C	c	u	F
ASCII	97	65	67	99	117	70

Po sortowaniu:

Litera	A	C	F	a	c	u
ASCII	65	67	70	97	99	117

Tab. 47.2. Przykład sortowania liter zgodnie z kodami ASCII

ZADANIA DO WYKONANIA

1. Sprawdź działanie algorytmu sortowania bąbelkowego dla danych, w których wielokrotnie występują te same liczby. Wykonaj symulacje w tabeli podobnej do tab. 47.1. Opisz krótko, jak się zachowuje algorytm dla par identycznych liczb.
2. Policz, a następnie zweryfikuj doświadczalnie, ile razy zostanie wykonane porównywanie par dla $n=12$ w algorytmie sortowania bąbelkowego. Wyniki pośrednie zapisuj w postaci dwunastoelementowych tablic.
3. Zapisz algorytm sortowania bąbelkowego w postaci pseudokodu.
4. Ułóż algorytm sortujący litery zapisane w tablicy jednowymiarowej. Zweryfikuj działanie tego programu. Czym różni się program sortujący liczby od programu sortującego litery?

5*. Ułóż algorytm układający według alfabetu n imion. Opisz jego działanie w dokumencie tekstowym.

PODSUMOWANIE LEKCJI

s. 290 • **Sortowanie bąbelkowe** polega na porównywaniu kolejno dwóch sąsiednich (występujących obok siebie w tablicy) liczb i zamianie ich miejscami, w przypadku, gdy nie są uporządkowane, czyli liczba o niższym indeksie jest większa.

s. 290 • **Algorytm sortowania bąbelkowego** powtarzany jest n-1 razy, a n jest liczbą sortowanych elementów.

s. 290 • **Podczas sortowania bąbelkowego największa liczba** jest przesuwana na ostatnią pozycję, której nie zajmuje już większa od niej.

s. 291 • **Litery** są sortowane według odpowiadających im liczb – **kodów ASCII**.

48. Przez wstawianie, czyli jeszcze o porządkowaniu liczb

NA TEJ LEKCJI:

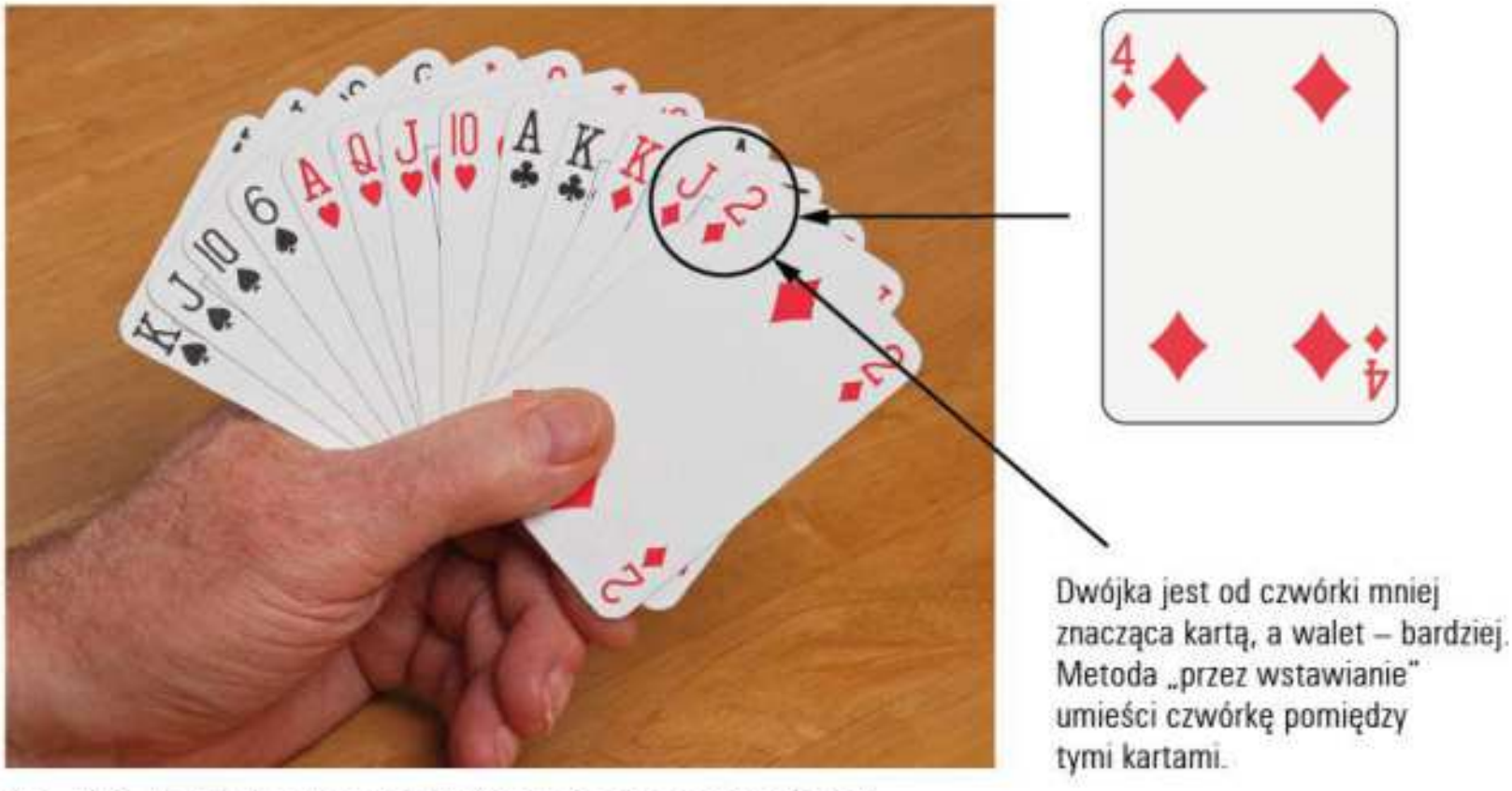
- poznasz metodę sortowania przez wstawianie;
- ułożysz algorytm dla metody sortowania przez wstawianie;
- przeprowadzisz symulację działania algorytmu sortującego przez wstawianie.

Poznałeś/Poznałaś metodę sortowania bąbelkowego. Wiesz, że mimo iż wykona swoje zadanie, jest nieefektywna w sortowaniu wielu obiektów, np. kilku tysięcy liczb. Zadajesz sobie sprawę, że istnieją inne algorytmy sortujące. Jeden z nich opisuje metodę przez wstawianie.

48.1. Dobieramy karty, czyli istota metody sortowania przez wstawianie

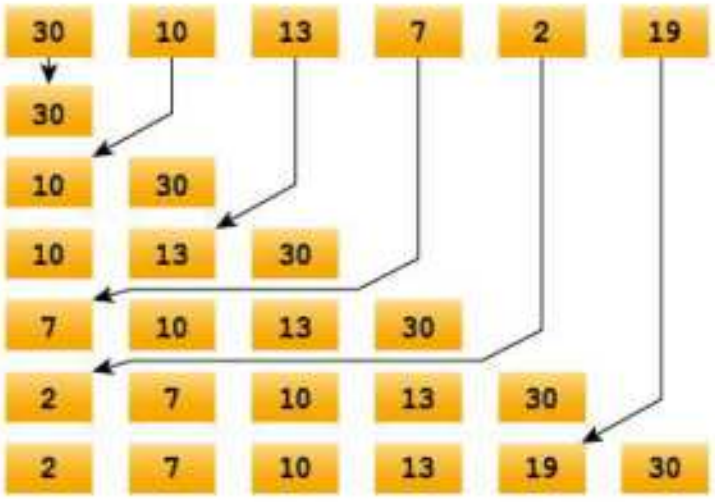
sortowanie przez wstawianie

Czy algorytm sortowania przez wstawianie ma coś wspólnego z grą w karty? Istnieją takie gry karciane, w których dobiera się karty. Dla porządku gracz przekłada je tam, gdzie bez trudu odnajdzie, np. pomiędzy „starszą" od niej a „młodszą". Przykład takiej sytuacji znajdziesz na rysunku 48.1. Kolejno dobierane karty trafiały na swoje miejsce. Przyszła kolej na 4 karo. Pomiędzy które karty należy ją włożyć, by znalazła się na właściwym miejscu?



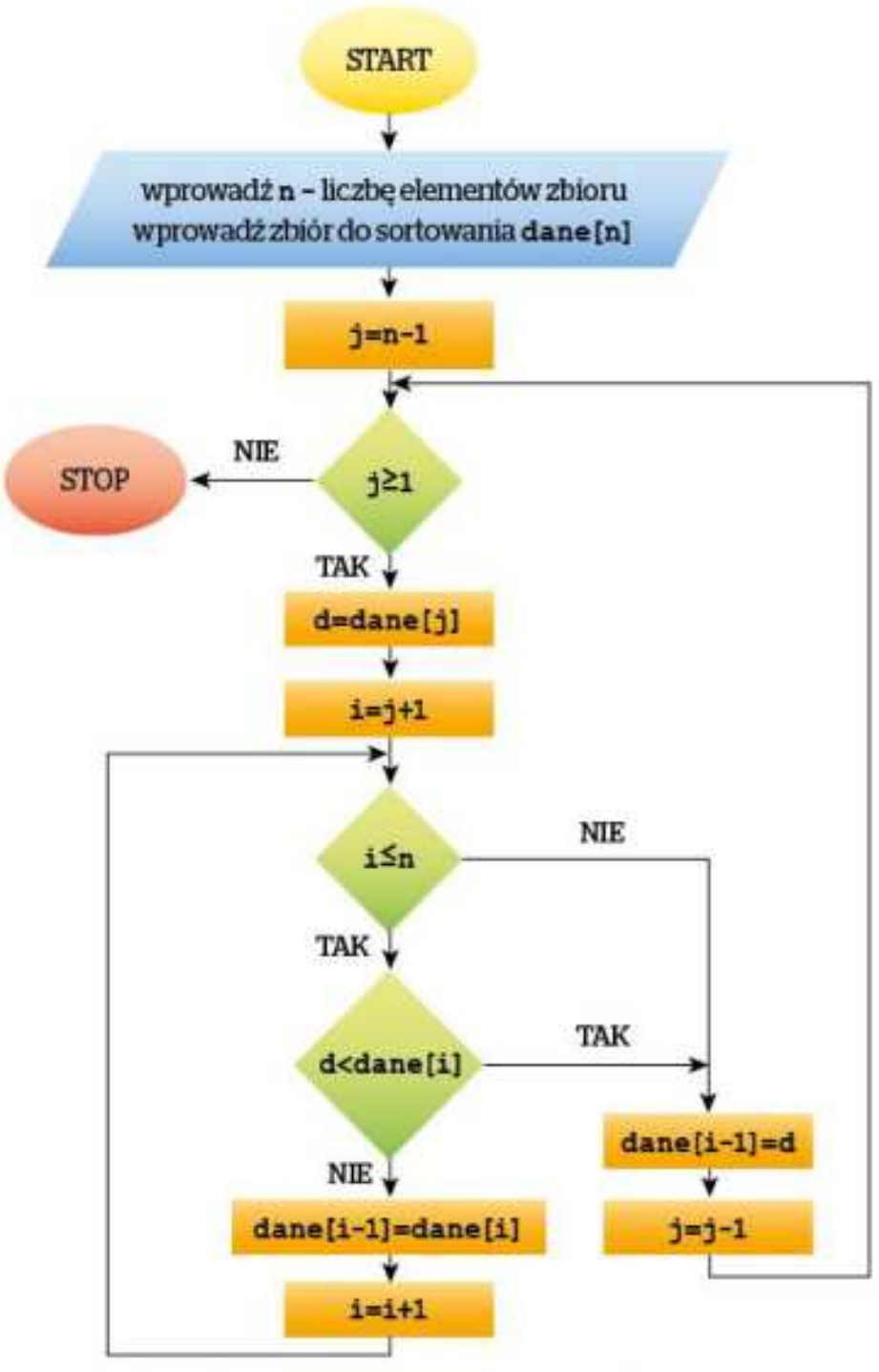
Rys. 48.1. Przykład sortowania kart metodą „przez wstawianie"

Oczywiście algorytm sortowania przez wstawianie dotyczy także liczb, a więc ma zastosowanie również w informatyce. Zrozumiesz jego działanie, gdy przeanalizujesz rysunek 48.2.



Rys. 48.2. Ilustracja metody sortowania liczb „przez wstawianie"

Ułóżmy algorytm w postaci schematu blokowego (rys. 48.3.).



Rys. 48.3. Algorytm sortowania metodą „przez wstawianie"

liczba występuje w tablicy dwukrotnie

W trakcie realizacji algorytmu zachodzą sytuacje, w których **jedna z liczb wystąpi w tablicy danych dwukrotnie** – na swoim nowym miejscu i na poprzednim. Powoduje to pozorne wykasowanie zmiennej, którą zastąpiła. W tym momencie zastąpioną liczbę przechowuje zmienna `d`.

W algorytmie założono, że ostatni element tablicy `dane[n]` na początku pozostanie na swojej pozycji. Sortowanie zaczniemy od pozycji o 1 niższej (`j=n-1`). Element z pozycji `j` zostaje zapamiętany pod nazwą zmiennej pomocniczej `d`. Tablicę wyjściową nazwijmy uporządkowaną. W trakcie działania algorytmu po jej prawej stronie na właściwych miejscach będą pojawiały się kolejne liczby. Oznacza to, że tablica podczas sortowania zostaje podzielona na dwie części – posortowaną (po prawej) i jeszcze nieposortowaną.

Pętla „zewnętrzna" wybiera z tablicy kolejne elementy od przedostatniego (`j=n-1`), aż do pierwszego. Pętla wewnętrzna wyszukuje miejsca dla danego elementu, podobnie jak gracz dla karty, a po jego znalezieniu pętla zewnętrzna wstawia ten element na swoim miejscu w aktualnej tablicy. Czynności powtarzają się aż do `j=0`.

Przeprowadźmy symulację działania algorytmu dla trzech liczb w tablicy `dane[3,7,1]` (tab. 48.1.).

Dane wejściowe:

```
n=3
dane[n]=dane[3,7,1]
```

Operacja algorytmu	Wynik operacji	Stan tablicy danych
j=n-1	n=3, j=2	dane[3,7,1]
j≥1	2≥1 tak	dane[3,7,1]
d=dane[j]	d=dane[2]=7	dane[3,7,1]
i=j+1	i=2+1=3	dane[3,7,1]
i≤n	i≤3 tak	dane[3,7,1]
d≤dane[3]	7≤1 nie	dane[3,7,1]
dane[i−1]=dane[i]	dane[2]=dane[3]=1	dane[3,1,1]
i=i+1	i=3+1=4	dane[3,1,1]
i≤n	4≤3 nie	dane[3,1,1]
dane[i−1]=d	dane[3]=7	dane[3,1,7]
j=j−1	j=2−1=1	dane[3,1,7]
j≥1	1≥1 tak	dane[3,1,7]
d=dane[1]	d=3	dane[3,1,7]

Operacja algorytmu	Wynik operacji	Stan tablicy danych
i=j+1	i=1+1=2	dane[3,1,7]
i≤n	2≤3 tak	dane[3,1,7]
d≤dane[i]	3≤1 nie	dane[3,1,7]
dane[i−1]=dane[i]	dane[1]=dane[2]=1	dane[1,1,7]
i=i+1	i=2+1=3	dane[1,1,7]
i≤n	3<=3 tak	dane[1,1,7]
d≤dane[i]	3≤7 tak	dane[1,1,7]
dane[i−1]=d	dane[2]=3	dane[1,3,7]
j=j−1	j=1−1=0	dane[1,3,7]
j≥1	0≥1 nie	dane[1,3,7]
Stop	dane posortowane	dane[1,3,7]

Tab. 48.1. Symulacja działania algorytmu sortującego przez wstawianie dla przykładowych danych

W realizacji programowej w języku C++ należy użyć pętli. Przekonasz się o tym w następnym rozdziale.

ZADANIA DO WYKONANIA

1. W rozdziale podano przykład sortowania przez wstawianie. Omów inny przykład, w którym zastosowano takie sortowanie.
2. Wzorując się na rysunku 48.2., wykonaj symulację sortowania przez wstawianie dla liczb [13, 45, 33, 19, 1, 22, 45]. Przedstaw sytuację, w której wystąpiły dwie identyczne liczby.
3. Tablica składa się z pięciu liter. Zamień je na kody ASCII i wzorując się na rysunku 48.2, posortuj metodą przez wstawianie.
4. Ułóż prezentację omawiającą zasadę sortowania przez wstawianie. Podaj w niej przykłady.
5*. Przeprowadź symulację działania algorytmu z rysunku 48.3. dla dowolnie ułożonych liczb pierwszych z przedziału od 2 do 15. Wzoruj się na tabeli 48.1.

PODSUMOWANIE LEKCJI

s. 293 ▸
- **Przykładem sortowania przez wstawianie** jest układanie kart według kolejności ustalonej w grze. Kolejno dobierane karty trafiają pomiędzy „młodszą" a „starszą".

s. 295 ▸
- Jedna z liczb może **wystąpić dwukrotnie w tablicy danych**: na swoim nowym miejscu i na poprzednim. Powoduje to pozorne wykasowanie zmiennej, którą zastąpiła – zastąpioną liczbę przechowuje inna zmienna.

49. Komputer porządkuje liczby, czyli układamy programy sortujące

NA TEJ LEKCJI:

- ułożysz programy sortujące na podstawie poznanych algorytmów;
- zweryfikujesz poprawność programów;
- poznasz działanie funkcji swap.

Wiesz już, jak przebiega sortowanie bąbelkowe i przez wstawianie. Algorytmy nie są skomplikowane i jeśli się postarasz, ułożysz na ich podstawie programy w języku C++. Jeśli będziesz mieć tym problemy, przeczytaj ten rozdział.

49.1. Bąbelki w C++, czyli układamy program na podstawie algorytmu sortowania bąbelkowego

Algorytm sortowania bąbelkowego (rys. 49.1.) wymaga użycia 2 pętli. Pierwsza wykonywana jest *n-1* razy, czyli do momentu posortowania całej tablicy. Wewnątrz niej działa pętla porównująca sąsiednie elementy tablicy: `T[j]` z `T[j+1]`. Jeśli element `T[j+1]` jest mniejszy od `T[j]` to zamieniają się one miejscami. Porównanie wymaga badania warunku, dlatego użyto instrukcji `if`.

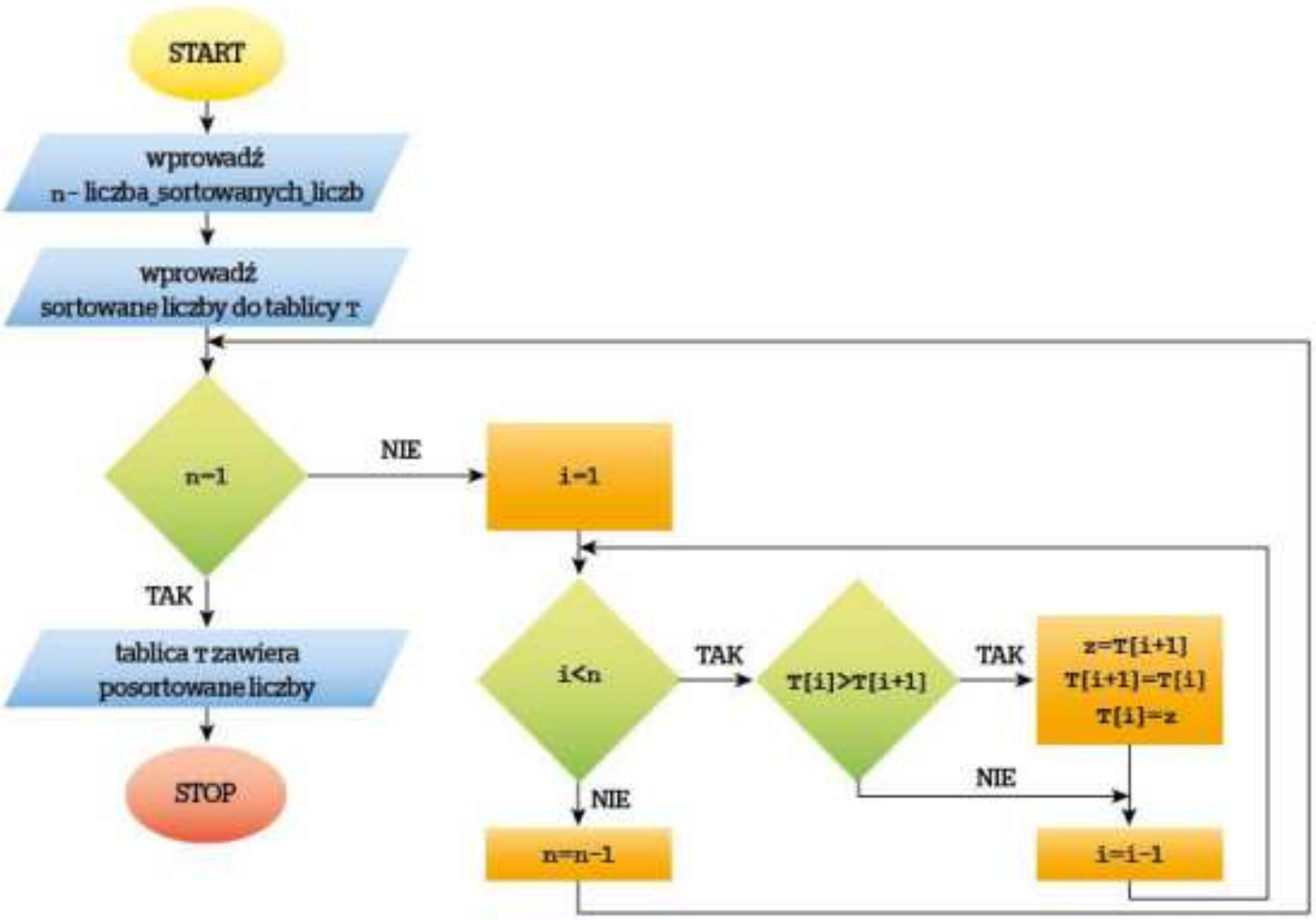
Zanim zaczniesz analizować przykładowy program, spróbuj ułożyć go samodzielnie. Nie zrażaj się początkowymi trudnościami. Przez cały czas tworzenia programu miej wgląd w algorytm i porównuj go ze swoim kodem. Wykorzystaj funkcję `swap`, która wykona zamianę liczb na wskazanych pozycjach, jeśli podasz jej odpowiednie argumenty:

```
swap (T[j], T[j+1]);
```

Ta podpowiedź powinna ci pomóc w programowaniu kluczowej operacji w algorytmie sortowania bąbelkowego.

Zwróć uwagę na to, że algorytm przewiduje wprowadzenie na wstępie liczby liczb do sortowania. Ponieważ liczbę najlepiej jest umieścić w tablicy jednowymiarowej, jej deklaracja może zostać wykonana dopiero po tej czynności.

Zastanów się, w którym miejscu programu ją umieścić i jak zapisać. W obu przykładach występuje podobny problem, jednak rozwiązany został w różny sposób.



Rys. 49.1. Algorytm sortowania bąbelkowego

iteracje w sortowaniu

Przejdźmy do przykładowego programu (rys. 49.2.). Zastosowano w nim konstrukcję iteracyjną.

```
1 #include<iostream>
2 #include<cstdlib>
3 using namespace std;
4
5 void sortb(int T[],int n)        //funkcja sortujaca
6 {
7 for (int i=0; i<n; i++)
8     for (int j=0; j<n; j++)
9         if (T[j]>T[j+1])
10            swap(T[j], T[j+1]);
11 }
12
13 int main()
14 {
15     int* T, n;
16     cout<<"Podaj liczbę sortowanych liczb  ";
17     cin>>n;
18
19     T = new int [n];          //zarezerwowanie miejsca w pamięci
20                               //dla n-elementowej tablicy T
21
22     for(int i=0;i<n;i++)      //wczytanie liczb do tablicy T
23         cin>>T[i];
24
25     sortb(T,n);               //wywołanie funkcji sortującej liczby
26
27     for(int i=0;i<n;i++)
28           cout<<T[i]<<" ";    //wyświetlenie w konsoli
29                               //posortowanej tablicy T
30
31   cout<<endl;
32   system("pause");
33   return 0;
34 }
```

Funkcja **sortb** oparta na algorytmie z rys. 29.1.
Funkcja **swap** zamienia wartości dwóch zmiennych, co jest podstawą działania algorytmu sortowania bąbelkowego.

Efekt działania programu dla przykładowych liczb użytych do testowania algorytmu w rozdziale 47.2.

```
Podaj liczbę sortowanych liczb  5
5
10
2
1
4
1 2 3 4 10
Press any key to continue . . .
```

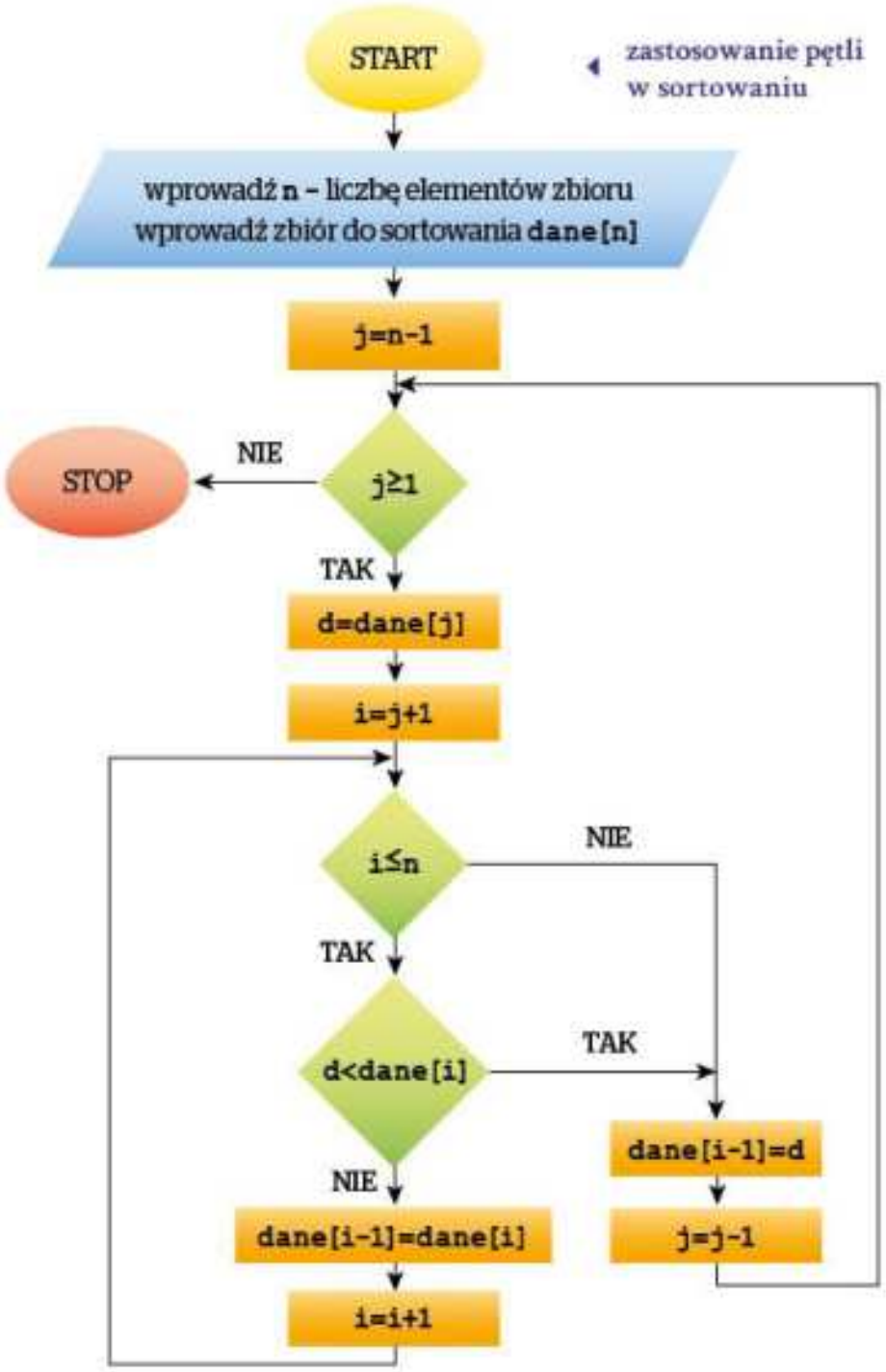
Rys. 49.2. Program sortujący liczby metodą bąbelkową

Zdefiniowano w tym programie funkcję `sortb`, która wykonuje część algorytmu odpowiedzialną za badanie relacji pomiędzy dwiema liczbami i ewentualną zamianę. Jest ona wywoływana *n–1* razy, a więc tyle, ile zakłada algorytm. W omawianym programie zmienna *i* występuje w funkcji i w programie głównym. Nie przechowuje ważnych danych, a jedynie spełnia funkcje pomocnicze.

W programie występuje jeszcze jeden element wymagający wytłumaczenia. Otóż, na początku program „nie wie", ile elementów będzie sortował. „Dowie się", gdy użytkownik wprowadzi tę informację, a on zapamięta tę liczbę pod nazwą zmiennej *n*. Dopiero wtedy można rezerwować w pamięci odpowiednią ilość miejsca na tablicę *n-wymiarową*. Osiągamy to w linii 23.

49.2. Wstawianie w programie, czyli układamy program sortujący przez wstawianie

Sortowanie przez wstawianie wydaje się prostym do rozwiązania problemem programistycznym. Będzie takim, jeśli dokładnie odwzorujesz algorytm (rys. 49.3.). Jak to zwykle bywa w takich przypadkach, trzeba użyć kilku znanych konstrukcji. Ponieważ działania sortujące według algorytmu należy wykonać *n–2* razy, to i pętla będzie musiała się tyle razy powtórzyć. Spróbuj swoich sił i ułóż program samodzielnie. Możesz zastosować główną pętlę z instrukcją `for` i wewnętrzne z instrukcją `while`. A może wpadniesz na inny pomysł? Pamiętaj o zweryfikowaniu działania programu dla różnych danych. Podobnie jak w przypadku sortowania bąbelkowego, jeśli napotkasz problemy, przeanalizuj przykładowy program (rys. 49.4.)



Rys. 49.3. Algorytm sortowania przez wstawianie

W przykładowym programie kluczowym elementem jest zmienna pomocnicza `d`. Przechowuje ona liczbę, która może w wyniku sortowania ulec „zamazaniu" przez liczbę przesuniętą na jej miejsce. Dzięki temu można ją ponownie umieścić w tablicy `dane [ ]` na miejscu „poniżej" liczby przesuniętej. Zwróć uwagę na operatory w pętli `while` i porównaj je z operatorami w blokach decyzyjnych algorytmu. Na pierwszy rzut oka wydają się błędne. Zwróć jednak uwagę w algorytmie na kierunek linii TAK i NIE oraz operator. Zrozumiesz, dlaczego użyto w przykładzie właśnie takiego zapisu.

Podobnie jak poprzedni program, ten także stosuje konstrukcję iteracyjną.

sortowanie wstawianie.cpp

```
#include <iostream>
#include <cstdlib>

using namespace std;
int d,i,j,n;

int main()
{
    cout << "ile liczb sortujemy?  ";
    cin>>n;
    int dane[n];
    for(int i=0;i<n;i++)
    cin>>dane[i];
    cout<<"dane przed sortowaniem"<<endl;
    for(i=0; i<n; i++)
          cout << " " << dane[i];

  for(j=n-2; j>=0; j--)          //start sortowania
  {
    d=dane[j];
    i=j+1;
    while((i<n)&&(d>dane[i]))
    {
      dane[i-1]=dane[i];
      i++;
    }
    dane[i-1]=d;
  }                                //koniec sortowania

  cout <<endl << "dane po sortowaniu"<<endl;
  for(i=0; i<n; i++)
      cout << " " << dane[i];
  return 0;
}

```

Rys. 49.4. Przykładowy program sortujący metodą „przez wstawianie"

W informatyce zazwyczaj nie ma jednego rozwiązania problemu programistycznego. Programy z rozdziału to tylko niektóre z możliwych. Może uda ci się opracować własne rozwiązania?

ZADANIA DO WYKONANIA

1. Dokonaj analizy programu sortującego metodą bąbelkową (rys. 49.2.). Omów działanie funkcji sortującej. Wskaż miejsca w algorytmie, które realizują poszczególne instrukcje.
2. Wykonaj symulację działania programu sortującego metodą bąbelkową (rys. 49.2.) dla trzech zmiennych [4,9,3], zapisując poszczególne działania i wartości zmiennych w odpowiednio zaprojektowanej tabeli. Wzoruj się na tabeli z rozdziałów 27. lub 48.
3. Dokonaj analizy programu sortującego metodą „przez wstawianie" (rys. 49.4.). Omów działanie głównej pętli sortowania. Wskaż miejsca w algorytmie, które realizują poszczególne instrukcje.
4. Wykonaj symulację działania programu sortującego metodą „przez wstawianie" (rys. 49.4.) dla trzech zmiennych [4,9,3], zapisując poszczególne działania i wartości zmiennych w odpowiednio zaprojektowanej tabeli. Wzoruj się na tabeli z rozdziałów 27. lub 48.

5*. Ułóż samodzielnie program sortujący metodą bąbelkową. Omów jego działanie i swój pomysł na realizację algorytmu.

6*. Ułóż samodzielnie program sortujący metodą „przez wstawianie". Omów jego działanie i swój pomysł na realizację algorytmu.

PODSUMOWANIE LEKCJI

- Funkcja `swap` wykonuje zamianę wartości zapamiętanych pod nazwami zmiennych `swap (a,b);`. ◂ s. 299
- Część główną programu (sortującą) możesz umieścić **w pętli** lub zapisać w funkcji wywoływanej odpowiednią liczbę razy. ◂ s. 298
- W programach sortowania metodami bąbelkową i „przez wstawianie" możesz wykorzystać pętle, ponieważ są to zazwyczaj rozwiązania **iteracyjne**, polegające na wielokrotnym powtarzaniu operacji na elementach sortowanych. ◂ s. 299

50. Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu

NA TEJ LEKCJI:

- poznasz ciąg Fibonacciego;
- ułożysz algorytm generujący kolejne liczby ciągu Fibonacciego, a na jego podstawie ułożysz program.

50.1. Boski wzór, czyli poznajemy własności ciągu Fibonacciego

Leonardo Fibonacci (czyt. fibonaczczi) urodził się w 1175 roku. Dlaczego więc zajmujemy się jego dokonaniami na informatyce w wieku XXI? Domyślasz się, że musiał stworzyć coś ważnego. I tak też było, chociaż wiele jego prac zapewne zaginęło, ponieważ w tamtym czasie nie znano jeszcze druku. Prace naukowe musiały być przepisywane ręcznie, dlatego nie powstawało zbyt wiele ich kopii. To właśnie Fibonacciemu zawdzięczamy przeniesienie na teren Europy arabskich i azjatyckich systemów matematycznych. Trudno uwierzyć, ale były to między innymi liczby ujemne, pojęcie zera, zapis liczb w systemie pozycyjnym – dziesiętnych, którym posługujemy się na co dzień, równanie liniowe i kwadratowe, a także zastosowania algebry w geometrii. Jednak najbardziej znanym osiągnięciem Fibonacciego jest jego „boski wzór" (rys. 50.1.).

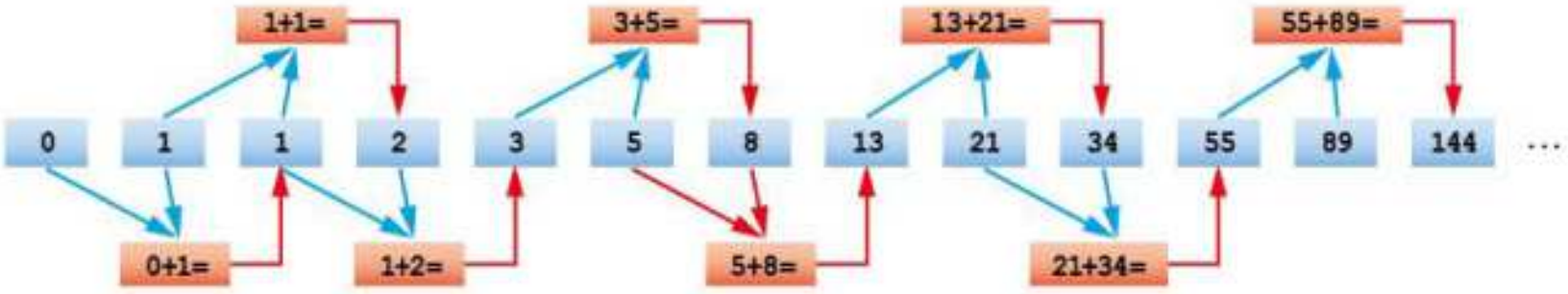
$$F_n = \begin{cases} 0 & \text{dla } n=0 \\ 1 & \text{dla } n=1 \\ F_{n-1} + F_{n-2} & \text{dla } n>1 \end{cases}$$

Rys. 50.1. Wzór opisujący ciąg Fibonacciego

ciąg Fibonacciego ▸ Ze wzoru łatwo odczytać, że pierwsze dwa wyrazy **ciągu Fibonacciego** mają wartość 1, a każdy następny jest sumą dwóch poprzednich (rys. 50.2.). 10 pierwszych wyrazów ciągu Fibonacciego przedstawiono w tabeli (tab. 50.1.)

F_0	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9
1	1	2	3	5	8	13	21	34	55

Tab. 50.1. Pierwszych 10 wyrazów ciągu Fibonacciego

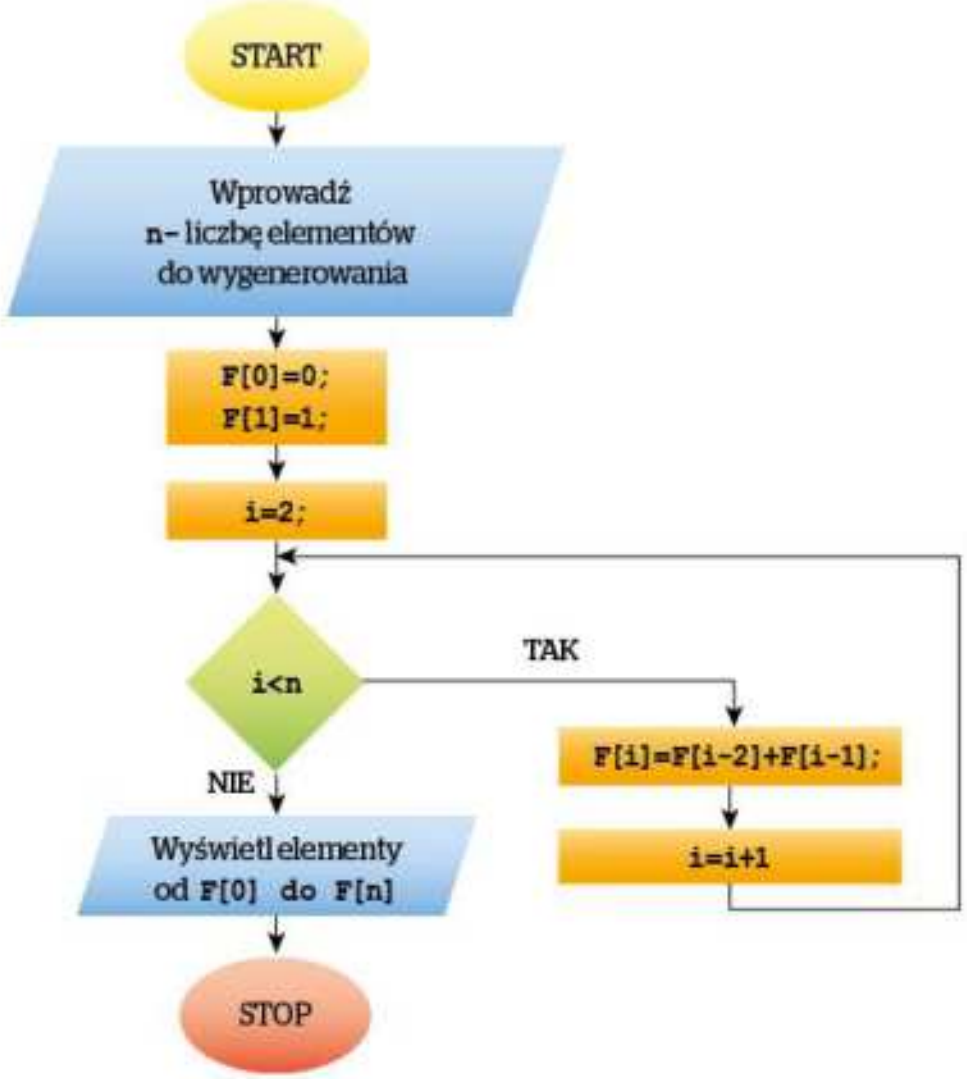


Rys. 50.2. Proces obliczania kolejnych wyrazów ciągu Fibonacciego

O związkach wyrazów ciągu Fibonacciego z naturą, np. budową roślin i organizmów, przeczytaj w sprawdzonych źródłach. Zrozumiesz wtedy, dlaczego nazwano go „boskim wzorem".

50.2. Ciąg Fibonacciego współcześnie, czyli układamy algorytm generujący jego kolejne elementy

Na podstawie wzoru (rys. 50.1.) można ułożyć algorytm, na podstawie którego utworzymy kod programu w C++. Spróbuj narysować go samodzielnie. Zwróć uwagę na wartość pierwszych dwóch wyrazów. Załóż, że liczby ciągu wypełnią tablicę jednowymiarową o rozmiarze *n*. Daną wejściową powinna być liczba elementów do obliczenia, a wyjściową – tablica z elementami ciągu Fibonacciego obliczonymi według wzoru (rys. 50.1.). Jeśli ułożenie algorytmu sprawi ci trudność, przeanalizuj algorytm przykładowy (rys. 50.3.).



Rys. 50.3. Przykładowy algorytm obliczający *n* wyrazów ciągu Fibonacciego

Sprawdźmy działanie algorytmu dla *n=5* (tab. 50.2.). Zauważ, że pierwszy element tablicy ma identyfikator 0, a więc ostatni zajmie w tablicy miejsce 4. Stąd także warunek *i<n*. Jest to także zgodne z zasadą numerowania pozycji w tablicy w języku C++ .

Operacja algorytmu	Wynik operacji	Stan tablicy danych
wprowadzenie n	n=5	F[...]
F[0]=0; F[1]=1;	F[0]=0; F[1]=1;	F[0,1,...]
i=2;	n=5, i=2	F[0,1,...]
i<n	2<5 TAK	F[0,1,...]
F[i]=F[i−2]+F[i−1];	F[2]=F[0]+F[1]=0+1=1	F[0,1,1...]
i=i+1;	i=2+1=3	F[0,1,1...]
i<n	3<5 TAK	F[0,1,1...]
F[i]=F[i−2]+F[i−1];	F[3]=F[1]+F[1]=1+1=2	F[0,1,1,2...]
i=i+1;	i=3+1=4	F[0,1,1,2...]
i<n	4<5 TAK	F[0,1,1,2...]
F[i]=F[i−2]+F[i−1];	F[4]=F[2]+F[3]=1+2=3	F[0,1,1,2,3...]
i=i+1;	i=4+1=5	F[0,1,1,2,3...]
i<n	5<5 NIE	F[0,1,1,2,3...]
Wyświetlenie n elementów F[]	0 1 1 2 3	
Stop		

Tab. 50.2. Weryfikacja działania algorytmu obliczającego kolejne wyrazy ciągu Fibonacciego dla *n*=5

50.3. Fibonacci w komputerze, czyli układamy program obliczający kolejne wyrazy ciągu

obliczanie wyrazów ciągu przez iterację

Nasz problem informatyczny możemy rozwiązać, stosując metodę iteracyjną. Obliczenia powtarzają się bowiem pewną określoną liczbę razy. Prześledź ponownie algorytm i zastanów się, którą pętlę zastosować. Przetestuj działanie programu dla *n=10*. Wyniki powinny być zgodne z tabelą 50.1. Jeśli napotkasz problem, przeanalizuj przykładowy program (rys. 50.4.).

```cpp
#include <iostream>
#include <cstdlib>

using namespace std;
int i, n;

int main()
{
    cout << "ile wyrazów ciągu Fibonacciego obliczyc? ";
    cin>>n;
    int F[n];
    F[0]=0;                    //dwa pierwsze wyrazy
    F[1]=1;                    //ciągu Fibonacciego

  for (i=2; i<n; i++)          //obliczanie wyrazów ciągu
      F[i]=F[i-2]+F[i-1];      //Fibonacciego

  cout <<n<<" wyrazów ciągu Fibonacciego"<<endl;
  for(i=0; i<n; i++)
      cout << " " << F[i];
  return 0;
}
```

```
ile wyrazów ciągu Fibonacciego obliczyc? 10
10 wyrazów ciągu Fibonacciego
 0 1 1 2 3 5 8 13 21 34
```

Rys. 50.4. Przykładowy program obliczający *n* wyrazów ciągu Fibonacciego

ZADANIA DO WYKONANIA

1. Wykonaj prezentację opisującą postać Fibonacciego. Zwróć uwagę na jego pochodzenie i podróże. Opisz ich wpływ na jego dzieła.
2. Znajdź w wiarygodnych źródłach graficzne reprezentacje „boskiego wzoru", np. budowę niektórych liści. Zaprezentuj je w dokumencie tekstowym oraz opatrz stosownym opisem.
3. Przeanalizuj ponownie algorytm i przykładowy program. Zastąp pętlę `for` pętlą `while`. Przetestuj poprawność działania programu. Uzasadnij zastosowane rozwiązanie.

4*. Ile razy zostanie wykonana pętla `for` w przykładowym programie dla n=21? Uzasadnij swoją odpowiedź.

PODSUMOWANIE LEKCJI

- **Ciąg Fibonacciego** to ciąg liczb, w których pierwszy ma wartość 0, następny 1, a kolejne oblicza się ze wzoru F[i–2]+F[i–1]. Oznacza to, że każdy następny wyraz ciągu jest sumą dwóch poprzednich. ◂ s. 302
- Program obliczania kolejnych wyrazów ciągu Fibonacciego jest oparty na metodzie iteracyjnej. ◂ s. 304

51. Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania

NA TEJ LEKCJI:

- dowiesz się, czym jest wyszukiwanie binarne;
- nauczysz się stosować wyszukiwanie binarne podczas porządkowania zbioru.

Główną ideą wyszukiwania binarnego jest zawężanie aktualnego zakresu możliwych odpowiedzi, które możemy uzyskać. Wykorzystujemy do tego wydajny algorytm znajdowania elementu z uporządkowanej listy elementów. Działa on poprzez wielokrotne dzielenie na pół fragmentu listy, który może zawierać szukany element, dopóki nie ograniczy jej tylko do jednej pozycji.

51.1. Wyszukiwanie binarne

wyszukiwanie binarne

Wyszukiwanie binarne realizujemy za pomocą algorytmu opierającego się na metodzie zwanej „dziel i zwyciężaj". Działanie tego algorytmu da nam odpowiedź, czy szukany element znajduje się w uporządkowanej tablicy, a jeśli tak, od razu poda nam jego indeks.

Jeżeli będziemy mieli do przeszukania tablicę składającą się z 1000 elementów, stosując wyszukiwanie binarne, wykonamy jedynie 10 kroków ($\log_2 1000 = 10$).

Algorytm ten ma złożoność obliczeniową rzędu O(log). Oznacza to, że ta metoda jest znacznie szybsza niż algorytm przeszukiwania linowego, w którym należałoby wykonać aż 1000 kroków. Tablica dzielona jest w nim na mniejsze podtablice tak długo, aż zostanie wyszukana (lub nie, jeżeli szukany element nie istnieje) pozycja wskazanego przez nas elementu.

Przykład

Przeanalizujmy zbiór 10 uporządkowanych liczb: 1, 2, 3, 5, 7, 9, 11, 13, 17, 19.

Potrzebujemy trzech pomocniczych zmiennych:
l – przechowuje numer lewego krańca tablicy,
s – przechowuje numer środkowego elementu tablicy,
p – przechowuje numer prawego krańca tablicy.

Do zmiennych przypisujemy numery elementów tablicy:
$l = 0$
$p = 9$
$s = (l + p)/2$

numer elementu	$l = 0$				$s = 4$					$p = 9$
wartość elementu	1	2	3	5	7	9	11	13	17	19

Sprawdzamy teraz, czy szukany element znajduje się na pozycji *s*. Jeżeli tak, to oznacza, że znaleźliśmy liczbę, której szukaliśmy – w przeciwnym wypadku sprawdzamy, czy szukana przez nas liczba jest mniejsza, czy większa niż liczba, która znajduje się na środkowej pozycji.

Jeżeli szukana liczba jest mniejsza, to oznacza, że musimy przeszukać lewą część naszej tablicy, zmieniając wartość zmiennej *p*:

$p = s - 1$
$p = 4–1$
$p = 3$

numer elementu	$l = 0$			$p = 3$	$s = 4$					
wartość elementu	1	2	3	5	7	9	11	13	17	19

W przypadku, gdy szukana przez nas liczba jest większa, to przeszukujemy prawą część tablicy, zmieniając wartość zmiennej *l*:

$l = s + 1$
$l = 4 + 1$
$l = 5$

numer elementu					$s = 4$	$l = 5$				$p = 9$
wartość elementu	1	2	3	5	7	9	11	13	17	19

Przykład

Z podanego w poprzednim przykładzie zbioru 10 liczb znajdźmy konkretną liczbę. Niech będzie to liczba 17.

nr elementu	$l = 0$				$s = 4$					$p = 9$
wartość elementu	1	2	3	5	7	9	11	13	17	19

$l = 0$
$p = 9$
$s = (l + p)/2$
$s = 4$

Na pozycji 4 mamy wartość 7, $17 > 7$ (*s* nie jest szukanym elementem). Przechodzimy do prawej części naszej tablicy i ją przeszukujemy.

nr elementu					$s = 4$	$l = 5$				$p = 9$
wartość elementu	1	2	3	5	7	9	11	13	17	19

$l = s + 1$
$l = 4 + 1$
$l = 5$

Szukamy nowego *s*.

$l = 5$
$p = 9$
$s = (l + p)/2$
$s = (5 + 9)/2$
$s = 7$

Na pozycji 7 mamy wartość 13, 17 > 13 (*s* nie jest szukanym elementem). Przechodzimy do prawej części naszej tablicy i ją przeszukujemy.

nr elementu						$l = 5$		$s = 7$		$p = 9$
wartość elementu	1	2	3	5	7	9	11	13	17	19

$l = s + 1$
$l = 7 + 1$
$l = 8$

Szukamy nowego *s*.

$l = 8$
$p = 9$
$s = (l + p)/2$
$s = (8 + 9)/2$
$s = 8{,}5$ (po zaokrągleniu 8)

Na pozycji 8 mamy wartość 17, 17 = 17. Szukany element znajduje się na pozycji 8.

ZADANIA DO WYKONANIA

1. Wykorzystując rozwiązanie iteracyjne, napisz program w języku C++, który będzie szukał danego elementu tablicy uporządkowanej za pomocą wyszukiwania binarnego.
2. Wykorzystując rozwiązanie rekurencyjne, napisz program w języku C++, który będzie szukał danego elementu tablicy uporządkowanej za pomocą wyszukiwania binarnego.

PODSUMOWANIE LEKCJI

s. 306 ▸

- **Wyszukiwanie binarne** jest wydajnym algorytmem znajdowania elementu z uporządkowanej listy. Działa poprzez wielokrotne dzielenie na pół fragmentu listy, który może zawierać szukany element, dopóki nie ograniczy jej tylko do jednej pozycji.

52. Przecinając oś, czyli wyznaczanie miejsc zerowych funkcji metodą połowienia

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest miejsce zerowe funkcji;
- dowiesz się, czym jest metoda połowienia;
- nauczysz się za pomocą metody połowienia wyznaczać miejsce zerowe funkcji.

Metoda równego podziału, zwana również metodą połowienia, bisekcji lub połowienia przedziału, pomoże nam w znalezieniu miejsca zerowego funkcji. Z pomocą przyjdzie nam poznana w poprzednim rozdziale metoda binarnego wyszukiwania.

52.1. Definicja miejsca zerowego

Miejscem zerowym funkcji nazywamy taki argument, dla którego wartość funkcji jest równa 0, czyli $f(x) = 0$.

◂ miejsce zerowe funkcji

Na wykresie funkcji f miejscami zerowymi będą miejsca przecięcia wykresu funkcji z osią OX.

Funkcja $f(x) = x + 2$ ma tylko jedno miejsce zerowe. Możemy je bardzo łatwo obliczyć:

$x + 2 = 0$

$x = -2$

Miejscem zerowym funkcji $f(x) = x + 2$ jest przecięcie osi X w punkcie -2.

Znajdowanie miejsc zerowych ma ogromne znaczenie w matematyce i naukach jej pokrewnych – fizyce, astronomii czy technice. Mamy do dyspozycji dwie podstawowe metody wyznaczania miejsc zerowych:

- metoda analityczna, która jest bardzo pracochłonna i wymaga dobrej znajomości matematyki, zwłaszcza gdy funkcja jest skomplikowana. Niepodważalną zaletą tej metody jest to, że otrzymamy bardzo dokładne rozwiązanie;
- wyliczenie pierwiastka przybliżonego z założoną dokładnością – jest to metoda, która zaczęła zyskiwać na popularności w momencie rozwoju informatyki. Jej wadą jest to, że operujemy na przybliżeniach, a zaletą – prostota.

52.2. Metoda połowienia

Metoda połowienia, inaczej bisekcji, służy do wyznaczenia miejsca zerowego funkcji i polega na cyklicznym połowieniu z góry wybranego przedziału. Czynność tę wykonujemy dopóty, dopóki nie zostanie znalezione miejsce zerowe.

◂ metoda połowienia

Do tego celu wykorzystamy metodę wyszukiwania binarnego. Żebyśmy mogli to zrobić, nasza funkcja musi spełniać następujące warunki:

1. Funkcja $f(x)$ jest ciągła w przedziale domkniętym $[a; b]$.
2. Funkcja $f(x)$ jest ciągła i przyjmuje różne znaki na końcach przedziału $f(a)f(b) < 0$.

Dane wejściowe:

$f(x)$ – funkcja, której miejsce zerowe chcemy znaleźć
a, b – miejsca krańcowe funkcji
$a,b \in R$

Dane wejściowe:

x_0 – miejsce zerowe funkcji

Zmienne pomocnicze:

ε_0 – określa dokładność porównania z zerem
ε_x – określa dokładność wyznaczenia miejsca zerowego

Liczba kroków:

Krok 1: Wczytaj a i b

Krok 2: $f_a = f(a)$; $f_b = f(b)$

Krok 3: Jeżeli $f_a \cdot f_b > 0$, to wypisz „Funkcja nie spełnia warunków" i idź do Krok 9

Krok 4: Dopóki $|a - b| > \varepsilon_x$, wykonuj kroki od Krok 5 do Krok 7

Krok 5: $x_0 = (a + b)/2$; $f_0 = f(x_0)$

Krok 6: Jeżeli $|f_0| < \varepsilon_0$, to idź do Krok 8

Krok 7: Jeżeli $f_a \cdot f_0 < 0$, to $b = x_0$, w przeciwnym razie $a = x_0$; $f_a = f_0$ i idź do Krok 4

Krok 8: Wypisz x_0

Krok 9: Koniec

ZADANIA DO WYKONANIA

1. Napisz program w języku C++ do wyznaczenia miejsca zerowego funkcji $f(x)$. Wykorzystaj metodę połowienia. Rozwiązanie zadania ma być iteracyjne.
2. Napisz program w języku C++ do wyznaczenia miejsca zerowego funkcji $f(x)$, stosując rozwiązanie rekurencyjne. Wykorzystaj metodę połowienia.
3. Napisz program w języku Java do wyznaczenia miejsca zerowego funkcji $f(x)$. Wykorzystaj metodę połowienia. Rozwiązanie zadania ma być iteracyjne.
4. Napisz program w języku Java do wyznaczenia miejsca zerowego funkcji $f(x)$, stosując rozwiązanie rekurencyjne. Wykorzystaj metodę połowienia.

PODSUMOWANIE LEKCJI

s. 309 ▸ • **Miejscem zerowym funkcji** nazywamy taki argument x_0, wtedy i tylko wtedy, gdy $f(x) = 0$.

s. 309 ▸ • **Metoda połowienia**, inaczej bisekcji, służy do wyznaczenia miejsca zerowego funkcji i polega na cyklicznym połowieniu z góry wybranego przedziału.

53. Prawie robi różnicę, czyli obliczanie przybliżonej wartości pierwiastka kwadratowego

NA TEJ LEKCJI:

- poznasz algorytm Newtona-Raphsona;
- dowiesz się, w jaki sposób wyznacza się przybliżony pierwiastek kwadratowy.

Rozwiązując zadania, często natrafiamy na pierwiastki, których wartości możemy podać tylko w przybliżeniu. Jak dokładne będzie to przybliżenie, zależy tylko od nas. Aby ułatwić sobie obliczenia, możemy skorzystać z metody Newtona-Raphsona (czyt. niutona-rafsona), której działanie opiera się na założeniu, że bok kwadratu o polu równym liczbie, dla której wyznaczamy pierwiastek, równa się właśnie temu pierwiastkowi.

53.1. Algorytm Newtona-Raphsona

algorytm Newtona-Raphsona

Algorytm Newtona-Raphsona, którym posłużymy się do wyznaczenia wartości pierwiastka kwadratowego, polega na wyznaczeniu kolejnych przybliżeń tego pierwiastka.

Algorytm ten zakłada, że postawiony problem jest identyczny z wyznaczaniem długości boku kwadratu o znanym polu powierzchni.

Pole kwadratu wyraża się wzorem – $P = a \cdot a$, co po przekształceniu da nam $a = P/a$ i w konsekwencji $P/a - a = 0$.

W ten sposób możemy sprawdzić, czy bok o długości a jest bokiem kwadratu, którego pole jest równe P.

Na początku sprawdzimy, czy $a = P/2$. Jeżeli ta wartość nie da nam odpowiedzi, będziemy wyznaczać kolejną długość boku a. Czynność tę będziemy powtarzać tak długo, aż $a = P/a$.

Metoda Newtona-Raphsona zakłada, że kolejne przybliżenia długości boku a zostaną wyznaczone przez średnią arytmetyczną aktualnej długości boku a i długości drugiego boku P/a. Dzięki takiemu rozwiązaniu prostokąt o długościach boków a i P/a będzie coraz bardziej przypominał kwadrat.

W programowaniu musimy wyznaczyć dokładność wyniku – epsilon (ε), w związku z tym w kodzie nasz zapis zmienimy z:

$P/a - a = 0$
na:
$|P/a - a| < ep$

Przykład

Sprawdźmy dla P=11 i dokładności 0,5, czy długość dwóch boków prostokąta będzie się różniła o wartość mniejszą niż założony błąd bezwzględny (epsilon).

$a = 11/2 = 5{,}5$

$|5{,}5 - 2| > 0{,}5$

$a = (11/5{,}5 + 5{,}5)/2 = 7{,}5/2 = 3{,}75$

$|11/3{,}75\text{-}3{,}75| > 0{,}5$

$a = (11/3{,}75 + 3{,}75) = 3{,}341667$

$|11/3{,}75\text{-}3{,}341667| < 0{,}5$

Wynik to 3,341667.

Kod w języku C++

```
#include <iostream>
#include <math.h>
using namespace std;
int main() {
    float P,a;
    float eps = 0.0001;
        cin >> P;
    a = P/2;
        while(fabs(P/a-a)>eps)
    {
        a = (P/a+a)/2;
    }
    cout << a;
    return 0;
}
```

ZADANIA DO WYKONANIA

1. Napisz program w języku C++ wyznaczający pierwiastek kwadratowy z trzech liczb pierwszych, poczynając od 3. Wykorzystaj algorytm Newtona-Raphsona.
2. Napisz program w języku Java wyznaczający pierwiastek kwadratowy z dowolnej liczby. Wykorzystaj algorytm Newtona-Raphsona.
3. Napisz program w języku C++ wyznaczający pierwiastek kwadratowy z dowolnej liczby nieparzystej. Wykorzystaj algorytm Newtona-Raphsona.
4. Napisz program w języku C++ wyznaczający pierwiastek kwadratowy z dowolnej liczby parzystej. Wykorzystaj algorytm Newtona-Raphsona.

PODSUMOWANIE LEKCJI

s. 311

- **Algorytm Newtona-Raphsona** jest algorytmem iteracyjnym służącym do wyznaczania przybliżenia pierwiastka kwadratowego z danej liczby.

54. Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest potęgowanie;
- dowiesz się, czym jest potęgowanie szybkie;
- poznasz rozwiązanie iteracyjne na potęgowanie szybkie.

Potęgowanie liczb jest niezwykle proste – w przypadku małych liczb możemy to robić w pamięci. W informatyce algorytm potęgowania liczby znajdujemy w wielu programach. Problem napotkamy wtedy, gdy przyjdzie nam wykonywać tę czynność na dużych liczbach. Z pomocą przychodzi wtedy potęgowanie szybkie.

Algorytm potęgowania szybkiego jest bardzo dobrym rozwiązaniem, jeżeli musimy obliczyć całkowite potęgi dużych liczb. Praktycznie używa się go w implementacjach szyfrowania RSA, w których jest wykorzystywany przy obliczaniu reszty z dzielenia potęgi przez liczbę.

54.1. Potęga o wykładniku naturalnym

Potęgowanie liczb to działanie matematyczne polegające na wielokrotnym mnożeniu liczby przez samą siebie. Liczba, którą chcemy potęgować, nazywa się **podstawą**, a potęga zapisana w indeksie górnym, czyli liczba, która informuje, ile razy podstawa ma być przemnożona przez samą siebie, to **wykładnik**.

◂ potęgowanie

Inaczej mówiąc, potęgowanie jest skróconym zapisem mnożenia jednakowych czynników:

$a^n = a \cdot a \cdot a \cdot \ldots \cdot a$

$a^1 = 1$, np. $5^1 = 5$, $(-10)^1 = -10$

$a^0 = 1$ dla $a \neq 0$ (a jest dowolną liczbą różną od zera).

Uwaga!

0^0 jest symbolem nieokreślonym, czyli nie oznacza żadnej liczby – nigdy nie podnosimy zera do potęgi zerowej.

Jeżeli podniesiemy zero do jakiejkolwiek potęgi, zawsze otrzymamy zero.

$0^n = 0$ dla $n \neq 0$, np. $0^3 = 0$

Potęga liczby ujemnej, której wykładnik jest liczbą parzystą, zawsze jest liczbą nieujemną, czyli dodatnią lub zerem. Natomiast potęga liczby ujemnej, której wykładnik jest liczbą nieparzystą, zawsze jest liczbą ujemną.

$(-3)^4 = (-3) \cdot (-3) \cdot (-3) \cdot (-3) = 81$

$(-3)^3 = (-3) \cdot (-3) \cdot (-3) = -27$

54.2. Potęgowanie szybkie

potęgowanie szybkie

Potęgowanie szybkie to metoda pozwalająca na szybkie obliczenie potęgi o wykładniku naturalnym. Metoda ta wykorzystuje pośrednio dwójkową reprezentację wykładnika potęgi, a jej złożoność, wyrażona jako liczba wykonywanych mnożeń, wynosi $\Theta(\log n)$, a n oznacza wykładnik obliczanej potęgi.

Standardowe potęgowanie dla wyrażenia a^n potrzebuje aż $n - 1$ mnożeń, natomiast szybkie potęgowanie pozwala na wykonanie maksymalnie $2 \cdot log_2 n$ mnożeń, czyli dokonuje się to o wiele szybciej.

Przeanalizujmy podniesienie liczby 2 do potęgi 10:

$2^{10} = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 1024$

To samo działanie można zapisać w skróconej postaci:

$2^{10} = (2^5)^2 = (2 \cdot 2^4)^2 = (2 \cdot (2^2)^2)^2 = (2 \cdot (2 \cdot 2)^2)^2$

Przyglądając się dokładnie tej formie zapisu, zauważymy, że użyliśmy tu mnożenia tylko cztery razy. Gdybyśmy chcieli do potęgi podnieść miliard, to stosując tę metodę, wykonalibyśmy około 100 działań mnożenia.

Przykład

Rozwiązanie iteracyjne zapisane w języku C++

```
#include<iostream>
using namespace std;
long long ps(long long a, unsigned int n)
{
      long long wynik = 1;
      while(n>0)
      {
            if (n%2 == 1)
                  wynik *= a;
            a*= a;
            n/=2;
      }
      return wynik;
}
int main()
{
      unsigned int n;
      long long a;
```

```
        cout<<"Podaj podstawę: ";
        cin>>a;
        cout<<"Podaj wykładnik: ";
        cin>>n;
        cout<<a<<" do potęgi "<<n<<" wynosi "<<ps(a,
n);
        cin.get();
        return 0;
}
```

ZADANIA DO WYKONANIA

1. Napisz program w języku C++, który będzie obliczał dowolną liczbę z przedziału od 100 do 1000 podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji iteracyjnej.
2. Napisz program w języku Java, który będzie obliczał dowolną liczbę z przedziału od 10 do 100 podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji iteracyjnej.
3. Napisz program w języku Java, który będzie obliczał dowolną liczbę podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji iteracyjnej.
4. Napisz program w języku Java, który będzie obliczał dowolną liczbę podniesioną do dowolnej potęgi. Zastosuj potęgowanie szybkie w wersji iteracyjnej.

PODSUMOWANIE LEKCJI

- Potęgowanie liczb to działanie matematyczne polegające na wielokrotnym mnożeniu liczby przez samą siebie. Liczba, którą chcemy potęgować, nazywa się podstawą, a potęga zapisana w indeksie górnym, czyli liczba, która informuje, ile razy podstawa ma być przez samą siebie przemnożona, to wykładnik. ◂ s. 313
- Potęgowanie szybkie to metoda pozwalająca na szybkie obliczenie potęgi o wykładniku naturalnym. Metoda ta wykorzystuje pośrednio dwójkową reprezentację wykładnika potęgi, a jej złożoność, wyrażona jako liczba wykonywanych mnożeń, wynosi $\Theta(\log n)$, a n oznacza wykładnik obliczanej potęgi. ◂ s. 314

55. Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak inaczej obliczyć potęgę o wykładniku naturalnym

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest rekurencja;
- poznasz rozwiązanie rekurencyjne na potęgowanie szybkie.

Powtarzalność czynności i stosowanie tych samych procedur towarzyszy nam przez całe życie, dlatego wykorzystanie rekurencji do rozwiązywania problemów wydaje się najbardziej odpowiednie i pozwalające znacząco skrócić czas działania samego algorytmu.

55.1. Potęgowanie w wersji rekurencyjnej

rekurencja ▸ Na początek przypomnienie definicji rekurencji. Metoda ta, zwana jest również rekursją (ang. *recursion*, czyt. rekurszyn, z łac. *recurrere, czyt. rekurrere* – przybiec z powrotem), to odwoływanie się np. funkcji lub definicji do samej siebie.

Przykład

Potęgowanie rekurencyjne w języku C++

```
#include<stdio.h>
#include<iostream>

using namespace std;

int potega(int a,int n)
      {
      if (n==0)return 1;
      else
      return a=a*potega(a,--n);
      }
int main()
{
cout<<potega(3,3);
return 0;
}
```

Potęgowanie rekurencyjne w języku Java

```
import java.util.Scanner;
public class Potegowanie {

	public static int potega(int a, int n) {
		if (n == 0)
		return 1;
		return a * potega(a, --n);
	}

	public static void main(String[] args) {
		int a, n;

		Scanner sc = new Scanner(System.in);
		System.out.print("Podaj podstawe: ");
		a = sc.nextInt();
		System.out.print("Podaj wykladnik: ");
		n = sc.nextInt();

		System.out.println(a + "^" + n + " = " +
potega(a,n));
	}
}
```

55.2. Potęgowanie szybkie w wersji rekurencyjnej

Powróćmy do przykładu z rozdziału 54., w którym przedstawiono wersję iteracyjną potęgowania szybkiego. Tym razem zajmiemy się wersją rekurencyjną.

Przykład

Potęgowanie szybkie w wersji rekurencyjnej w języku C++

```
#include<iostream>

using namespace std;

long long ps(long long a, unsigned int n)
{
	if(n==0)
		return 1;

	if(n%2 == 1)
		return a * ps(a, n - 1);
	long long b = ps(a, n/2);
	return b * b;
}

int main()
{
```

```
    unsigned int n;
    long long a;

    cout<<"Podaj podstawę: ";
    cin>>a;
    cout<<"Podaj wykładnik: ";
    cin>>n;

    cout<<a<<" do potęgi "<<n<<" wynosi "<<ps(a, n);

    cin.get();
    return 0;
}
```

Rozważymy dokładniejsze oszacowanie czasu działania tego algorytmu. Pozostańmy przy liczeniu wyłącznie tego, ile wykonaliśmy mnożeń, co oczywiście zależy od tego, ile było wywołań rekurencyjnych (na każde wywołanie przypada jedna operacja mnożenia). Każde następne wywołanie ma wykładnik dwa razy mniejszy (jeśli n było parzyste) albo o jeden mniejszy (dla n nieparzystych) od poprzedniego. Ten drugi przypadek – przejście od n do n–1 – nie może się jednak zdarzyć dwa razy z rzędu. Jeśli n było nieparzyste, to n–1 będzie parzyste, a następne wywołanie podzieli wykładnik przez 2.

Widzimy tu, że każde jedno wywołanie lub dwa kolejne wywołania zmniejszają wykładnik n dwukrotnie. Wiemy również, że liczbę n można dzielić przez 2 dokładnie $\log_2 n$ razy, zanim osiągniemy 1.

W tym wypadku wywołań będzie co najmniej $\log_2 n$, a co najwyżej $2 \cdot \log_2 n$.

ZADANIA DO WYKONANIA

1. Napisz program w języku C++, który będzie obliczał dowolną liczbę z przedziału od 100 do 1000 podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji rekurencyjnej.
2. Napisz program w języku Java, który będzie obliczał dowolną liczbę z przedziału od 10 do 100 podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji rekurencyjnej.
3. Napisz program w języku Java, który będzie obliczał dowolną liczbę podniesioną do potęgi 10. Zastosuj potęgowanie szybkie w wersji rekurencyjnej.
4. Napisz program w języku Java, który będzie obliczał dowolną liczbę podniesioną do dowolnej potęgi. Zastosuj potęgowanie szybkie w wersji rekurencyjnej.

PODSUMOWANIE LEKCJI

s. 316 ▸

- Rekurencja, zwana również rekursją, to odwoływanie się na przykład funkcji lub definicji do samej siebie.

56. Suma binarna, czyli dodawanie w systemie binarnym

NA TEJ LEKCJI:

- przypomnisz sobie, czym jest system binarny;
- dowiesz się, jak zamienić liczbę zapisaną w systemie dziesiętnym na zapisaną w systemie binarnym;
- nauczysz się wykonywać działanie dodawania w systemie binarnym.

Na liczbach binarnych możemy wykonywać podstawowe operacje arytmetyczne w podobny sposób jak na liczbach dziesiętnych. Do tego celu wykorzystujemy zapis dwójkowy.

56.1. System dwójkowy zwany binarnym

W dwójkowym systemie liczbowym, nazywanym również systemem binarnym lub systemem pozycyjnym, podstawą jest liczba 2, a do zapisu liczb potrzebne są tylko dwie liczby – 0 i 1. Podobnie jak w systemie dziesiętnym każdą z liczb możemy wyrazić za pomocą dodawania kolejnych potęg pomnożonych przez liczbę $124 = 1 \cdot 10^2 + 2 \cdot 10^1 + 4 \cdot 10^0$.

◂ system binarny

Potęgi 2

$2^0 = 1$
$2^1 = 2$
$2^2 = 4$
$2^3 = 8$
$2^4 = 16$
$2^5 = 32$
$2^6 = 64$
$2^7 = 128$
$2^8 = 256$
$2^9 = 512$
$2^{10} = 1024$

Spróbujmy teraz zapisać naszą przykładową liczbę 124 w systemie binarnym:

$64 + 32 + 16 + 8 + 4 = 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0$

Mając przekształcony zapis liczby 124 w sumę iloczynów kolejnych potęg, możemy spisać kolejne mnożone jedynki i zera oraz połączyć je w całość – wówczas powstanie liczba 124 w zapisie binarnym: 1111100.

Przykład

Lista kroków przekształcenia liczby 10 na liczbę binarną:

Krok 1: Podziel liczbę całkowitą przez 2, zapisz resztę z dzielenia jako pierwszą cyfrę w postaci binarnej. Przypisz do liczby wynik z dzielenia.

Krok 2: Jeżeli wynik dzielenia jest większy od 0, to przejdź do pierwszego kroku.

56.2. Dodawanie dwójkowe

dodawanie dwójkowe

Na liczbach binarnych możemy dokonywać prostych działań algebraicznych, takich jak dodawanie, odejmowanie, mnożenie i dzielenie. W tym rozdziale zajmiemy się dodawaniem. Potrzebna mam będzie znajomość tabliczek logicznych. Na początek dodawanie.

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10

Tab. 56.1. Suma binarna

Przykład

Sumowanie liczb binarnych 11110 oraz 100

Liczby zapisujemy tak samo jak w sumowaniu liczb dziesiętnych – w tzw. słupku, czyli jedna pod drugą.

	1	1	1			
		1	1	1	1	0
+		0	0	1	0	0
	1	0	0	0	1	0

Rys. 56.1. Dodawanie binarne

Sumowanie zaczynamy od ostatniej kolumny, zapisując wynik pod kreską. Jeśli wynik dodawania jest dwucyfrowy (sprawdź w tabelce: 1 + 1 = 10), to pod kreską zapisujemy 0, a jeden przechodzi do następnej kolumny, czyli stosujemy przeniesienie identyczne jak w sumowaniu liczb dziesiętnych. Przeniesienie jest zaznaczone na czerwono. Jeżeli w którejś z liczb zabrakło cyfr, to uzupełniamy je zerami (tak jak w drugim wierszu naszego przykładu).

Przykład

Algorytm dodawania dwóch liczb binarnych (lista kroków)

Krok 1: Wczytaj dwie liczby binarne $a_k a_{k-1}...a_1 a_0$ i $b_k b_{k-1}...b_1 b_0$

Krok 2: Dla każdej pozycji oblicz bit sumy s_i i bit przeniesienia p_i

Krok 3: Kolejne bity od *i*-tego do *k*-tego oblicz według wzoru $s_i = (a_i + b_i + p_{i-1})$ mod 2 (pamiętaj, że bit przeniesienia pi będzie ustawiony na 1, jeśli co najmniej dwie zmienne spośród a_i, b_i, p_{i-1} również są ustawione na 1 – wynika to z tabeli dodawania binarnego)

Krok 4: Wypisz wynik sumowania

ZADANIA DO WYKONANIA

1. Napisz program w języku C++ przekształcający liczbę zapisaną w systemie dziesiętnym na system dwójkowy.
2. Napisz program w języku Java przekształcający liczbę zapisaną w systemie dziesiętnym na system dwójkowy.
3. Napisz program w języku C++ obliczający sumę dwóch liczb dwójkowych.
4. Napisz program w języku Java obliczający sumę dwóch liczb dwójkowych.

PODSUMOWANIE LEKCJI

- W dwójkowym systemie liczbowym, nazywanym również systemem binarnym lub systemem pozycyjnym, podstawą jest liczba 2, a do zapisu liczb są potrzebne tylko dwie liczby – 0 i 1. ◂ s. 319
- Na liczbach binarnych możemy dokonywać prostych działań algebraicznych, takich jak dodawanie, odejmowanie, mnożenie i dzielenie. ◂ s. 320

57. Różnica binarna, czyli odejmowanie w systemie binarnym

NA TEJ LEKCJI:

- nauczysz się wykonywać działanie odejmowania w systemie binarnym;
- dowiesz się, co to jest pożyczka do następnej pozycji;
- poznasz tabliczkę odejmowania w systemie dwójkowym.

Na liczbach binarnych możemy wykonywać działania odejmowania w podobny sposób jak na liczbach dziesiętnych. Wykorzystujemy do tego celu zapis dwójkowy. Musimy pamiętać, że jeszcze nie znamy pojęcia liczb ujemnych w zapisie dwójkowym, więc będziemy liczby mniejsze odejmować od większych.

57.1. Różnica binarna

różnica binarna ▸ Z odejmowaniem mamy do czynienia od najmłodszych klas szkoły podstawowej. Jest to działanie dwuargumentowe przyporządkowujące dwóm liczbom a i b taką liczbę c, że $a - b = c$. Mówimy również, że a odejmujemy od liczby b. Innymi słowy, odjąć liczbę b od liczby a to znaleźć taką liczbę c, że $a = b + c$. Wynik tego działania nazywamy różnicą.

Odejmowanie możemy wykonać również w systemie liczb binarnych. W tym przypadku, tak samo jak przy dodawaniu, musimy skorzystać z tabliczki odejmowania, która w systemie dwójkowym jest bardzo prosta:

0	-	0	=	0
0	-	1	=	1 i pożyczka do następnej pozycji
1	-	0	=	1
1	-	1	=	0

Tab. 57.1. Różnica binarna

Przykład

Odejmowanie liczb binarnych 1101110 oraz 1111

Liczby zapisujemy tak samo jak w odejmowaniu liczb dziesiętnych – w tzw. słupku, czyli jedna pod drugą.

Na razie zakładamy, że od liczy większej będziemy odejmować liczbę mniejszą – w przeciwnym razie musielibyśmy wprowadzić liczby ujemne.

```
  1  1  0  1  1  1  0
-          1  1  1  1
-------------------------
  1  0  1  1  1  1  1
```

Rys. 57.1. Odejmowanie binarne

Odejmowanie zaczynamy od ostatniej kolumny, zapisując wynik pod kreską. W tym przykładzie odjęcie zera od jeden da wynik jeden oraz pożyczkę do następnej kolumny. Pożyczka ta jest zapisana kolorem czerwonym.

```
                 1
  1  1  0  1  1  1  0
-          1  1  1  1
-------------------------
  1  0  1  1  1  1  1
```

Rys. 57.2. Zapisanie pożyczki w odejmowaniu binarnym

Przechodzimy do kolumny drugiej od końca. Wykonujemy działanie 1 – 1 = 0. Od tego wyniku odejmujemy pożyczkę, czyli 0 – 1 = 1, a pożyczkę zapisujemy nad kolumną trzecią.

```
                    1   1
    1   1   0   1   1   1   0
-               1   1   1   1
-------------------------------
    1   0   1   1   1   1   1
```

Rys. 57.3. Zapisanie pożyczki w trzeciej kolumnie po wykonaniu odejmowania w drugiej kolumnie

Według tej zasady wykonujemy dalsze odejmowanie, pamiętając o pożyczkach – jeżeli w dolnym wierszu zabraknie nam cyfr w krótszej liczbie, to uzupełniamy to zerami, tak jak w przykładzie.

```
        1   1   1   1   1
    1   1   0   1   1   1   0
-   0   0   0   1   1   1   1
-------------------------------
    1   0   1   1   1   1   1
```

Rys. 57.4. Wstawienie pożyczki przy dalszym odejmowaniu oraz uzupełnienie zerami krótszej liczby

Sprawdźmy, czy otrzymany wynik jest poprawny:

$(1101110)_2 = 1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 64 + 32 + 8 + 4 + 2 = (110)_{10}$

$(1111)_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 20 = 8 + 4 + 2 + 1 = (15)_{10}$

$(1011111)_2 = 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 64 + 16 + 8 + 4 + 2 + 1 = (95)_{10}$

110 – 15 = 95

ZADANIA DO WYKONANIA

1. Napisz algorytm – listę kroków – realizujący różnicę binarną, przyjmując, że będziesz odejmować mniejszą liczbę od większej.
2. Napisz program w języku C++ realizujący algorytm z zadania 1.
3. Napisz program w języku Java realizujący algorytm z zadania 1.

PODSUMOWANIE LEKCJI

- Odejmowanie to działanie dwuargumentowe przyporządkowujące dwóm liczbom *a* i *b* taką liczbę *c*, że $a - b = c$. Odejmowanie binarne (różnica binarna) jest wykonywane tak samo jak na liczbach w systemie dziesiętnym. Korzystamy przy tym z tabliczki odejmowania liczb binarnych. ◂ s. 322

58. Iloczyn binarny, czyli mnożenie w systemie binarnym

NA TEJ LEKCJI:

- nauczysz się wykonywać działanie mnożenia w systemie dwójkowym;
- poznasz tabliczkę mnożenia w systemie dwójkowym.

Zasada mnożenia w systemie dwójkowym jest identyczna jak w systemie dziesiętnym, z tym że działania wykonujemy tylko na liczbach 0 i 1.

58.1. Iloczyn binarny

iloczyn binarny ▸ Mnożenie jest dwuargumentowym działaniem arytmetycznym (przyporządkowanym dwóm liczbom, a i b), które daje liczbę c jako wynik, czyli iloczyn. Zapisujemy je w postaci $a \cdot b = c$. Liczba otrzymana w wyniku mnożenia liczb całkowitych dodatnich określa nam sumę, którą otrzymamy, dodając a razy liczbę b: $a \cdot b = b + b + b + \ldots + b$ (lub b razy a).

Mnożenie możemy wykonać również w systemie liczb binarnych. W tym przypadku, tak samo jak w dodawaniu i odejmowaniu, musimy skorzystać z tabeli mnożenia liczb binarnych, która w systemie dwójkowym jest bardzo prosta:

0	·	0	=	0
0	·	1	=	0
1	·	0	=	0
1	·	1	=	1

Tab. 58.1. Iloczyn binarny

Przykład

Mnożenie liczb binarnych 1101 przez 1011

Liczby zapisujemy tak samo jak w mnożeniu liczb dziesiętnych – w tzw. słupku, czyli jedna pod drugą.

```
        1  1  0  1
  ·     1  0  1  1
  ---------------
```

Rys. 58.1. Zapis mnożenia binarnego

Mnożenie binarne, tak samo jak w systemie liczb dziesiętnych, zaczynamy od ostatniej kolumny, zapisując poszczególne cyfry mnożnika w odpowiednich kolumnach. Znaczącą różnicą w porównaniu z mnożeniem w systemie dziesiętnym jest to, że w systemie binarnym wynik zawsze będzie jednocyfrowy. Czynność wykonujemy aż do wymnożenia każdej z kolumn.

				1	1	0	1
.				1	0	1	1
				1	1	0	1
			1	1	0	1	
		0	0	0	0		
	1	1	0	1			

Rys. 58.2. Wykonanie mnożenia w systemie binarnym

Zwróć uwagę na to, że mnożenie przez jeden to powtórzenie zapisu. Podobnie jak w mnożeniu liczb dziesiętnych dodajemy wszystkie liczby do siebie, z tym że w naszym przypadku musimy pamiętać o przeniesieniach. Puste kolumny uzupełniamy zerami.

				1	1	0	1
.				1	0	1	1
	0	0	0	1	1	0	1
	0	0	1	1	0	1	0
+	1	1	0	1	0	0	0
1	0	0	0	1	1	1	1

Rys. 58.3. Ostateczny zapis z wynikiem mnożenia binarnego

Zwróć uwagę, że został usunięty wiersz samych zer z części, w której będziemy wykonywali dodawanie. Takie posunięcie ułatwi nam sumowanie otrzymanych liczb.

Sprawdźmy, czy wykonane w przykładzie mnożenie dało poprawny wynik:

$(1101)_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 20 = 8 + 4 + 1 = (13)_{10}$

$(1011)_{10} = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 20 = 8 + 2 + 1 = (11)_{10}$

$(10001111)_2 = 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 20 =$

$128 + 8 + 4 + 2 + 1 = (143)_{10}$

$13 \cdot 11 = 143$

ZADANIA DO WYKONANIA

1. Napisz algorytm – listę kroków – realizujący iloczyn binarny.
2. Napisz program w języku C++ realizujący algorytm z zadania 1.
3. Napisz program w języku Java realizujący algorytm z zadania 1.

PODSUMOWANIE LEKCJI

s. 324 ▸ • Mnożenie jest dwuargumentowym działaniem arytmetycznym (przyporządkowanym dwóm liczbom, *a* i *b*), które daje liczbę *c* jako wynik.

s. 324 ▸ • Mnożenie możemy wykonać również w systemie liczb binarnych. W tym przypadku, tak samo jak w dodawaniu i odejmowaniu, musimy skorzystać z tabeli mnożenia liczb binarnych.

59. Iloraz binarny, czyli dzielenie w systemie binarnym

NA TEJ LEKCJI:

- nauczysz się wykonywać działanie dzielenia w systemie dwójkowym.

Dzielenie binarne jest najbardziej skomplikowane z czterech podstawowych działań arytmetycznych. Stworzono wiele algorytmów, które je wykonują. W tym rozdziale poznasz najprostszy, który wykorzystuje odejmowanie.

59.1. Iloraz binarny

iloraz binarny ▸ Dzielenie to dwuargumentowe działanie arytmetyczne przyporządkowujące dwóm liczbom *a* i *b* (*b* musi być różne od 0) taką liczbę całkowitą *c*, która pomnożona przez liczbę *b* da liczbę *a*.

Dzielenie wykonamy również w systemie liczb binarnych. Iloraz binarny jest jedynym działaniem arytmetycznym, które nie wykorzystuje tabeli działań na liczbach binarnych.

Przykład

Podzielmy liczbę binarną 1101 przez 10

	1	1	0	1	dzielna
	1	0			dzielnik przesunięty

Rys. 59.1. Zapis dzielenia binarnego

Przesuwamy dzielnik w lewo, aż zrówna się jego najstarszy różny od zera bit z niezerowym najstarszym bitem dzielnej. Tak jak w zapisie dzielenia w systemie liczb dziesiętnych, nad liczbami rysujemy kreseczkę.

Porównujemy teraz dzielną z dzielnikiem. Jeżeli dzielna będzie większa od dzielnika lub jemu równa, to odejmujemy od niej dzielnik. Nad kreską na ostatniej pozycji dzielnej zapisujemy liczbę 1. Jeżeli natomiast dzielna będzie mniejsza od dzielnika, to nie wykonujemy odejmowania, ale przesuwamy dzielnik o jedną pozycję w prawo i powtarzamy opisaną czynność. Jeśli w ogóle nie będziemy mogli odjąć dzielnika od dzielnej (np. jeżeli będziemy próbowali podzielić mniejszą liczbę przez większą), to wynik dzielenia wyniesie 0, a dzielna będzie miała wartość reszty z dzielenia.

			1		
		1	1	0	1
+		1	0		
=		0	1	0	1

Rys. 59.2. Wykonanie dzielenia w systemie binarnym

Otrzymaliśmy pierwszą liczbę w ilorazie. Dzielnik przesuwamy teraz o jeden bit w prawo i odejmujemy go od różnicy, którą otrzymaliśmy w wyniku odejmowania. Jeżeli odejmowanie jest możliwe, do ilorazu dopisujemy jedynkę w następnej kolumnie. Ponownie przesuwamy dzielnik o jeden bit i odejmujemy go od otrzymanej różnicy. Jeśli odejmowanie jest możliwe, to wstawiamy 1, a jeżeli nie jest możliwe, to wstawiamy 0 nad kreską. Operację tę wykonujemy do momentu, aż zapiszemy ostatnią wartość nad kreską. Pozostała dzielna będzie resztą z dzielenia.

			1	1	0
		1	1	0	1
-		1	0		
		0	1	0	1
-			1	0	
		0	0	0	1
-				1	0
		0	0	0	1

Rys. 59.3. Pełen zapis dzielenia binarnego

Naszym ilorazem jest liczba w systemie binarnym 110. Sprawdźmy, czy wykonane w przykładzie mnożenie dało poprawny wynik:

$(1101)_{10} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 8 + 4 + 1 = (13)_{10}$

$(10)_{10} = 1 \cdot 2^1 + 0 \cdot 2^0 = (2)_{10}$

$(110)_2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 4 + 2 = (6)_{10}$

Reszty: $(1)_{10} = (1)_2$

13 : 2 = 6 reszty 1

ZADANIA DO WYKONANIA

1. Napisz algorytm – listę kroków – realizujący iloraz binarny.
2. Napisz program w języku C++ realizujący algorytm z zadania 1.
3. Napisz program w języku Java realizujący algorytm z zadania 1.

PODSUMOWANIE LEKCJI

s. 326 ▸ • Dzielenie jest dwuargumentowym działaniem arytmetycznym, które dwóm liczbom *a* i *b* (*b* musi być różne od 0) przyporządkowuje taką liczbę całkowitą *c*, która pomnożona przez liczbę *b* da liczbę *a*.

s. 326 ▸ • Dzielenie wykonamy również w systemie liczb binarnych. Nie wykorzystujemy przy tym tabeli działań na liczbach binarnych.

60. Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby pi

NA TEJ LEKCJI:

- poznasz metodę Monte Carlo;
- dowiesz się, jak zastosować metodę Monte Carlo do obliczania przybliżonej wartości liczby pi.

Nazwa metody, którą poznamy w tym rozdziale, pochodzi od dzielnicy Monako – Monte Carlo, ponieważ wykorzystuje ona losowość w rozwiązywaniu problemów matematycznych, a Monte Carlo słynie z możliwości uprawiania w nim gier losowych.

60.1. Metoda Monte Carlo

metoda Monte Carlo

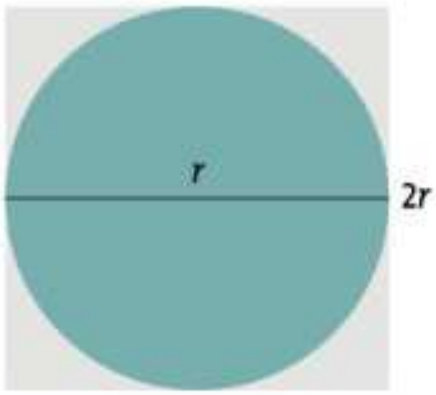
Metoda Monte Carlo jest numerycznym sposobem rozwiązywania problemów. Wykonujemy wiele obliczeń, opartych na losowych symulacjach, a na ich podstawie określamy wynik. Metoda ta została opracowana przez zespół znanego matematyka Johna von Neumanna (czytaj: dżona fon nojmana). Swój udział miał w tym polski matematyk Stanisław Ulam, który wywodził się z tzw. lwowskiej szkoły matematyków.

Metoda Monte Carlo służy do modelowania złożonych procesów, których rozwiązanie w sposób analityczny przysporzyłoby nam wielu problemów lub byłoby praktycznie niemożliwe. Symulacje, które wykonuje się metodą Monte Carlo, są kluczowym zagadnieniem związanym z losowością.

60.2. Metoda Monte Carlo i obliczanie liczby pi

Liczbę pi możemy obliczyć na wiele sposobów. Jednym z nich jest zastosowanie metody Monte Carlo.

Do obliczeń wykorzystamy koło o promieniu r wpisane w kwadrat o boku $2r$.



Rys. 60.1. Koło wpisane w kwadrat

Gdybyśmy chcieli obliczyć pole kwadratu, to skorzystalibyśmy ze wzoru $4r^2$, a pole koła wynosi πr^2 – oznacza to, że stosunek pola koła do pola kwadratu wynosi:

$$\frac{P_{\text{koła}}}{P_{\text{kwadratu}}} = \frac{\pi r^2}{4 r^2} = \frac{\pi}{4}$$

Dalej przekształćmy nasz wzór, aby obliczyć pi:

$$\pi = 4 \frac{P_{\text{koła}}}{P_{\text{kwadratu}}}$$

Aby obliczyć pole koła, musimy znać pi, ale istotą metody Monte Calro jest fakt, że możemy zastosować powyższą równość bez obliczania pola koła.

Wyobraźmy sobie, że rzucamy ołówek w narysowany kwadrat z umieszczonym w jego środku kołem. Jeżeli będziemy wykonywać tę czynność dostatecznie długo, w końcu nasze koło zostanie w całości pokryte kropkami, które zostawił upadający ołówek. Stosunek liczby kropek w naszym kole do wszystkich kropek w kwadracie będzie równy stosunkowi pola koła do pola kwadratu.

Przyjmijmy, że losujemy punkty o współrzędnych od $-2r$ do $2r$. Przeanalizujmy następnie stosunek liczby punktów, które zawierają się w kole o środku w punkcie <0,0> i promieniu r, do wszystkich wylosowanych punktów. Okaże się, że ten stosunek dąży w nieskończoność do stosunku pola tego koła do pola kwadratu o boku $2r$.

Cała metoda sprowadza się więc do tego, abyśmy losowali punkty, a zarazem sprawdzali, czy są one w kole. Liczby, które otrzymamy, podstawimy do naszego wzoru na obliczenie liczby pi. Losując bardzo dużą liczbę punktów, otrzymamy przybliżoną wartość liczby pi.

Lista kroków algorytmu obliczającego przybliżoną wartość liczby pi za pomocą metody Monte Carlo:

Krok 1: Wczytujemy liczbę losowanych punktów.

Krok 2: Zerujemy a – licznik punktów, które są zawarte w kole o promieniu r. Zmiennej i, która zlicza liczbę wylosowanych punktów, przypisujemy wartość początkową 1.

Krok 3: Generujemy dwie liczby losowe z zakresu od 0 do 1 – współrzędne x i y punktu.

Krok 4: Sprawdzamy, czy wylosowany punkt o współrzędnych x,y leży w kole o promieniu r.

Krok 5: Jeśli warunek z kroku 5 jest spełniony, zwiększamy o 1 zawartość zmiennej a, natomiast gdy warunek nie jest spełniony, przechodzimy do kroku 6;

Krok 6: Zwiększamy o 1 licznik wylosowanych punktów i.

Krok 7: Sprawdzamy, czy należy losować kolejne punkty. Jeśli nie, wracamy do kroku 3.

Krok 8: Obliczamy i zapamiętujemy w zmiennej pi przybliżoną wartość liczby pi.

Krok 9: Wypisujemy wynik.

ZADANIA DO WYKONANIA

1. Napisz algorytm blokowy wyliczający przybliżoną wartość liczby pi, posiłkując się algorytmem (lista kroków) zawartym w rozdziale.
2. Napisz program w języku C++ realizujący algorytm z zadania 1.
3. Napisz program w języku Java realizujący algorytm z zadania 1.

PODSUMOWANIE LEKCJI

s. 329 ▸

- Metoda Monte Carlo jest numerycznym sposobem rozwiązywania problemów. Wykonujemy wiele obliczeń opartych na losowych symulacjach, a na ich podstawie określamy wynik.

PODSUMOWANIE PRZED SPRAWDZIANEM

46. Powtarzanie pętli i wywoływanie siebie, czyli iteracja i rekurencja w algorytmach

- **Iteracja** (lac. *iteratio* – powtarzanie) to czynność wykonywana określoną liczbę razy lub do spełnienia danego warunku. W informatyce jest często stosowaną metodą układania programu. ◂ s. 285
- Z punktu widzenia informatyki **rekurencją** nazywamy odwoływanie się funkcji lub definicji do samej siebie. ◂ s. 285
- Ten sam problem programistyczny można często rozwiązać, stosując rekurencję lub iterację. Wybór metody należy dostosować do rozwiązywanego problemu lub algorytmu. ◂ s. 286

47. Sortowanie bąbelkowe, czyli każda liczba jest mniejsza od maksymalnej lub jej równa

- **Sortowanie bąbelkowe** polega na porównywaniu kolejno dwóch sąsiednich (występujących obok siebie w tablicy) liczb i zamianie ich miejscami w przypadku, gdy nie są uporządkowane, czyli liczba o niższym indeksie jest większa. ◂ s. 290
- **Algorytm sortowania bąbelkowego** powtarzany jest *n*-1 razy, a *n* jest liczbą sortowanych elementów. ◂ s. 290
- **Podczas sortowania bąbelkowego największa liczba** jest przesuwana na ostatnią pozycję, której nie zajmuje już większa od niej. ◂ s. 290
- **Litery** są sortowane według odpowiadających im liczb – **kodów ASCII**. ◂ s. 291

48. Przez wstawianie, czyli jeszcze o porządkowaniu liczb

- **Przykładem sortowania przez wstawianie** jest układanie kart według ustalonej w grze kolejności. Kolejno dobierane karty trafiają pomiędzy „młodszą" a „starszą". ◂ s. 293
- Jedna z liczb może **wystąpić dwukrotnie w tablicy danych**: na swoim nowym miejscu i na poprzednim. Powoduje to pozorne wykasowanie zmiennej, którą zastąpiła – zastąpioną liczbę przechowuje inna zmienna. ◂ s. 295

49. Komputer porządkuje liczby, czyli układamy programy sortujące

- Funkcja swap wykonuje zamianę wartości zapamiętanych pod nazwami zmiennych swap (a,b); . ◂ s. 299
- Część główną programu (sortującą) możesz umieścić **w pętli** lub zapisać w funkcji wywoływanej odpowiednią liczbę razy. ◂ s. 298
- W programach sortowania metodami bąbelkową i „przez wstawianie" możesz wykorzystać pętle, ponieważ są to zazwyczaj rozwiązania **iteracyjne**, polegające na wielokrotnym powtarzaniu operacji na elementach sortowanych. ◂ s. 299

50. Fibonacci i jego wzór, czyli generujemy kolejne liczby ciągu

- **Ciąg Fibonacciego** to ciąg liczb, w których pierwszy ma wartość 0, następny 1, a kolejne oblicza się ze wzoru F[i–2]+F[i–1]. Oznacza to, że każdy następny wyraz ciągu jest sumą dwóch poprzednich. ◂ s. 302
- Program obliczania kolejnych wyrazów ciągu Fibonacciego jest oparty na metodzie iteracyjnej. ◂ s. 304

51. Szukamy lidera, czyli porządkowanie zbioru metodą binarnego wyszukiwania

s. 306 ▸ • **Wyszukiwanie binarne** jest wydajnym algorytmem znajdowania elementu z uporządkowanej listy. Działa poprzez wielokrotne dzielenie na pół fragmentu listy, który może zawierać szukany element, dopóki nie ograniczy jej tylko do jednej pozycji.

52. Przecinając oś, czyli wyznaczanie miejsc zerowych funkcji metodą połowienia

s. 309 ▸ • **Miejscem zerowym funkcji** nazywamy taki argument $x = 0$, wtedy i tylko wtedy, gdy $f(x) = 0$.

s. 309 ▸ • **Metoda połowienia**, inaczej bisekcji, służy do wyznaczenia miejsca zerowego funkcji i polega na cyklicznym połowieniu z góry wybranego przedziału.

53. Prawie robi różnicę, czyli obliczanie przybliżonej wartości pierwiastka kwadratowego

s. 311 ▸ • **Algorytm Newtona-Raphsona** jest algorytmem iteracyjnym służącym do wyznaczania pierwiastka kwadratowego z danej liczby.

54. Szybkie potęgowanie liczb w wersji iteracyjnej, czyli jak obliczyć potęgę o wykładniku naturalnym

s. 313 ▸ • Potęgowanie liczb to działanie matematyczne polegające na wielokrotnym mnożeniu liczby przez samą siebie. Liczba, którą chcemy potęgować, nazywa się podstawą, a potęga zapisana w indeksie górnym, czyli liczba, która informuje, ile razy podstawa ma być przez samą siebie przemnożona, to wykładnik.

s. 314 ▸ • Potęgowanie szybkie to metoda pozwalająca na szybkie obliczenie potęgi o wykładniku naturalnym. Metoda ta wykorzystuje pośrednio dwójkową reprezentację wykładnika potęgi, a jej złożoność, wyrażona jako liczba wykonywanych mnożeń, wynosi $\Theta(\log n)$, a n oznacza wykładnik obliczanej potęgi.

55. Szybkie potęgowanie liczb w wersji rekurencyjnej, czyli jak inaczej obliczyć potęgę o wykładniku naturalnym

s. 316 ▸ • Rekurencja, zwana również rekursją, to odwoływanie się na przykład funkcji lub definicji do samej siebie.

56. Suma binarna, czyli dodawanie w systemie binarnym

s. 319 ▸ • W dwójkowym systemie liczbowym, nazywanym również systemem binarnym lub systemem pozycyjnym, podstawą jest liczba 2, a do zapisu liczb są potrzebne tylko dwie liczby – 0 i 1.

s. 320 ▸ • Na liczbach binarnych możemy dokonywać prostych działań algebraicznych, takich jak dodawanie, odejmowanie, mnożenie i dzielenie.

57. Różnica binarna, czyli odejmowanie w systemie binarnym

s. 322 ▸ • Odejmowanie to działanie dwuargumentowe przyporządkowujące dwóm liczbom a i b taką liczbę c, że $a - b = c$. Odejmowanie binarne (różnica binarna) jest wykonywane tak samo jak na liczbach w systemie dziesiętnym. Korzystamy przy tym z tabliczki odejmowania liczb binarnych.

58. Iloczyn binarny, czyli mnożenie w systemie binarnym

- Mnożenie jest dwuargumentowym działaniem arytmetycznym (przyporządkowanym dwóm liczbom, *a* i *b*), które daje liczbę *c* jako wynik. ◂ s. 324
- Mnożenie możemy wykonać również w systemie liczb binarnych. W tym przypadku, tak samo jak w dodawaniu i odejmowaniu, musimy skorzystać z tabeli mnożenia liczb binarnych. ◂ s. 324

59. Iloraz binarny, czyli dzielenie w systemie binarnym

- Dzielenie to dwuargumentowe działanie arytmetyczne przyporządkowujące dwóm liczbom *a* i *b* (*b* musi być różne od 0) taką liczbę całkowitą *c*, która pomnożona przez liczbę *b* da liczbę *a*. ◂ s. 326
- Dzielenie wykonamy również w systemie liczb binarnych. Nie wykorzystujemy przy tym tabeli działań na liczbach binarnych. ◂ s. 326

60. Metoda Monte Carlo, czyli jak obliczyć przybliżoną wartość liczby pi

- Metoda Monte Carlo jest numerycznym sposobem rozwiązywania problemów. Wykonujemy wiele obliczeń opartych na losowych symulacjach, a na ich podstawie określamy wynik. ◂ s. 329