

Wiesław Rychlicki



Programowanie w języku

Java

Zbiór zadań z (p)odpowiedziami



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Redaktor prowadzący: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn
Obarek, Pokoński, Pazdrijowski, Zaprucki

Wydawnictwo HELION
ul. Kościuszki 1c, 44-100 GLIWICE
tel. 32 231 22 19, 32 230 98 63
e-mail: helion@helion.pl
WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
<http://helion.pl/user/opinie?projaz>
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Materiały do książki można znaleźć pod adresem:
<ftp://ftp.helion.pl/przyklady/projaz.zip>

ISBN: 978-83-246-4319-6

Copyright © Helion 2012

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

Spis treści

| | |
|---|-----------|
| Od autora | 5 |
| Rozdział 1. Pierwszy krok — rozpoczynamy naukę programowania w języku Java | 7 |
| 1. Historia Javy i pierwsze zadania | 7 |
| 2. JDK, Notatnik i klasyczny przykład „Hello World” | 9 |
| 3. Znaki, tablice znaków i klasa String | 11 |
| 4. Klasa String — operacje na tekstach | 15 |
| 5. Tablica argumentów aplikacji | 18 |
| 6. Prawda czy fałsz — logiczny typ danych | 19 |
| 7. Liczby całkowite typu int i klasa Integer | 22 |
| 8. Inne typy liczb całkowitych w Javie | 25 |
| 9. Typy liczb zmiennoprzecinkowych | 27 |
| Rozdział 2. Drugi krok — operacje wejścia-wyjścia i instrukcje sterujące w Javie | 31 |
| 10. Wyświetlanie sformatowanych wyników w konsoli. Stałe i metody z klasy Math | 31 |
| 11. Wczytywanie danych — klasa Scanner | 34 |
| 12. Operacje na tekstach — klasy StringBuffer i StringBuilder | 36 |
| 13. Instrukcje warunkowe i instrukcja selekcji | 38 |
| 14. Instrukcja pętli typu do-while | 41 |
| 15. Instrukcja pętli typu while | 42 |
| 16. Instrukcja pętli typu for | 44 |
| Rozdział 3. Trzeci krok — budujemy własne metody i klasy | 47 |
| 17. Obsługa wyjątków | 47 |
| 18. Liczby pseudolosowe i tablice jednowymiarowe — budujemy metody statyczne | 48 |
| 19. Dokumentacja klasy | 53 |
| 20. Działania na ułamkach — budujemy klasę Fraction | 55 |
| 21. Klasa opakowująca Angle — miara kąta i funkcje trygonometryczne | 61 |
| 22. Liczby rzymskie i klasa Roman | 64 |
| 23. Trójmian kwadratowy i klasa KwadratPoly | 66 |
| 24. Liczby zespolone — budujemy klasę Complex | 69 |

| | |
|---|------------|
| Rozdział 4. Czwarty krok — pliki, tablice i macierze | 77 |
| 25. Operacje na plikach tekstowych | 77 |
| 26. Tablice jednowymiarowe i wielomiany | 80 |
| 27. Obliczenia statystyczne | 82 |
| 28. Tablice wielowymiarowe i macierze | 87 |
| 29. Obliczanie wartości funkcji, rekurencja i inne zadania | 92 |
| Rozdział 5. Rozwiązania zadań z rozdziału 1 | 97 |
| 1. Historia Javy i pierwsze zadania | 97 |
| 2. JDK, Notatnik i klasyczny przykład „Hello World” | 101 |
| 3. Znaki, tablice znaków i klasa String | 104 |
| 4. Klasa String — operacje na tekstach | 111 |
| 5. Tablica argumentów aplikacji | 117 |
| 6. Prawda czy fałsz — logiczny typ danych | 120 |
| 7. Liczby całkowite typu int i klasa Integer | 129 |
| 8. Inne typy liczb całkowitych w Javie | 135 |
| 9. Typy liczb zmiennoprzecinkowych | 140 |
| Rozdział 6. Rozwiązania zadań z rozdziału 2 | 147 |
| 10. Wyświetlanie sformatowanych wyników w konsoli. Stałe i metody z klasy Math | 147 |
| 11. Wczytywanie danych — klasa Scanner | 152 |
| 12. Operacje na tekstach — klasy StringBuffer i StringBuilder | 157 |
| 13. Instrukcje warunkowe i instrukcja selekcji | 162 |
| 14. Instrukcja pętli typu do-while | 179 |
| 15. Instrukcja pętli typu while | 187 |
| 16. Instrukcja pętli typu for | 193 |
| Rozdział 7. Rozwiązania zadań z rozdziału 3 | 201 |
| 17. Obsługa wyjątków | 201 |
| 18. Liczby pseudolosowe i tablice jednowymiarowe — budujemy metody statyczne | 206 |
| 19. Dokumentacja klasy | 221 |
| 20. Działania na ułamkach — budujemy klasę Fraction | 237 |
| 21. Klasa opakująca Angle — miara kąta i funkcje trygonometryczne | 269 |
| 22. Liczby rzymskie i klasa Roman | 290 |
| 23. Trójmian kwadratowy i klasa KwadratPoly | 301 |
| 24. Rozwiązania zadań — liczby zespolone | 318 |
| Rozdział 8. Rozwiązania zadań z rozdziału 4 | 347 |
| 25. Operacje na plikach tekstowych | 347 |
| 26. Tablice jednowymiarowe i wielomiany | 358 |
| 27. Obliczenia statystyczne | 370 |
| 28. Tablice wielowymiarowe i macierze | 385 |
| 29. Obliczanie wartości funkcji, rekurencja i inne zadania | 420 |

Rozdział 4.

Czwarty krok — pliki, tablice i macierze

25. Operacje na plikach tekstowych

Zadanie 25.1.

Korzystając z obiektów i metod klasy `FileWriter`, napisz program zapisujący do pliku tekstowego *tekst.txt* jeden wiersz tekstu: *Programowanie obiektowe*.

Zadanie 25.2.

Plik tekstowy *tekst.txt* zawiera jeden wiersz tekstu: *Programowanie obiektowe*. Korzystając z obiektu i metod klasy `FileWriter`, napisz program dopisujący do tego pliku w pierwszym wierszu tekst: *w języku Java*, a w kolejnym wierszu tekst: *jest bardzo interesujące*.



Wskazówka

W systemie Windows koniec wiersza w pliku tekstowym składa się z dwóch znaków *CR* (ang. *carriage return*, wartość ASCII 13, znak '\r') i *LF* (ang. *line feed* wartość ASCII 10, znak '\n'). W systemach UNIX i Linux końcem wiersza jest *LF*, a w systemie Mac OS — znak *CR*.

Zadanie 25.3.

Korzystając z obiektu i metod klasy `FileWriter`, napisz program obliczający i zapisujący w pliku *silnia.txt* wartości $n!$ (n silnia) dla $n = 1, 2, \dots, 12$. Każdy wynik zapisz w odrębnym wierszu, w postaci $12! = 479001600$.



Wskazówka

Przypomnijmy znaczenie symbolu $n!$: $1! = 1$, $2! = 1! \cdot 2 = 1 \cdot 2$, $3! = 2! \cdot 3 = 1 \cdot 2 \cdot 3$ itd.



Uwaga

W tym zadaniu i zadaniach podobnych starajmy się, aby plik wyjściowy nie zawierał na końcu pustego wiersza (jeśli obecność tego ostatniego wiersza nie jest zamierzona). Taki wiersz może nam sprawić różne niespodzianki, gdy plik będzie odczytywany.

Zadanie 25.4.

Napisz program zapisujący w pliku *pierwiastki.txt* wartości pierwiastków kwadratowych i sześciennych dla liczb naturalnych od 2 do 15. Każdy wiersz pliku powinien zawierać trzy liczby oddzielone znakami tabulatora — liczbę naturalną, pierwiastek kwadratowy z tej liczby i pierwiastek sześcienny. Pierwiastki podaj z precyzją do 8 miejsc po przecinku.



Wskazówka

Do formatowania wyników użyj metody `format()` z klasy `String`.

Zadanie 25.5.

Napisz program zapisujący w pliku tekstowym *sto.txt* sto liczb całkowitych wylosowanych z zakresu od 1 do 20. Liczby w pliku powinny być oddzielone odstępami.

Zadanie 25.6.

Napisz program zapisujący w pliku tekstowym *dane.txt* 50 par liczb. Każda para liczb powinna być umieszczona w odrębnym wierszu. Pierwsza liczba w parze powinna być rzeczywista, dodatnia i nie większa od 10 oraz podana z dokładnością do dwóch miejsc po przecinku, druga liczba powinna być całkowita i ma należeć do przedziału $\langle 2, 8 \rangle$. Liczby w wierszu oddzielamy odstępem.



Uwaga

Po rozwiązaniu zadań 25.1 – 25.6 będziemy mieli w bieżącym folderze pięć plików: *tekst.txt*, *silnia.txt*, *pierwiastki.txt*, *sto.txt* i *dane.txt*. Zawartość tych plików będziemy odczytywali w kolejnych zadaniach.

Zadanie 25.7.

Korzystając z obiektu i metod klasy `FileReader`, napisz program odczytujący zawartość pliku tekstowego i wyświetlający jego zawartość w konsoli.



Wskazówka

Bezparametrowa metoda `read()` z klasy `FileReader` odczytuje z pliku jeden znak i zwraca jego kod, liczbę typu `int` z zakresu od 0 do 65 535 lub liczbę `-1`, gdy nie można odczytać znaku. Metoda `ready()` zwraca wartość logiczną `true`, gdy z pliku można odczytać kolejny znak, i `false` w przeciwnym wypadku.

Zadanie 25.8.

Korzystając z obiektu i metod klasy `FileReader`, utwórz metodę (`readLine()`) odczytującą wiersz pliku tekstowego. Napisz program odczytujący i wyświetlający w konsoli wszystkie wiersze pliku tekstowego.



Klasa `FileReader` nie zawiera metody `readLine()`, więc musisz ją sam zbudować. Plik tekstowy otwórz przed wywołaniem metody, obiekt (klasy `FileReader`) skojarzony z plikiem przekaż jako parametr do metody `readLine()`, która przeczytany z pliku wiersz tekstu zwróci w postaci łańcucha znaków.

Zadanie 25.9.

Napisz program wyświetlający w konsoli kod źródłowy programu w języku Java wraz z numerami linii. Nazwę pliku (bez rozszerzenia) użytkownik powinien podawać z klawiatury. Jeśli w określonej lokalizacji nie ma wskazanego pliku, to program powinien wyświetlić odpowiedni komunikat.



Do wprowadzenia nazwy pliku z konsoli oraz do odczytania wierszy tekstu z pliku wykorzystaj obiekty klasy `BufferedReader` i metodę `readLine()` z tej klasy. Obecność pliku możesz sprawdzić, stosując metodę `exists()` z klasy `File`.

Zadanie 25.10.

Napisz program zapisujący do pliku kod źródłowy programu w języku Java wraz z numerami linii. Nazwę pliku (bez rozszerzenia) użytkownik powinien podawać z klawiatury. Jeśli w określonej lokalizacji nie ma wskazanego pliku, to program powinien wyświetlić odpowiedni komunikat. Nazwa pliku wyjściowego powinna być taka jak nazwa pliku źródłowego, rozszerzenie *java* zamienimy na rozszerzenie *txt*.



Do wprowadzenia nazwy pliku z konsoli oraz do odczytania wierszy tekstu z pliku wykorzystaj obiekty klasy `Scanner`. Do zapisania pliku wyjściowego użyj metod klasy `PrintWriter`.

Zadanie 25.11.

W pliku tekstowym wpisany jest ciąg liczb całkowitych oddzielonych odstępami. Napisz program, który odczyta i wyświetli w konsoli liczby z pliku oraz obliczy ich sumę.



Do testów możesz użyć pliku `sto.txt` (rozwiązanie zadania 25.5). Nie wykorzystuj jednak faktu, że znasz ilość liczb zapisanych w tym pliku.

Zadanie 25.12.

W pliku tekstowym wpisany jest ciąg liczb całkowitych oddzielonych odstępami. Napisz program, który znajdzie najmniejszą liczbę w tym pliku oraz obliczy, ile razy ta liczba w tym pliku występuje.



Do testów możesz użyć pliku *sto.txt* (rozwiązanie zadania 25.5). Nie wykorzystuj jednak faktu, że znasz ilość liczb zapisanych w pliku.

Zadanie 25.13.

W pliku tekstowym zapisane są w kolejnych wierszach pary liczb — liczba zmiennoprecinkowa i liczba całkowita. Napisz program, który odczytuje pary liczb z pliku, oblicza iloczyn każdej pary i sumuje iloczyny. Wynik obliczeń należy wypisać w konsoli i zapisać w ostatnim wierszu pliku z danymi.



Do testów możesz użyć pliku *dane.txt* (rozwiązanie zadania 25.6).

26. Tablice jednowymiarowe i wielomiany

Jednym z zastosowań tablic jednowymiarowych może być przechowywanie współczynników wielomianu.

Wielomian n -tego stopnia jednej zmiennej $w(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ ma $n+1$ współczynników, które możemy zapisać w tablicy `double[] a = {a0, a1, ..., an};`.

Zadanie 26.1.

Napisz program obliczający wartości wielomianu. Stopień wielomianu, współczynniki i kolejne wartości argumentu użytkownik będzie wprowadzał z klawiatury. Podanie argumentu $x = 0$ będzie sygnałem do zakończenia pracy programu.



Do obliczenia wartości wielomianu wykorzystaj *schemat Hornera*.

Zadanie 26.2.

Napisz program, który obliczy i zapisze w pliku tekstowym (*wielomian.txt*) tablicę wartości wielomianu $w(x) = 2x^3 + 5x^2 - x + 3$ w przedziale $\langle -2, 3 \rangle$ z krokiem $h = 0,125$. Plik powinien zawierać trzy wiersze z informacjami o rozwiązywanym zadaniu, według schematu: stopień wielomianu (pierwszy wiersz), współczynniki wielomianu oddzielone odstępami w drugim wierszu (zaczynając od wyrazu wolnego) i krańce

przedziału oraz krok w trzecim wierszu. W kolejnych wierszach umieścimy pary liczb (argument i wartość) oddzielone odstępem.

Zadanie 26.3.

Napisz program, który obliczy i wyświetli w konsoli sumę dwóch wielomianów.



Skorzystaj z metod dostępnych w klasie `Arrays`. Przyjmij, że wyświetlając wynik lub prezentując dane, wielomian $w(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ zapiszemy w postaci łańcucha znaków `"w = [a0, a1, ..., an]"`, gdzie a_0, a_1, \dots, a_n są liczbami wyrażającymi współczynniki wielomianu.

Zadanie 26.4.

Napisz program, który obliczy i wyświetli w konsoli różnicę dwóch wielomianów.



Zob. wskazówkę do zadania 26.3.

Zadanie 26.5.

Napisz program, który obliczy i wyświetli w konsoli iloczyn wielomianu przez liczbę.

Zadanie 26.6.

Napisz program, który obliczy i wyświetli w konsoli iloczyn dwóch wielomianów.

Zadanie 26.7.

Napisz program, który obliczy i wyświetli w konsoli pochodną wielomianu.

Zadanie 26.8.

Napisz program, który obliczy i wyświetli w konsoli całkę nieoznaczoną (funkcję pierwotną) wielomianu.

Zadanie 26.9.

Utwórz klasę `Polynomial` (plik `Polynomial.java`) umożliwiającą wykonywanie podstawowych działań na wielomianach zapisanych w postaci tablicy współczynników. Napisz aplikację prezentującą możliwości utworzonych metod i konstruktorów.

Zadanie 26.10.

Napisz program wykonujący dzielenie wielomianu

$$w(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

przez dwumian o postaci $(x-c)$.

Zadanie 26.11.

Dołącz do klasy `Polynomial` dwie metody — `division()` i `remainder()` — obliczające iloraz i resztę z dzielenia wielomianu reprezentowanego przez obiekt wywołujący metodę przez dwumian $(x-c)$, gdzie liczba c typu `double` jest parametrem wywołania metody. Napisz program demonstrujący działanie tych metod.

Zadanie 26.12.

Napisz program obliczający całkę oznaczoną $s = \int_a^b w(x)dx$ dla wielomianu

$$w(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

Wszystkie potrzebne dane użytkownik powinien wprowadzić z klawiatury — najpierw dane wielomianu (stopień i współczynniki), a później granice całkowania.



Do wyznaczenia funkcji pierwotnej użyj metody (`integral()`) opracowanej w rozwiązaniach zadań 26.8 i 26.9, a następnie skorzystaj ze wzoru $\int_a^b w(x)dx = F(b) - F(a)$.

Zadanie 26.13.

Mając dane wszystkie pierwiastki rzeczywiste wielomianu (x_1, x_2, \dots, x_n) , napisz program wyznaczający współczynniki wielomianu.

27. Obliczenia statystyczne

Zadania 27.1 – 27.19 wykonamy dla n -elementowej próbki zapisanej w tablicy:

```
double[] x = {1.35, 2.45, 2.05, 1.20, 2.15, 1.70, 1.45, 1.95, 2.00, 1.65, 1.65,
2.05, 1.75, 1.25, 2.25, 1.40};
```

Czytelnik może samodzielnie zmienić zestaw danych lub sposób ich pobierania przez program — wprowadzanie danych z klawiatury lub odczytywanie z pliku. Należy zwrócić uwagę na rozbieżności pomiędzy zakresem indeksów. W tablicach w języku Java indeksowanie rozpoczynamy od 0 i kończymy na indeksie o 1 mniejszym od rozmiaru tablicy, natomiast we wzorach statystyki opisowej indeksy wartości próbki będą w granicach od 1 do n .

W zadaniach bardzo często będziemy mieli do czynienia z obliczaniem sumy ciągu

liczb x_1, x_2, \dots, x_n (oznaczanej symbolem $\sum_{i=1}^n x_i$) zapisanego w tablicy `x[0], x[1], \dots,`

`x[n-1]`. Zrealizujemy to przy użyciu instrukcji pętli:

```
double suma = 0;
for(double xi: x)
    suma += xi;
```

lub:

```
double suma = 0;
for(int i = 0; i < n; ++i)
    suma += x[i];
```

Zadanie 27.1.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz najmniejszą i największą wartość w ciągu oraz rozstęp badanej cechy.



Rozstępem badanej cechy jest różnica pomiędzy wartością maksymalną i minimalną

$$R = x_{\max} - x_{\min}.$$

Zadanie 27.2.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz średnią arytmetyczną.



Średnią arytmetyczną liczb x_1, \dots, x_n nazywamy liczbę $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$.

Zadanie 27.3.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz średnią geometryczną.



Średnią geometryczną dodatnich liczb x_1, \dots, x_n nazywamy liczbę $\bar{g} = \sqrt[n]{\prod_{i=1}^n x_i}$.

Jeżeli wszystkie $x_i > 0$, to $\log \bar{g} = \frac{1}{n} \sum_{i=1}^n \log x_i$.

Zadanie 27.4.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz średnią harmoniczną.



Średnią harmoniczną różnych od zera liczb x_1, \dots, x_n nazywamy liczbę

$\bar{h} = \left(\frac{1}{n} \sum_{i=1}^n \frac{1}{x_i} \right)^{-1}$, gdy $\sum_{i=1}^n \frac{1}{x_i} \neq 0$ (odwrotność średniej arytmetycznej odwrotności tych liczb).

Zadanie 27.5.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz średnią potęgową rzędu r . Obliczenia wykonaj dla $r = 2$ i $r = 3$.



Wskazówka

Średnią potęgową rzędu r dodatnich liczb x_1, \dots, x_n nazywamy liczbę

$$\bar{p}^{(r)} = \sqrt[r]{\frac{1}{n} \sum_{i=1}^n x_i^r}. \text{ Dla } r = -1 \text{ otrzymujemy średnią harmoniczną } (\bar{p}^{(-1)} = \bar{h}),$$

a dla $r = 1$ średnią arytmetyczną ($\bar{p}^{(1)} = \bar{x}$).

Zadanie 27.6.

Dla podanej próbki n -elementowej x_1, \dots, x_n wyznacz medianę.



Wskazówka

Posortuj tablicę z danymi i wybierz element środkowy, gdy n jest nieparzyste, lub oblicz średnią arytmetyczną dwóch środkowych liczb, gdy n jest parzyste:

$$m_e = \begin{cases} x_{\frac{n+1}{2}}, & \text{gdy } n \text{ jest nieparzyste} \\ \frac{1}{2} \left(x_{\frac{n}{2}} + x_{\frac{n}{2}+1} \right), & \text{gdy } n \text{ jest parzyste} \end{cases}$$

Należy pamiętać o przesunięciu indeksu wynikającego z różnicy pomiędzy indeksami we wzorach a indeksami w tablicach.

Zadanie 27.7.

Wyznacz wartość modalną (dominantę) n -elementowej próbki x_1, \dots, x_n .



Wskazówka

Wartością modalną (dominantą, modą) próbki x_1, \dots, x_n o powtarzających się wartościach nazywamy najczęściej powtarzającą się wartość, o ile taka istnieje. Ponadto wartość ta nie może być wartością minimalną lub maksymalną.

Zadanie 27.8.

Oblicz wariancję n -elementowej próbki x_1, \dots, x_n . Wykorzystaj wszystkie niżej podane wzory i porównaj uzyskane wyniki.



Wskazówka

Wariancją s^2 (dyspersją) próbki x_1, \dots, x_n nazywamy średnią arytmetyczną kwadratów

$$\text{odchyleń wartości } x_i \text{ od średniej arytmetycznej } \bar{x} \text{ próbki: } s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2.$$

Można zastosować wzory równoważne $s^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2$ lub

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - a)^2 - (\bar{x} - a)^2, \text{ gdzie } a \text{ jest dowolną stałą.}$$

Zadanie 27.9.

Oblicz odchylenie standardowe n -elementowej próbki x_1, \dots, x_n .



Odchylenie standardowe s próbki x_1, \dots, x_n jest równe pierwiastkowi kwadratowemu z wariancji s^2 (zob. zadanie 27.8).

Zadanie 27.10.

Oblicz odchylenie przeciętne d próbki x_1, \dots, x_n od stałej a . Obliczenia wykonaj dla $a = 2$.



Odchyleniem przeciętnym d od stałej a próbki x_1, \dots, x_n nazywamy średnią arytmetyczną wartości bezwzględnych odchyleń poszczególnych wartości x_i od stałej a :

$$d = \frac{1}{n} \sum_{i=1}^n |x_i - a|.$$

Zadanie 27.11.

Oblicz odchylenie przeciętne d_1 próbki x_1, \dots, x_n od wartości średniej \bar{x} .

Zadanie 27.12.

Oblicz odchylenie przeciętne d_2 próbki x_1, \dots, x_n od mediany m_e .

Zadanie 27.13.

Wyznacz kwartył dolny Q_1 i kwartył górny Q_3 próbki x_1, \dots, x_n . Oblicz odchylenie ćwiartkowe Q .



Wartości uporządkowanej próbki dzielimy na dwie grupy: wartości mniejsze od mediany i medianę oraz medianę i wartości większe od mediany. Kwartyłem dolnym (Q_1) jest mediana pierwszej grupy, a górnym (Q_3) mediana drugiej grupy. Odchylenie ćwiartkowe Q obliczamy ze wzoru $Q = \frac{Q_3 - Q_1}{2}$.

Zadanie 27.14.

Oblicz moment zwykły m_l rzędu l próbki x_1, \dots, x_n . Obliczenia wykonaj dla $l = 2, 3$ i 4 .



$$\text{Wzór } m_l = \frac{1}{n} \sum_{i=1}^n x_i^l, \quad l \in N.$$

Zadanie 27.15.

Oblicz *moment centralny* M_l rzędu l próbki x_1, \dots, x_n . Obliczenia wykonaj dla $l = 2, 3$ i 4 .



Wzór $M_l = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^l$, $l \in N$. Z własności średniej arytmetycznej wynika, że $M_1 = 0$, natomiast M_2 jest wariancją.

Zadanie 27.16.

Oblicz *moment absolutny zwykły* a_l rzędu l próbki x_1, \dots, x_n . Obliczenia wykonaj dla $l = 2, 3$ i 4 .



Wzór $a_l = \frac{1}{n} \sum_{i=1}^n |x_i|^l$, $l \in N$.

Zadanie 27.17.

Oblicz *moment absolutny centralny* b_l rzędu l próbki x_1, \dots, x_n . Obliczenia wykonaj dla $l = 2, 3$ i 4 .



Wzór $b_l = \frac{1}{n} \sum_{i=1}^n |x_i - \bar{x}|^l$, $l \in N$. Absolutny moment centralny rzędu pierwszego jest odchyleniem przeciętnym od średniej arytmetycznej.

Zadanie 27.18.

Oblicz *współczynnik zmienności* v próbki x_1, \dots, x_n .



Wzór $v = \frac{s}{\bar{x}} \cdot 100\%$, gdzie s jest odchyleniem standardowym, a \bar{x} średnią arytmetyczną próbki.

Zadanie 27.19.

Oblicz *współczynnik nierównomierności* H próbki x_1, \dots, x_n .



Wzór $H = \frac{d_1}{\bar{x}} \cdot 100\%$, gdzie d_1 jest odchyleniem przeciętnym od średniej arytmetycznej \bar{x} .

W zadaniach 27.1 – 27.19 przedstawiono podstawowe wzory związane z obliczeniami statystyki opisowej dla pojedynczej próbki, zwykle nieprzekraczającej 30 elementów. Czytelnik może samodzielnie na podstawie rozwiązań tych zadań konstruować programy mające na celu rozwiązywanie problemów z zakresu statystyki.

Zadanie 27.20.

Na podstawie problemów zawartych w zadaniach 27.1 – 27.19 utwórz klasę `Stat` zawierającą metody statyczne do obliczeń statystycznych. Napisz aplikację konsolową pokazującą działanie wybranych metod z klasy `Stat`. Sporządź dokumentację tej klasy.

Zadanie 27.21.

Na podstawie problemów zawartych w zadaniach 27.1 – 27.19 utwórz klasę `Statpr` umożliwiającą utworzenie obiektu (próbki) zawierającego metody do rozwiązywania tych problemów. Napisz aplikację konsolową pokazującą działanie wybranych metod z klasy `Statpr`. Sporządź dokumentację tej klasy.

28. Tablice wielowymiarowe i macierze

Zadanie 28.1.

Utwórz dwuwymiarową tablicę liczb całkowitych o trzech wierszach. W pierwszym wierszu tablicy umieść liczby od 1 do 10, w drugim kwadraty tych liczb, a w trzecim sześciany liczb z pierwszego wiersza. Napisz program tworzący i wyświetlający tę tablicę w konsoli.

Zadanie 28.2.

W tablicy dwuwymiarowej nie wszystkie wiersze muszą mieć ten sam rozmiar. Napisz program, który utworzy tablicę liczb całkowitych o dziesięciu wierszach. Wypełnij tablicę kolejnymi liczbami naturalnymi, zaczynając od liczby 1. W pierwszym wierszu umieść jedną liczbę, w drugim dwie liczby, w trzecim trzy itd. — w dziesiątym dziesięć liczb. Oblicz sumy liczb w kolejnych wierszach i sumę wszystkich liczb zapisanych w tablicy. Wyświetl w konsoli tablicę liczb oraz obliczone sumy.

Zadanie 28.3.

Utwórz klasę `TInt`, która będzie zawierać metody statyczne `input()` i `print()` umożliwiające wprowadzanie danych z konsoli do tablicy lub wyświetlanie danych z tablicy w konsoli. Parametrem wywołania tych metod powinna być tablica liczb całkowitych jedno- lub dwuwymiarowa. Napisz program demonstrujący działanie tych metod.

Zadanie 28.4.

Utwórz w klasie `TInt` metodę statyczną `setRandom()`, która wypełni tablicę liczb całkowitych wartościami wylosowanymi z zakresu od 0 do n (liczba całkowita $n > 0$).

Tablicę oraz zakres wartości podaj jako parametry metody. Napisz program demonstrujący działanie tej metody.

Zadanie 28.5.

Dodaj do klasy `TInt` metodę statyczną `printf()` wyświetlającą tablicę liczb całkowitych w konsoli. Metoda ta powinna mieć dwa parametry: łańcuch formatujący i identyfikator tablicy. Napisz program demonstrujący działanie tej metody.

Zadanie 28.6.

Na podstawie zadań 28.3, 28.4 i 28.5 utwórz klasę `TDouble` z metodami statycznymi `input()`, `print()`, `printf()` i `setRandom()`, ułatwiającymi pobieranie i wypisywanie danych oraz losowe ustawianie wartości w jedno- i dwuwymiarowych tablicach liczb zmiennoprzecinkowych typu `double`. Napisz program pokazujący działanie wybranych metod z tej klasy.

Macierz jest uporządkowaną prostokątną tablicą liczb, dla której zdefiniowane są działania algebraiczne dodawania (odejmowania) i mnożenia:

- ◆ Dodawanie (odejmowanie) dwóch macierzy jest możliwe tylko dla macierzy o jednakowych liczbach kolumn (n) i wierszy (m). Suma (różnica) macierzy A i B jest macierzą C taką, że $c_{ij} = a_{ij} + b_{ij}$ ($c_{ij} = a_{ij} - b_{ij}$).
- ◆ Mnożenie macierzy jest możliwe, gdy liczba kolumn pierwszej macierzy (m) jest równa liczbie wierszy drugiej macierzy. Iloczyn macierzy A i B jest macierzą C taką, że $c_{ij} = \sum_{k=1}^m a_{ik} b_{kj}$. Mnożenie macierzy nie jest przemienne.
- ◆ Zawsze określone jest mnożenie macierzy przez liczbę λ , polegające na pomnożeniu każdego elementu macierzy przez tę liczbę ($c_{ij} = \lambda a_{ij}$).

Macierze możemy przedstawiać w programach jako tablice dwuwymiarowe. Należy pamiętać, że tablice są indeksowane od zera, a indeksy macierzy rozpoczynamy od jedynek.

Zadanie 28.7.

Utwórz metodę `sum()` dodającą dwie macierze zapisane w postaci tablic dwuwymiarowych. Napisz program demonstrujący działanie metody `sum()`.



Uwaga

W tym i kolejnych zadaniach do wprowadzania danych używaj metod z klasy `TInt` lub `TDouble`.

Zadanie 28.8.

Utwórz metodę `difference()` obliczającą różnicę dwóch macierzy zapisanych w postaci tablic dwuwymiarowych. Napisz program demonstrujący działanie metody `difference()`.

Zadanie 28.9.

Utwórz metodę `product()` obliczającą iloczyn dwóch macierzy zapisanych w postaci tablic dwuwymiarowych. Napisz program demonstrujący działanie metody `product()`.

Zadanie 28.10.

Utwórz metodę `product()` obliczającą iloczyn macierzy zapisanej w postaci tablicy dwuwymiarowych przez liczbę. Napisz program demonstrujący działanie metody `product()`.

Zadanie 28.11.

Zbuduj metodę `transp()` tworzącą macierz transponowaną z macierzy podanej jako parametr metody. Napisz program demonstrujący działanie metody `transp()`.



Macierz transponowana (przestawiona) powstaje z danej macierzy poprzez zamianę jej wierszy na kolumny i kolumn na wiersze.

Zadanie 28.12.

Utwórz metodę statyczną `toDouble()` konwertującą macierz o elementach całkowitych na macierz o elementach zmiennoprzecinkowych. Napisz program demonstrujący działanie tej metody. Dołącz ją do klasy `TInt`.

Zadanie 28.13.

Dołącz do klasy `TDouble` metodę statyczną o nazwie `valueOf()`, zwracającą tablicę (macierz) z elementami typu `double` o elementach odpowiadających elementom tablicy (macierzy) liczb całkowitych. Napisz program demonstrujący działanie tej metody.

Zadanie 28.14.

W klasie `TDouble` utwórz statyczną metodę `toInt()`, która będzie zwracać tablicę (macierz) o wartościach całkowitych, odpowiadających (zamiana przez rzutowanie) tablicy liczb zmiennoprzecinkowych podanej jako parametr. Napisz program demonstrujący działanie tej metody.

Zadanie 28.15.

W klasie `TInt` utwórz statyczną metodę `valueOf()`, która będzie budować tablicę (macierz) o wartościach całkowitych, odpowiadających (zamiana przez rzutowanie) tablicy liczb zmiennoprzecinkowych podanej jako parametr. Napisz program demonstrujący działanie tej metody.

Macierze, w których liczba wierszy jest równa liczbie kolumn, nazywamy *macierzami kwadratowymi*. Z macierzami kwadratowymi związany jest szereg pojęć, takich jak: *śląd macierzy*, *wyznacznik macierzy*, *macierz diagonalna*, *macierz trójkątna*, *macierz jednostkowa* i *macierz odwrotna*.

Zadanie 28.16.

Utwórz statyczną metodę `trace()` wyznaczającą ślad macierzy. Napisz program demonstrujący działanie tej metody.

Zadanie 28.17.

Utwórz statyczną metodę `getI()` zwracającą macierz jednostkową stopnia n (stopień podamy jako parametr wywołania metody). Napisz program demonstrujący działanie tej metody.

Zadanie 28.18.

Utwórz statyczną metodę `setI()` tworzącą z macierzy kwadratowej podanej jako parametr macierz jednostkową. Napisz program demonstrujący działanie tej metody.

Zadanie 28.19.

Utwórz metodę statyczną `det()` obliczającą wyznacznik macierzy kwadratowej. Do obliczenia wyznacznika użyj rozwinięcia Laplace'a. Napisz program demonstrujący działanie tej metody.

Zadanie 28.20.

Utwórz metodę statyczną `upperTriangular()` przekształcającą macierz kwadratową podaną jako parametr na macierz trójkątną górną. Napisz program demonstrujący działanie tej metody.

Zadanie 28.21.

Utwórz metodę statyczną `lowerTriangular()` przekształcającą macierz kwadratową podaną jako parametr na macierz trójkątną dolną. Napisz program demonstrujący działanie tej metody.

Zadanie 28.22.

Utwórz metodę statyczną `diagonal()` przekształcającą macierz kwadratową podaną jako parametr na macierz diagonalną. Napisz program demonstrujący działanie tej metody.

Zadanie 28.23.

Utwórz metodę statyczną `inverse()` obliczającą i zwracającą macierz odwrotną do macierzy kwadratowej podanej jako parametr. Napisz program demonstrujący działanie tej metody.

Zadanie 28.24.

Napisz program rozwiązujący układ n -równań liniowych z n niewiadomymi ($n < 10$). Układ równań rozwiąż, stosując metodę wyznaczników. Wszystkie niezbędne metody umieść w klasie programu.

Zadanie 28.25.

Napisz program rozwiązujący układ n -równań liniowych z n niewiadomymi. Układ równań rozwiąż, stosując rachunek macierzy: $A \cdot X = B$, $X = A^{-1} \cdot B$, gdzie A — macierz podstawowa układu, X — wektor niewiadomych, B — kolumna wyrazów wolnych. Wszystkie niezbędne metody umieść w klasie programu.

Zadanie 28.26.

Napisz program rozwiązujący układ n -równań liniowych z n niewiadomymi. Układ równań rozwiąż, stosując metodę eliminacji. Wszystkie niezbędne metody umieść w klasie programu.

Zadanie 28.27.

Utwórz klasę `Matrix`, która na podstawie tablic dwuwymiarowych, podanych wymiarów macierzy lub innych obiektów klasy `Matrix` (konstruktor kopiujący) będzie umożliwiać tworzenie obiektów reprezentujących macierze. Utwórz metody ułatwiające dostęp do elementów macierzy (pól obiektu), wprowadzanie i wyświetlanie danych oraz wypełnianie macierzy wartościami losowymi. Napisz program demonstrujący działanie metod klasy `Matrix`.

Zadanie 28.28.

Dołącz do klasy `Matrix` metody umożliwiające wykonywanie podstawowych działań na macierzach: dodawanie, odejmowanie i mnożenie macierzy oraz mnożenie macierzy przez skalar. Napisz program demonstrujący działania na macierzach.

Zadanie 28.29.

Dołącz do klasy `Matrix` metody umożliwiające przekształcanie macierzy kwadratowych na postać trójkątną lub diagonalną. Napisz program demonstrujący działanie tych metod.

Zadanie 28.30.

Dołącz do klasy `Matrix` metodę umożliwiającą obliczanie wyznacznika macierzy. Napisz program demonstrujący działanie tej metody.

Zadanie 28.31.

Dołącz do klasy `Matrix` metodę umożliwiającą obliczanie macierzy odwrotnej. Napisz program demonstrujący obliczanie macierzy odwrotnej. Sprawdź uzyskany wynik, wykonując odpowiednie mnożenie.

Zadanie 28.32.

Wykorzystując możliwości klasy `Matrix` i obliczenia na macierzach, rozwiąż układ n -równań liniowych z n niewiadomymi.

Zadanie 28.33.

Wykorzystując możliwości klasy `Matrix` i metodę wyznaczników, rozwiąż układ n -równań liniowych z n niewiadomymi.



Utwórz metodę pomocniczą `replaceCol()`, która w macierzy wywołującej tę metodę zastąpi wskazaną kolumnę przekazaną jako parametr (będzie to kolumna wyrazów wolnych).

29. Obliczanie wartości funkcji, rekurencja i inne zadania

Zadanie 29.1.

Napisz aplikację testującą działanie podanej metody dla różnych argumentów. Określ, co oblicza ta metoda. Nadaj jej odpowiednią nazwę.

```
static double f(double x) {
    return (x > 0)?x:-x;
}
```

Zadanie 29.2.

Określono dwie funkcje: $\min(x, y) = \begin{cases} x, & \text{gdy } x \leq y \\ y, & \text{gdy } x > y \end{cases}$ oraz $\max(x, y) = \begin{cases} x, & \text{gdy } x \geq y \\ y, & \text{gdy } x < y \end{cases}$.

Utwórz klasę `MinMax` z metodami `min()` i `max()` obliczającymi i zwracającymi wartości tych funkcji. Napisz aplikację pokazującą działanie tych metod.

Zadanie 29.3.

Określono funkcję $f(x) = \begin{cases} -1, & \text{gdy } x < 0 \\ 0, & \text{gdy } x = 0 \\ 1, & \text{gdy } x > 0 \end{cases}$. Co oblicza ta funkcja? Utwórz metodę ob-

liczającą wartość tej funkcji. Napisz aplikację pokazującą działanie zbudowanej metody.

Zadanie 29.4.

Określono funkcję $f(x) = \begin{cases} 0, & \text{gdy } x = 0 \\ \frac{|x|}{x}, & \text{gdy } x \neq 0 \end{cases}$. Co oblicza ta funkcja? Utwórz metodę ob-

liczającą wartość tej funkcji. Napisz aplikację pokazującą działanie zbudowanej metody.

Zadanie 29.5.

Napisz aplikację testującą wartości funkcji określonych wzorami:

$$\text{a) } f(x, y) = \frac{x + y + |x - y|}{2}$$

$$\text{b) } g(x, y) = \frac{x + y - |x - y|}{2}$$

$$\text{c) } h(x) = f(x, -x)$$

Co obliczają te funkcje? Utworzonym metodom nadaj odpowiednie nazwy.

Zadanie 29.6.

Napisz definicje metod `square()` i `cube()` obliczających kwadrat i sześcian liczby x . Zastosuj utworzone metody do obliczenia kwadratów i sześcianów liczb:

a) całkowitych od 1 do 15,

b) rzeczywistych od 1 do 3 z krokiem 0,25.

Zadanie 29.7.

Potęę o wykładniku całkowitym dodatnim określamy wzorem $a^n = \underbrace{a \cdot a \cdot \dots \cdot a}_n$. Utwórz metodę `adocen()` obliczającą a^n . Napisz aplikację pokazującą działanie tej metody.

Zadanie 29.8.

Potęę o wykładniku całkowitym nieujemnym możemy określić wzorem rekurencyjnym: $a^n = \begin{cases} 1, & \text{dla } n = 0 \\ a \cdot a^{n-1}, & \text{dla } n > 0 \end{cases}$. Napisz definicję metody rekurencyjnej `adocen()` obliczającej a^n oraz aplikację pokazującą działanie tej metody.

Zadanie 29.9.

Szybkie potęgowanie — inną wersję metody rekurencyjnej obliczającej a^n (szybszą ze względu na mniejszą liczbę wywołań rekurencyjnych i zredukowanie liczby mnożeń) możemy zrealizować na podstawie wzoru:

$$a^n = \begin{cases} 1 & \text{dla } n = 0 \\ a \cdot \left(a^{\frac{n-1}{2}}\right)^2 & \text{dla } n \text{ nieparzystego } (n > 0) \\ \left(a^{\frac{n}{2}}\right)^2 & \text{dla } n \text{ parzystego } (n > 0) \end{cases}$$

Utwórz metodę `adocen()` obliczającą a^n . Napisz aplikację pokazującą działanie tej metody.

Zadanie 29.10.

Potęę o podstawie a różnej od 0 i wykładniku całkowitym n możemy zdefiniować

$$\text{w następujący sposób: } a^n = \begin{cases} \underbrace{a \cdot a \cdot \dots \cdot a}_n, & \text{dla } n > 0 \\ 1, & \text{dla } n = 0 \\ \frac{1}{a^n}, & \text{dla } n < 0 \end{cases}.$$

Napisz definicję metody `power()` (prototyp: `double power(double a, int n)`) obliczającej a^n dla dowolnej liczby całkowitej oraz aplikację pokazującą działanie tej funkcji.



Wskazówka

Funkcję `adocen` (zob. zadanie 29.7, 29.8 lub 29.9) obliczającą a^n dla $n > 1$ wykorzystamy w funkcji `power` (do obliczania a^n lub $a^{-n} = \frac{1}{a^n}$ dla $n = 1, 2, 3, \dots$).

Zadanie 29.11.

Pierwiastek drugiego stopnia z liczby dodatniej a możemy obliczyć metodą iteracyjną na podstawie wzoru $x = \frac{1}{2} \left(x + \frac{a}{x} \right)$, przyjmując jako pierwsze przybliżenie $x = 1$. Obliczenia kontynuujemy do chwili, gdy różnica pomiędzy dwoma kolejnymi przybliżeniami pierwiastka będzie dostatecznie mała. Napisz metodę `sqrtd()` (ang. *square root*) obliczającą pierwiastek kwadratowy z podanej liczby dodatniej. Zbuduj aplikację pokazującą działanie metody `sqrtd()` i porównującą otrzymane wyniki z wynikami metody bibliotecznej `Math.sqrt()`.

Zadanie 29.12.

Pierwiastek trzeciego stopnia z liczby dodatniej a możemy obliczyć metodą iteracyjną na podstawie wzoru $x = \frac{1}{3} \left(2x + \frac{a}{x^2} \right)$, przyjmując jako pierwsze przybliżenie $x = 1$. Obliczenia kontynuujemy do chwili, gdy różnica pomiędzy dwoma kolejnymi przybliżeniami pierwiastka będzie dostatecznie mała. Napisz metodę `cbtrd()` (ang. *cube root*) obliczającą pierwiastek trzeciego stopnia z podanej liczby dodatniej. Zbuduj aplikację pokazującą działanie metody `cbtrd()` i porównującą otrzymane wyniki z wynikami metody bibliotecznej `Math.cbrt()`.



Uwaga

Podany wzór wynika z *metody Newtona-Raphsona* — iteracyjnego algorytmu wyznaczania przybliżonej wartości pierwiastka funkcji. Dotyczy to również wzoru z zadania 29.11.

Zadanie 29.13.

Pierwiastek n -tego stopnia z liczby dodatniej a możemy obliczyć metodą iteracyjną na podstawie wzoru $x = \frac{1}{n} \left((n-1)x + \frac{a}{x^{n-1}} \right)$, przyjmując jako pierwsze przybliżenie $x = 1$. Obliczenia kontynuujemy do chwili, gdy różnica pomiędzy dwoma kolejnymi przybliżeniami pierwiastka będzie dostatecznie mała. Na podstawie podanego wzoru napisz metodę `nRoot()` obliczającą pierwiastek n -tego stopnia z podanej liczby dodatniej. Zbuduj aplikację pokazującą działanie metody `nRoot()` i porównującą otrzymane wyniki z wynikami uzyskanymi przy zastosowaniu funkcji bibliotecznej.



Uwaga

Ponieważ $\sqrt[n]{a} = a^{\frac{1}{n}}$, to wartość pierwiastka możemy obliczyć przy zastosowaniu funkcji `Math.pow(a, 1.0/n)`.



Wskazówka

Skorzystaj z opracowanej w zadaniu 29.7 metody `power()` do obliczania wartości x^{n-1} .

Zadanie 29.14.

W klasie `Math` zdefiniowano metody obliczające funkcje hiperboliczne — *sinus hiperboliczny* (`Math.sinh()`) i *cosinus hiperboliczny* (`Math.cosh()`). Napisz program wyświetlający na ekranie tablice wszystkich funkcji hiperbolicznych w przedziale $\langle -5, 5 \rangle$ z krokiem $0,1$. Wyniki obliczeń zapisz w pliku tekstowym `FunkcjeHiperboliczne.txt`.



Wskazówka

Wyniki pracy programu można zapisać w pliku tekstowym, stosując w konsoli polecenie:

```
java Z29_14 > FunkcjeHiperboliczne.txt
```

(przyjmujemy, że rozwiązanie zadania zapisano w pliku źródłowym `Z29_14.java`).

Zadanie 29.15.

W klasie `Math` zdefiniowano metodę `Math.exp()` obliczającą wartość funkcji wykładniczej e^x . Rozwiąż zadanie 29.14, nie korzystając z metod `Math.sinh()` i `Math.cosh()`.



Wskazówka

Wartości funkcji można wyliczyć na podstawie wzorów: $\sinh x = \frac{e^x - e^{-x}}{2}$,

$\cosh x = \frac{e^x + e^{-x}}{2}$ itp.

Zadanie 29.16.

Zbuduj klasę FH (funkcje hiperboliczne) zawierającą metody statyczne obliczające wartości wszystkich funkcji hiperbolicznych i funkcji do nich odwrotnych. Napisz aplikację pokazującą działanie tych metod.

Zadanie 29.17.

Zbuduj klasę FTD, która będzie zawierać metody statyczne obliczające wartość sześciu funkcji trygonometrycznych i sześciu funkcji do nich odwrotnych. Argumenty funkcji trygonometrycznych i wartości funkcji odwrotnych powinny być wyrażane w stopniach. Udostępnij również metody konwersji stopni na radiany i radianów na stopnie. Napisz aplikację pokazującą możliwości metod statycznych zawartych w tej klasie.

Zadanie 29.18.

Zbuduj klasę FTR, która będzie zawierać metody statyczne obliczające wartość sześciu funkcji trygonometrycznych i sześciu funkcji do nich odwrotnych. Argumenty funkcji trygonometrycznych i wartości funkcji odwrotnych powinny być wyrażane w radianach. Udostępnij również metody konwersji stopni na radiany i radianów na stopnie. Napisz aplikację pokazującą możliwości metod statycznych zawartych w tej klasie.

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



- 1. ZAREJESTRUJ SIĘ**
- 2. PREZENTUJ KSIĄŻKI**
- 3. ZBIERAJ PROWIZJĘ**

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

<http://program-partnerski.helion.pl>

Poznaj niezawodną Javę!

Gdy przegląda się oferty pracy dla programistów, trudno nie natknąć się na słowo „Java”. Ten obiektowy, niezależny od platformy język programowania przez niemal dwie dekady swojego istnienia zdążył zapracować na reputację łatwego w użyciu i wydajnego narzędzia do różnych zastosowań, a także zjednać sobie przychylność rzesz profesjonalnych programistów i miłośników technologii informatycznych, traktujących programowanie jako hobby. Za teoretyczną znajomością Javy jednak nie zawsze idzie praktyczna umiejętność korzystania z technik programistycznych stosowanych w tym języku.

Właśnie dlatego powstała książka *Programowanie w języku Java. Zbiór zadań z (p)odpowiedziami*. Poprzedzające każdy dział publikacji krótkie, rzeczowe wprowadzenia teoretyczne umożliwią rozpoczęcie przygody z Javą nawet osobom, które nie miały z nią wcześniej żadnego kontaktu, a stanowiące rdzeń książki zadania pozwolą utrwalić wiadomości zarówno początkującym, jak i nieco bardziej zorientowanym użytkownikom tego języka. Całość uzupełniono o odpowiedzi, dzięki którym każdy czytelnik może zweryfikować poprawność rozwiązań, oraz wskazówki dotyczące tego, gdzie należy szukać dodatkowych informacji.

- Narzędzia przydatne przy programowaniu w języku Java
- Podstawowe typy logiczne, liczbowe i znakowe oraz sposoby ich używania
- Zastosowanie odpowiednich klas do przeprowadzania operacji na tekstach
- Wczytywanie danych od użytkownika i wyświetlanie wyników na ekranie
- Korzystanie z instrukcji warunkowych i instrukcji pętli
- Tworzenie i używanie złożonych typów danych
- Praktyczne wykorzystywanie klas bibliotecznych
- Obsługa wyjątków, przetwarzanie plików tekstowych i obliczenia statystyczne

Skończ z teorią, zacznij tworzyć prawdziwe programy!

helion.pl
księgarnia
internetowa

Nr katalogowy: 8694



Księgarnia internetowa:

<http://helion.pl>



Zamówienia telefoniczne:

0 801 339900



0 601 339900



Helion

Sprawdź najnowsze promocje:

• <http://helion.pl/promocje>

Książki najchętniej czytane:

• <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

• <http://helion.pl/nowosci>

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
<http://helion.pl>

sięgnij po WIĘCEJ



KOD KORZYŚCI

ISBN 978-83-246-4319-6



9 788324 643196

Cena: 59,00 zł

Informatyka w najlepszym wydaniu