



Mirosław J. Kubiak

# Java

*Zadania z programowania  
z przykładowymi rozwiązaniami*

*Java w analizie konkretnych przykładów*

- *Proste operacje wejścia/wyjścia*
- *Tablice oraz iteracje*
- *Programowanie obiektowe i pliki tekstowe*

Hellon



## » Idź do

- Spis treści
- Przykładowy rozdział

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 32 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991–2011

## Java. Zadania z programowania z przykładowymi rozwiązaniami

Autor: Mirosław Kubiak  
ISBN: 978-83-246-2944-2  
Format: 140×208, stron: 112



### Java w analizie konkretnych przykładów

- Proste operacje wejścia/wyjścia
- Tablice oraz iteracje
- Programowanie obiektowe i pliki tekstowe

Java to jeden z najważniejszych używanych obecnie obiektowych języków programowania. Do jego największych zalet należą łatwe przenoszenie programów między platformami i możliwość stosowania na urządzeniach przenośnych. Dobry programista, student lub nauczyciel informatyki, a także każdy człowiek zainteresowany programowaniem powinien znać podstawy tego języka i posiadać umiejętność rozwiązywania konkretnych problemów. Podobnie zresztą powinien opanować najważniejsze zagadnienia dotyczące programowania w językach C++ i Turbo Pascal – i stosować je w praktyce. Trzyczęściowy zbiór, w którym zamieszczono te same lub bardzo zbliżone zadania wraz z rozwiązaniami w każdym z wyżej wymienionych języków, pozwala sprawdzić i uzupełnić wiedzę poprzez analizę podanego kodu.

Książka „Java. Zadania z programowania z przykładowymi rozwiązaniami” to jedna z trzech części zbioru zadań programistycznych, zawierająca zadania w języku Java. Będziesz mógł dokładnie prześledzić, jak wyglądają w tym języku operacje wejścia/wyjścia i instrukcje warunkowe. Dowiesz się, do czego potrzebne są iteracje oraz w jaki sposób używa się tablic. Zobaczysz, na czym polega programowanie obiektowe i jak działa ono w Javie. Poznasz także zastosowania plików tekstowych. Taki układ książki ułatwi Ci przyswojenie sobie najważniejszych zagadnień z języka Java w najlepszy możliwy sposób – na prostych, konkretnych przykładach.

- Proste operacje wejścia/wyjścia
- Instrukcje warunkowe
- Iteracje
- Tablice – deklarowanie, dostęp do elementów, dwuwymiarowe
- Programowanie obiektowe
- Pliki tekstowe

### Praktycznie opanuj podstawy języka Java

# Spis treści

	<b>Od autora</b>	<b>5</b>
<b>Rozdział 1.</b>	<b>Proste operacje wejścia-wyjścia</b>	<b>7</b>
	Operacje wejścia-wyjścia — informacje ogólne	7
<b>Rozdział 2.</b>	<b>Podejmujemy decyzje w programie</b>	<b>19</b>
	Instrukcje warunkowe w języku Java	19
<b>Rozdział 3.</b>	<b>Iteracje</b>	<b>31</b>
	Iteracje — informacje ogólne	31
	Pętla for	32
	Pętla do ... while	33
	Pętla while	33
<b>Rozdział 4.</b>	<b>Tablice</b>	<b>57</b>
	Deklarowanie tablic jednowymiarowych	57
	Dostęp do elementów tablicy	58
	Tablice dwuwymiarowe	62
<b>Rozdział 5.</b>	<b>Programowanie obiektowe</b>	<b>79</b>
	Programowanie obiektowe — informacje ogólne	79
	Rekurencja	92
<b>Rozdział 6.</b>	<b>Pliki tekstowe</b>	<b>97</b>
	Pliki tekstowe — informacje ogólne	97

# 1

## Proste operacje wejścia-wyjścia

*W tym rozdziale zamieszczono proste zadania wraz z przykładowymi rozwiązaniami ilustrujące, w jaki sposób komputer komunikuje się z użytkownikiem w języku Java.*

### Operacje wejścia-wyjścia — informacje ogólne

Każda aplikacja powinna posiadać możliwość komunikowania się z użytkownikiem. Wykorzystując proste przykłady, pokażemy, w jaki sposób program napisany w języku Java komunikuje się z nim poprzez standardowe operacje wejścia-wyjścia.

Operacje wejścia-wyjścia w Javie są realizowane poprzez strumienie. **Strumień** jest pojęciem abstrakcyjnym. Może on wysyłać i pobierać informacje i jest połączony z fizycznym urządzeniem (np. klawiatura, ekran) poprzez system wejścia-wyjścia. W języku tym zdefiniowano dwa typy strumieni: bajtowe i znakowe. Standardowy strumień wyjściowy w Javie jest reprezentowany przez obiekt<sup>1</sup> `out` znajdujący się

---

<sup>1</sup> Obiekty zostaną omówione w rozdziale 5.

w klasie `System`. Jest to obiekt statyczny klasy `PrintStream` zawierający metody `print()` i `println()`.

Metoda `println()` wyświetla argumenty podane w nawiasach `()`, a następnie przechodzi do początku nowej linii. Pewną jej odmianą jest metoda `print()`. Jej działanie polega na wyświetlaniu argumentów podanych w nawiasach `()` bez przemieszczania kursora do nowego wiersza.

---

**ZADANIE****1.1**

Napisz program, który oblicza pole prostokąta. Wartości boków `a` i `b` wprowadzamy z klawiatury. W programie należy przyjąć, że zmienne `a`, `b` oraz pole są typu `double` (rzeczywistego).

---

*Przykładowe rozwiązanie — listing 1.1*

---

```
package zadanie11; //Zadanie 1.1
import java.io.*;

public class Main
{
    public static void main(String[] args)
        throws IOException
    {
        double a, b, pole;

        BufferedReader br = new BufferedReader(new
            ↳InputStreamReader(System.in));

        System.out.println("Program oblicza pole prostokąta.");
        System.out.println("Podaj bok a.");
        a = Double.parseDouble(br.readLine());
        System.out.println("Podaj bok b.");
        b = Double.parseDouble(br.readLine());
        pole = a*b;

        System.out.print("Pole prostokąta o boku a = " + a + "
            ↳i boku b = " + b);
        System.out.println(" wynosi " + pole + ".");
    }
}
```

---

Klasy w Javie grupowane są w jednostki zwane pakietami (ang. *package*). **Pakiet** to zestaw powiązanych ze sobą tematycznie klas. Do jego utworzenia służy słowo kluczowe `package`, po którym następuje nazwa pakietu zakończona średnikiem, co ilustruje linijka kodu poniżej:

```
package zadanie11; //Zadanie 1.12
```

### Linijka kodu

```
double a, b, pole;
```

umożliwia deklarację zmiennych `a`, `b` i `pole` (wszystkie są typu rzeczywistego — `double`) w programie. Instrukcja

```
System.out.println("Program oblicza pole prostokąta.");
```

wyświetla na ekranie komputera komunikat *Program oblicza pole prostokąta*.

W celu czytania z klawiatury liter i cyfr należy skorzystać z dwóch klas: `InputStreamReader` oraz `BufferedReader`. Najpierw tworzymy nowy obiekt klasy `InputStreamReader`, przekazując jej konstruktorowi obiekt `System.in`. Można go następnie wykorzystać w konstruktorze klasy `BufferedReader`. Tak opisana konstrukcja ma następujący zapis:

```
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
```

Powstały obiekt klasy `BufferedReader` możemy przypisać do zmiennej referencyjnej `br` i dalej, poprzez metodę `readLine()`, możemy wykorzystać go do wczytywania zmiennej `a` typu `double` ze strumienia wejściowego. Ilustruje to następująca linijka kodu:

```
a = Double.parseDouble(br.readLine());
```

Wczytywanie liczb odbywa się tak samo jak wczytywanie tekstu, musimy jednak dokonać odpowiedniej konwersji, tzn. zamiany ciągu znaków na odpowiadającą mu wartość liczbową. Służy do tego jedna z poniższych metod statycznych:

- ❑ `parseByte` z klasy `Byte` do odczytu bajtów,
- ❑ `parseDouble` z klasy `Double` do odczytu liczb typu `double`,
- ❑ `parseFloat` z klasy `Float` do odczytu liczb typu `float`,
- ❑ `parseInt` z klasy `Int` do odczytu liczb typu `int`,
- ❑ `parseLong` z klasy `Long` do odczytu liczb typu `long`.

Aby nasz program mógł zostać skompilowany, musimy do niego dodać następujące dwie linijki kodu:

```
import java.io.*;
```

---

<sup>2</sup> Komentarze w programie oznaczamy dwoma ukośnikami `//`; `// to jest komentarz`.

oraz

throws IOException

Są one niezbędne do obsługi błędów wejścia-wyjścia. Słowo kluczowe `import` oznacza, że do programu zaimportowano wszystkie (po kropce \*) pakiety `java.io`.

Pole prostokąta zostaje obliczone w instrukcji

```
pole = a*b;
```

Za wyświetlenie wartości zmiennych `a` i `b` oraz `pole` wraz z odpowiednim opisem są odpowiedzialne następujące linijki kodu:

```
System.out.print("Pole prostokąta o boku a = " + a + " i boku b = " + b);  
System.out.println(" wynosi " + pole + ".");
```

Rezultat działania programu można zobaczyć na rysunku 1.1.

**Program oblicza pole prostokąta.**

**Podaj bok a.**

**1**

**Podaj bok b.**

**2**

**Pole prostokąta o boku a = 1.0 i boku b = 2.0 wynosi 2.0.**

*Rysunek 1.1. Efekt działania programu Zadanie 1.1*

### ZADANIE

## 1.2

Napisz program, który wyświetla na ekranie komputera wartość predefiniowanej stałej  $\pi = 3,14\dots$  Należy przyjąć format wyświetlania tej stałej z dokładnością do pięciu miejsc po przecinku.

### Wskazówka

Język Java umożliwia formatowanie wyświetlanych danych w podobny sposób jak w języku C. Służy do tego metoda `printf`. Jej składnia jest następująca:

```
String format;  
System.out.printf(format, arg_1, arg_2, ..., arg_n); .
```

---

*Przykładowe rozwiązanie — listing 1.2*

---

```
package zadanie12; //Zadanie 1.2

public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Program wyświetla liczbę pi z zadaną
        ↳dokładnością.");
        System.out.printf("Pi = " + "%6.5f\n", Math.PI);
    }
}
```

---

Specyfikatory typów mogą być następujące:

- ❑ %d — integer,
- ❑ %e — double,
- ❑ %f — float.

Pomiędzy znakiem % i literą przyporządkowaną danemu typowi można określić ilość pól, na których ma zostać wyświetlona liczba, np.:

%7.2f — oznacza przyznanie siedmiu pól na liczbę typu float, w tym dwóch pól na jej część ułamkową;

%4d — oznacza przyznanie czterech pól na liczbę typu całkowitego.

W programie zapis

```
System.out.printf("Pi = " + "%6.5f\n", Math.PI);
```

powoduje, że na wydruk liczby  $\pi$  przeznaczonych zostaje sześć pól, w tym pięć na część ułamkową. Znak specjalny "... $\backslash$ n" (ang. *new line*) oznacza przejście na początek nowego wiersza. Math jest standardową klasą Javy, która umożliwi obliczenia matematyczne.

Rezultat działania programu można zobaczyć na rysunku 1.2.

**Program wyświetla liczbę pi z zadaną dokładnością.**  
**Pi = 3,14159**

*Rysunek 1.2. Efekt działania programu Zadanie 1.2*



## ZADANIE

**1.3**

Napisz program, który wyświetla na ekranie komputera pierwiastek kwadratowy z wartości predefiniowanej  $\pi = 3,14\dots$ . Należy przyjąć format wyświetlania pierwiastka kwadratowego ze stałej  $\pi$  z dokładnością do dwóch miejsc po przecinku.

*Przykładowe rozwiązanie — listing 1.3*

---

```
package zadanie13; //Zadanie 1.3

public class Main
{
    public static void main(String[] args)
    {
        System.out.println("Program wyświetla pierwiastek kwadratowy ");
        System.out.println("z liczby pi z dokładnością dwóch miejsc po  
↳przecinku.");
        System.out.printf("Sqrt(Pi) = " + "%2.2f\n", Math.sqrt(Math.PI));
    }
}
```

---

Metoda `sqrt()` pozwala na obliczenie pierwiastka kwadratowego. Jest ona metodą standardowej klasy `Math`.

Rezultat działania programu można zobaczyć na rysunku 1.3.

**Program wyświetla pierwiastek kwadratowy  
z liczby pi z dokładnością dwóch miejsc po przecinku.  
Sqrt(Pi) = 1,77**

*Rysunek 1.3. Efekt działania programu Zadanie 1.3*

## ZADANIE

**1.4**

Napisz program, który oblicza objętość kuli o promieniu  $r$ . Wartość promienia wprowadzamy z klawiatury. W programie należy przyjąć, że zmienne: promień  $r$  i `objetosc`, są typu `double` (rzeczywistego). Dla tych zmiennych należy przyjąć format wyświetlania na ekranie z dokładnością do dwóch miejsc po przecinku.

---

*Przykładowe rozwiązanie — listing 1.4*

---

```
package zadanie14; //Zadanie 1.4
import java.io.*;

public class Main
{
    public static void main(String[] args)
        throws IOException
    {
        double r, objetosc;

        BufferedReader br = new BufferedReader(new
            ↳InputStreamReader(System.in));

        System.out.println("Program oblicza objętość kuli.");
        System.out.println("Podaj promień r.");
        r = Double.parseDouble(br.readLine());

        objetosc = 4.0*Math.PI*r*r*r/3;

        System.out.print("Objętość kuli o promieniu r = ");
        System.out.printf("%2.2f", r);
        System.out.print(" wynosi ");
        System.out.printf("%2.2f.", objetosc);
    }
}
```

---

Objętość kuli o promieniu  $r$  oblicza następująca linijka kodu:

```
objetosc = 4.0*Math.PI*r*r*r/3;
```

gdzie potęgowanie zamieniono na mnożenie.

Rezultat działania programu można zobaczyć na rysunku 1.4.

**Program oblicza objętość kuli.**

**Podaj promień r.**

**1**

**Objętość kuli o promieniu  $r = 1,00$  wynosi 4,19.**

*Rysunek 1.4. Efekt działania programu Zadanie 1.4*

**ZADANIE****1.5**

Napisz program, który oblicza wynik dzielenia całkowitego bez reszty dwóch liczb całkowitych:  $a = 37$  i  $b = 11$ .

**Wskazówka**

W języku Java w przypadku zastosowania operatora dzielenia / dla liczb całkowitych reszta wyniku jest pomijana (tak samo jest w C i C++).

W Turbo Pascalu należy zastosować operator dzielenia całkowitego bez reszty div.

---

*Przykładowe rozwiązanie — listing 1.5*

---

```
package zadanie15; //Zadanie 1.5

public class Main
{
    public static void main(String[] args)
    {
        int a = 37;
        int b = 11;

        System.out.println("Program wyświetla wynik dzielenia całkowitego");
        System.out.println("bez reszty dwóch liczb całkowitych.");
        System.out.println("Dla liczb a = " + a + ", b = " + b);
        System.out.println(a + "/" + b + " = "+ a/b + ".");
    }
}
```

---

Rezultat działania programu można zobaczyć na rysunku 1.5.

**Program wyświetla wynik dzielenia całkowitego bez reszty dwóch liczb całkowitych.**  
**Dla liczb a = 37, b = 11**  
**37/11 = 3.**

*Rysunek 1.5. Efekt działania programu Zadanie 1.5*

## ZADANIE

## 1.6

Napisz program, który oblicza resztę z dzielenia całkowitego dwóch liczb całkowitych:  $a = 37$  i  $b = 11$ .

**Wskazówka**

Należy zastosować operator reszty z dzielenia całkowitego modulo, który oznaczamy w języku Java symbolem `%`. Podobnie jak w językach C i C++, operator ten umożliwi uzyskanie tylko reszty z dzielenia, natomiast wartość całkowita jest odrzucana.

*Przykładowe rozwiązanie — listing 1.6*

```
package zadanie16; //Zadanie 1.6

public class Main
{
    public static void main(String[] args)
    {
        int a = 37;
        int b = 11;

        System.out.println("Program oblicza resztę z dzielenia
        ↪całkowitego");
        System.out.println("dwóch liczb całkowitych.");
        System.out.println("Dla liczb a = " + a + ", b = " + b);
        System.out.println(a + "%" + b + " = " + a%b + ".");
    }
}
```

Rezultat działania programu można zobaczyć na rysunku 1.6.

**Program oblicza resztę z dzielenia całkowitego  
dwóch liczb całkowitych.  
Dla liczb  $a = 37$ ,  $b = 11$   
 $37\%11 = 4$ .**

*Rysunek 1.6. Efekt działania programu Zadanie 1.6*

## ZADANIE

## 1.7

Napisz program, który oblicza sumę, różnicę, iloczyn i iloraz dla dwóch liczb  $x$  i  $y$  wprowadzanych z klawiatury. W programie należy założyć, że zmienne  $x$  i  $y$  są typu `float` (rzeczywistego). Dla zmiennych  $x$ ,  $y$ , `suma`, `roznica`, `iloczyn` i `iloraz` należy przyjąć format ich wyświetlania na ekranie z dokładnością do dwóch miejsc po przecinku.

---

*Przykładowe rozwiązanie — listing 1.7*

---

```
package zadanie17; //Zadanie 1.7
import java.io.*;

public class Main
{

    public static void main(String[] args)
        throws IOException
    {
        float x, y, suma, roznica, iloczyn, iloraz;

        BufferedReader br = new BufferedReader(new
            ↳InputStreamReader(System.in));

        System.out.println("Program oblicza sumę, różnicę, iloczyn
            ↳i iloraz ");
        System.out.println("dla dwóch liczb x i y wprowadzonych
            ↳z klawiatury.");
        System.out.println("Podaj x.");
        x = Float.parseFloat(br.readLine());
        System.out.println("Podaj y.");
        y = Float.parseFloat(br.readLine());

        suma = x+y;
        roznica = x-y;
        iloczyn = x*y;
        iloraz = x/y;

        System.out.printf("Dla liczb x = " + "%2.2f",x);
        System.out.printf(" i y = " + "%2.2f",y);
        System.out.println(); //wyświetlenie pustej linii
        System.out.printf("suma = " + "%2.2f,\n", + suma);
        System.out.printf("różnica = " + "%2.2f,\n", + roznica);
        System.out.printf("iloczyn = " + "%2.2f,\n", + iloczyn);
        System.out.printf("iloraz = " + "%2.2f,\n", + iloraz);

    }
}
```

---

Rezultat działania programu można zobaczyć na rysunku 1.7.

**Program oblicza sumę, różnicę, iloczyn i iloraz  
dla dwóch liczb x i y wprowadzonych z klawiatury.**

**Podaj x.**

**3**

**Podaj y.**

**2**

**Dla liczb  $x = 3,00$  i  $y = 2,00$**

**suma = 5,00,**

**różnica = 1,00,**

**iloczyn = 6,00,**

**iloraz = 1,50.**

*Rysunek 1.7. Efekt działania programu Zadanie 1.7*

» Java to jeden z najważniejszych używanych obecnie obiektowych języków programowania. Do jego największych zalet należą łatwe przenoszenie programów między platformami i możliwość stosowania na urządzeniach przenośnych. Dobry programista, student lub nauczyciel informatyki, a także każdy człowiek zainteresowany programowaniem powinien znać podstawy tego języka i posiadać umiejętność rozwiązywania konkretnych problemów. Podobnie zresztą powinien opanować najważniejsze zagadnienia dotyczące programowania w językach C++ i Turbo Pascal – i stosować je w praktyce. Trzyczęściowy zbiór, w którym zamieszczono te same lub bardzo zbliżone zadania wraz z rozwiązaniami w każdym z wyżej wymienionych języków, pozwala sprawdzić i uzupełnić wiedzę poprzez analizę podanego kodu.

» Książka *Java. Zadania z programowania z przykładowymi rozwiązaniami* to jedna z trzech części zbioru zadań programistycznych, zawierająca zadania w języku Java. Będziesz mógł dokładnie prześledzić, jak wyglądają w tym języku operacje wejścia/wyjścia i instrukcje warunkowe. Dowiesz się, do czego potrzebne są iteracje oraz w jaki sposób używa się tablic. Zobaczysz, na czym polega programowanie obiektowe i jak działa ono w Javie. Poznasz także zastosowania plików tekstowych. Taki układ książki ułatwi Ci przyswojenie sobie najważniejszych zagadnień z języka Java w najlepszy możliwy sposób – na prostych, konkretnych przykładach.

- *Proste operacje wejścia/wyjścia*
- *Instrukcje warunkowe*
- *Iteracje*
- *Tablice – deklarowanie, dostęp do elementów, dwuwymiarowe*
- *Programowanie obiektowe*
- *Pliki tekstowe*

**Praktycznie opanuj podstawy języka Java.**

nr katalogowy: 5600



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/najpopularny>  
Zamów informacje o nowościach:  
• <http://helion.pl/news>

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

**helion.pl**  
Księgarnia  
Internetowa

Cena: 19,90 zł

ISBN 978-83-246-2944-2



9 788324 629442

**Informatyka w najlepszym wydaniu**