Mirosław J. Kubiak

C++

Zadania z programowania z przykładowymi rozwiązaniami

C++ w analizie konkretnych przykładów

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

Helion

C++. Zadania z programowania z przykładowymi rozwiązaniami

Autor: Mirosław Kubiak
ISBN: 978-83-246-2943-5
Format: 140×208, stron: 128

C++ w analizie konkretnych przykładów

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

Odrobinę zapomniany już język C++ wciąż ma ogromną wartość; w wielu miejscach i zastosowaniach nadal sprawdza się znakomicie. Dobry programista, student lub nauczyciel informatyki, a także każdy człowiek zainteresowany programowaniem powinien znać podstawy tego języka i umieć rozwiązywać konkretne zadania. Podobnie zresztą powinien opanować najważniejsze zagadnienia dotyczące programowania w językach Java 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 we wszystkich tych językach.

Książka „C++. Zadania z programowania z przykładowymi rozwiązaniami" to jedna z trzech części zbioru zadań programistycznych, zawierająca zadania w języku C++. Znajdziesz tu ćwiczenia w zakresie komunikowania się komputera z użytkownikiem (standardowe operacje wejścia/wyjścia), wykorzystania instrukcji warunkowych oraz iteracji, używania tablic jedno- i dwuwymiarowych. Kolejne zadania dotyczyć będą podprogramów, programowania obiektowego oraz zastosowania plików tekstowych. Taki układ książki ułatwi Ci przyswojenie sobie najważniejszych zagadnień z języka C++ w najlepszy możliwy sposób – na prostych, konkretnych przykładach.

- Operacje wejścia/wyjścia
- Instrukcje warunkowe
- Iteracje
- Tablice jedno- i dwuwymiarowe
- Podprogramy
- Programowanie obiektowe
- Pliki tekstowe

Praktycznie opanuj podstawy języka C++

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

Spis treści

1

Proste operacje wejścia-wyjścia

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

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

Plik nagłówkowy z instrukcji

```
#include <iostream.h>
```

zawiera definicje klas[1] umożliwiających wykonywanie operacji wejścia-wyjścia na strumieniach. Do wyprowadzania danych na ekran służy standardowy strumień wyjściowy `cout`, który w języku C++ domyślnie przypisuje ekran do standardowego urządzenia wyjściowego systemu operacyjnego. Aby wyświetlić komunikat lub dane, trzeba do strumienia wyjściowego `cout` zastosować symbol podwójnego znaku mniejszości << (operacja wstawiania). Dwa znaki mniejszości należy wprowadzić z klawiatury.

[1] Więcej informacji na temat klas czytelnik znajdzie w rozdziale 6.

Do wprowadzania danych do programu służy standardowy strumień wejściowy `cin` oraz operator `>>` (dwa znaki większości, które również wprowadzamy z klawiatury), np. `cin >> a;`.

Do formatowania strumienia wyjściowego będziemy używali flagi formatującej `fixed` i manipulatora `setprecision(n)`. Flaga `fixed` używa do liczb zmiennoprzecinkowych ustalonej kropki dziesiętnej, natomiast manipulator `setprecision(n)` ustala ich precyzję na *n* — np. zapis `cout << setprecision(2);` oznacza, że liczby zmiennoprzecinkowe będą wyświetlane z dokładnością dwóch miejsc po kropce.

Zastosowanie manipulatora `setprecision(n)` wymaga włączenia do programu pliku nagłówkowego:

```
#include <iomanip.h>
```

Opisane powyżej podejście do operacji wejścia-wyjścia nazywa się obiektowym[2].

ZADANIE

1.1 Napisz program, który oblicza pole prostokąta. Wartości boków `a` i `b` wprowadzamy z klawiatury. W programie należy przyjąć, że zmienne `a` i `b` oraz `pole` są typu `float` (rzeczywistego). Przyjmujemy format wyświetlania ich na ekranie z dokładnością dwóch miejsc po kropce.

Przykładowe rozwiązanie — listing 1.1

```
#include <iostream.h> // Zadanie 1.1
#include <iomanip.h>
#include <conio.h>

main()
{
    float a, b, pole;

    cout << "Program oblicza pole prostokata." << endl;
    cout << "Podaj bok a." << endl;
    cin >> a;
    cout << "Podaj bok b." << endl;
    cin >> b;
    pole = a*b;
```

[2] Więcej informacji na temat obiektowych operacji wejścia-wyjścia, flag i manipulatorów znajdzie czytelnik na stronach WWW poświęconych językowi programowania C++ pod adresem *http://www.cplusplus.com/*.

```
    cout << fixed; // flaga
    cout << setprecision(2); // ustalenie precyzji
    cout << "Pole prostokata o boku a = " << a << " i boku b = " << b;
    cout << " wynosi " << pole << "." << endl;

    getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Linijka kodu

```
float a, b, pole;
```

umożliwia zadeklarowanie zmiennych `a`, `b` i `pole` (wszystkie zmienne w programie są typu rzeczywistego `float`). Instrukcja

```
cout << "Program oblicza pole prostokata." << endl;
```

wyświetla na ekranie komputera komunikat *Program oblicza pole prostokata*. Instrukcja `cin >> a;` czeka na wprowadzenie z klawiatury komputera liczby, która następnie zostanie przypisana zmiennej `a`. Pole prostokąta zostaje obliczone w wyrażeniu

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

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

```
    cout << "Pole prostokata o boku a = " << a << " i boku b = " << b;
    cout << " wynosi " << pole << "." << endl;
```

Flaga `fixed` używa ustalonej kropki dziesiętnej dla liczb zmiennoprzecinkowych. Zapis

```
cout << setprecision(2);
```

oznacza, że liczby te będą wyświetlane na ekranie z dokładnością dwóch miejsc po kropce. Natomiast funkcja

```
getch();
```

(ang. *get character* — wczytaj znak) czeka na wczytanie dowolnego znaku z klawiatury (naciśnięcie dowolnego klawisza). Prototyp tej funkcji znajduje się w pliku nagłówkowym *conio.h*. Instrukcja

```
endl;
```

(ang. *end of line* — koniec linii) przenosi kursor na początek następnej linii.

Komentarze w programie oznaczamy dwoma ukośnikami

```
// to jest komentarz do programu
```

Są one ignorowane w procesie kompilacji.

Rezultat działania programu można zobaczyć na rysunku 1.1.

```
Program oblicza pole prostokata.
Podaj bok a.
1
Podaj bok b.
2
Pole prostokata o boku a = 1.00 i boku b = 2.00 wynosi 2.00.
```

Rysunek 1.1. *Efekt działania programu Zadanie 1.1*

ZADANIE

1.2

Napisz program, który wyświetla na ekranie komputera wartość predefiniowanej stałej $\pi = 3{,}14\ldots$ Należy przyjąć format prezentowania tej stałej, oznaczanej w języku C++ jako `M_PI`, z dokładnością pięciu miejsc po kropce.

> **Wskazówka**
>
> Stała `M_PI` znajduje się w pliku nagłówkowym *math.h*, który poleceniem `#include <math.h>` należy dołączyć do programu.

Przykładowe rozwiązanie — listing 1.2

```
#include <iostream.h> // Zadanie 1.2
#include <iomanip.h>
#include <math.h>
#include <conio.h>

main()
{
 cout << "Program wyswietla wartosc predefiniowanej stalej pi" << endl;
 cout << "z dokladnoscia pieciu miejsc po kropce." << endl;
 cout << "pi = " << fixed << setprecision(5) << M_PI << endl;

 getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Rezultat działania programu można zobaczyć na rysunku 1.2.

```
Program wyswietla wartosc predefiniowanej stalej pi
z dokladnoscia pieciu miejsc po kropce.
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 stałej $\pi = 3{,}14...$ Należy przyjąć format wyświetlania tego pierwiastka z dokładnością dwóch miejsc po kropce.

> **Wskazówka**
>
> Pierwiastek kwadratowy ze stałej π obliczamy, korzystając z funkcji `sqrt()`. Funkcja ta znajduje się w pliku nagłówkowym *math.h*.

Przykładowe rozwiązanie — listing 1.3

```
#include <iostream.h> // Zadanie 1.3
#include <iomanip.h>
#include <math.h>
#include <conio.h>

main()
{
 cout << "Program wyswietla pierwiastek kwadratowy z pi";
 cout << " z dokladnoscia dwoch miejsc po kropce." << endl;
 cout << "Sqrt(pi) = " << fixed << setprecision(2) << sqrt(M_PI) << endl;

getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Rezultat działania programu można zobaczyć na rysunku 1.3.

```
Program wyswietla pierwiastek kwadratowy
z pi z dokladnoscia dwoch miejsc po kropce.
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 `r` jest typu `float` (rzeczywistego). Dla zmiennych `r` oraz `objetosc` należy przyjąć format wyświetlania ich na ekranie z dokładnością dwóch miejsc po kropce.

Przykładowe rozwiązanie — listing 1.4

```
#include <iostream.h> // Zadanie 1.4
#include <iomanip.h>
#include <math.h>
#include <conio.h>

main()
{
  float r, objetosc;

  cout << "Program oblicza objetosc kuli o promieniu r." << endl;
  cout << "Podaj promien r." << endl;
  cin >> r;
  objetosc = 4*M_PI*r*r*r/3;
  cout << fixed;
  cout << setprecision(2);
  cout << "Objetosc kuli o promieniu r = " << r << " wynosi ";
  cout << objetosc << "." << endl;

  getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Objętość kuli o promieniu `r` oblicza linijka kodu

```
objetosc = 4*M_PI*r*r*r/3;
```

gdzie potęgowanie zamieniono na mnożenie.

Rezultat działania programu można zobaczyć na rysunku 1.4.

```
Program oblicza objetosc kuli o promieniu r.
Podaj promien r.
1
Objetosc 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 dla dwóch liczb całkowitych a = 37 i b = 11.

> **Wskazówka**
>
> W języku C++ w przypadku zastosowania operatora dzielenia / dla liczb całkowitych reszta wyniku jest pomijana[3].

Przykładowe rozwiązanie — listing 1.5

```
#include <iostream.h> // Zadanie 1.5
#include <conio.h>

main()
{
 int a = 37;
 int b = 11;

 cout << "Program oblicza wynik dzielenia calkowitego" << endl;
 cout << "dla dwoch liczb calkowitych." << endl;
 cout << "Dla liczb a = " << a << " i b = " << b << endl;
 cout << a << "/" << b << " = " << a/b << "." << endl;

 getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Rezultat działania programu można zobaczyć na rysunku 1.5.

```
Program oblicza wynik dzielenia calkowitego
dla dwoch liczb calkowitych.
Dla liczb a = 37 i b = 11
37/11 = 3.
```

Rysunek 1.5. *Efekt działania programu Zadanie 1.5*

[3] W języku Turbo Pascal należy zastosować operator dzielenia całkowitego bez reszty `div`.

ZADANIE

1.6

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

Wskazówka

Należy zastosować operator reszty z dzielenia całkowitego modulo, który oznaczamy w języku C++ symbolem %. Operator ten umożliwia uzyskanie tylko reszty z dzielenia, natomiast całkowita wartość liczbowa jest odrzucana.

Przykładowe rozwiązanie — listing 1.6

```
#include <iostream.h> // Zadanie 1.6
#include <conio.h>

main()
{
 int a = 37;
 int b = 11;

 cout << "Program oblicza reszte z dzielenia calkowitego";
 cout << " dwoch liczb calkowitych." << endl;
 cout << "Dla liczb a = " << a << " i b = " << b << endl;
 cout << a << "%" << b << " = " << a%b << "." << endl;

 getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Rezultat działania programu można zobaczyć na rysunku 1.6.

```
Program oblicza reszte z dzielenia calkowitego dwoch liczb
calkowitych.
Dla liczb a = 37 i 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 przyjmujemy, że liczby `x` i `y` są typu `float` (rzeczywistego). Dla zmiennych `x`, `y`, `suma`, `roznica`, `iloczyn` i `iloraz` należy przyjąć format wyświetlania ich na ekranie z dokładnością dwóch miejsc po kropce.

Przykładowe rozwiązanie — listing 1.7

```
#include <iostream.h> // Zadanie 1.7
#include <iomanip.h>
#include <conio.h>

main()
{
 float x, y, suma, roznica, iloczyn, iloraz;

 cout << "Program oblicza sume, roznice, iloczyn i iloraz" << endl;
 cout << "dla dwoch liczb x i y wprowadzanych z klawiatury." << endl;
 cout << endl;
 cout << "Podaj x." << endl;
 cin >> x;
 cout << "Podaj y." << endl;
 cin >> y;

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

 cout << fixed;
 cout << setprecision (2);
 cout << "Dla x = " << x << " i y = " << y << endl;
 cout << endl; // wydruk pustej linii
 cout << "suma = " << suma << "," << endl;
 cout << "roznica = " << roznica << "," << endl;
 cout << "iloczyn = " << iloczyn << "," << endl;
 cout << "iloraz = " << iloraz << ".";

 getch(); // czeka na nacisniecie dowolnego klawisza
}
```

Rezultat działania programu można zobaczyć na rysunku 1.7.

```
Program oblicza sume, roznice, iloczyn i iloraz
dla dwoch liczb x i y wprowadzanych z klawiatury.

Podaj x.
1
Podaj y.
2
Dla x = 1.00 i y = 2.00

suma = 3.00,
roznica = -1.00,
iloczyn = 2.00,
iloraz = 0.50.
```

Rysunek 1.7. *Efekt działania programu Zadanie 1.7*

Odrobinę zapomniany już język C++ wciąż ma ogromną wartość; w wielu miejscach i zastosowaniach nadal sprawdza się znakomicie. Dobry programista, student lub nauczyciel informatyki, a także każdy człowiek zainteresowany programowaniem powinien znać podstawy tego języka i umieć rozwiązywać konkretne zadania. Podobnie zresztą powinien opanować najważniejsze zagadnienia dotyczące programowania w językach Java 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 we wszystkich tych językach.

Książka *„C++. Zadania z programowania z przykładowymi rozwiązaniami"* to jedna z trzech części zbioru zadań programistycznych, zawierająca zadania w języku C++. Znajdziesz tu ćwiczenia w zakresie komunikowania się komputera z użytkownikiem (standardowe operacje wejścia/wyjścia), wykorzystania instrukcji warunkowych oraz iteracji, używania tablic jedno- i dwuwymiarowych. Kolejne zadania dotyczyć będą podprogramów, programowania obiektowego oraz zastosowania plików tekstowych. Taki układ książki ułatwi Ci przyswojenie sobie najważniejszych zagadnień z języka C++ w najlepszy możliwy sposób — na prostych, konkretnych przykładach.

- *Operacje wejścia/wyjścia*
- *Instrukcje warunkowe*
- *Iteracje*
- *Tablice jedno- i dwuwymiarowe*
- *Podprogramy*
- *Programowanie obiektowe*
- *Pliki tekstowe*

Praktycznie opanuj podstawy języka C++.

Nr katalogowy: 5801

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Informatyka w najlepszym wydaniu

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

helion.pl
księgarnia internetowa

Cena: 19,90 zł

ISBN 978-83-246-2943-5