

# Podejmujemy decyzje w programie

W języku C++ istnieją dwie instrukcje warunkowe:

- instrukcja warunkowa *if ... else*,
- instrukcja wyboru *switch .. case*.

Instrukcja warunkowa *if ... else* służy do sprawdzania poprawności wyrażenia warunkowego i w zależności od tego, czy dany warunek jest prawdziwy, czy nie, pozwala na wykonanie różnych bloków programu.

Jej ogólna postać jest następująca:

```
if (warunek)
{
    // instrukcje do wykonania, kiedy warunek
    // jest prawdziwy
}

else
{
    // instrukcje do wykonania, kiedy warunek
    // jest fałszywy
}
```

Blok `else` jest opcjonalny i instrukcja warunkowa w wersji skróconej ma następującą postać:

```
if (warunek) { // instrukcje do wykonania,
// kiedy warunek jest prawdziwy }
```

Jeżeli `warunek` nie jest prawdziwy, to instrukcja warunkowa `if` nie zostanie wykonana.

Instrukcja wyboru `switch ... case` pozwala w wygodny i przejrzysty sposób sprawdzić ciąg warunków i wykonywać kod w zależności od tego, czy są one prawdziwe, czy fałszywe.

Jej ogólna postać jest następująca:

```
switch (wyrażenie)
{
    case wartość_1 : instrukcje_1;
    break;

    case wartość_2 : instrukcje_2;
    break;

    .....
    case wartość_n : instrukcje_n;
    break;

    default : instrukcje;
}
```

Instrukcja *break* przerywa wykonywanie całego bloku *case*.

## UWAGA!

Jej brak może doprowadzić do nieoczekiwanych wyników i błędów w programie.

Przykładowy program napisany w języku C++ podejmujący decyzje w programie

Program, który dla trzech boków trójkąta, **a**, **b** i **c**, wprowadzonych z klawiatury sprawdza, czy tworzą one trójkąt prostokątny (zakładamy, że  $a > 0$ ,  $b > 0$ ,  $c > 0$ ).

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

using namespace std;

int main()
{
    int a, b, c;

    cout << "Program sprawdza, czy boki a, b i c tworzą trójkąt
prostokątny." << endl;
    cout << "Podaj bok a." << endl;
    cin >> a;
    cout << "Podaj bok b." << endl;
    cin >> b;
    cout << "Podaj bok c." << endl;
    cin >> c;

    if ((a*a + b*b) == (c*c))
    {
        cout << "Boki a = " << a << ", b = " << b << " i c = " << c;
        cout << " tworzą trójkąt prostokątny." << endl;
    }
    else
    {
        cout << "Boki a = " << a << ", b = " << b << " i c = " << c;
        cout << " nie tworzą trójkąta prostokątnego." << endl;
    }
}
```

```
    }  
  
    _getch();  
  
    return 0;  
}
```

## Opis instrukcji, linii kodu

Sprawdzenie twierdzenia Pitagorasa dla wczytanych boków a, b i c zostało zawarte w następujących linijkach kodu:

```
if ((a*a + b*b) == (c*c))  
{  
    cout << "Boki a = " << a << ", b = " << b << " i c = " << c;  
    cout << " tworzą trójkąt prostokątny." << endl;  
}  
else  
{  
    cout << "Boki a = " << a << ", b = " << b << " i c = " << c;  
    cout << " nie tworzą trójkąta prostokątnego." << endl;  
}
```

Łatwo sprawdzić, że boki  $a = 3$ ,  $b = 4$ ,  $c = 5$  tworzą trójkąt prostokątny (liczby te spełniają twierdzenie Pitagorasa) i na ekranie pojawi się komunikat *Boki ... tworzą trójkąt prostokątny*, natomiast boki  $a = 1$ ,  $b = 2$ ,  $c = 3$  nie tworzą trójkąta prostokątnego (liczby te nie spełniają twierdzenia Pitagorasa), więc na ekranie zostanie wyświetlony komunikat *Boki ... nie tworzą trójkąta prostokątnego*.

## Zadania

2.1 Napisz program, który z wykorzystaniem instrukcji warunkowej *if* oblicza pierwiastki równania kwadratowego  $ax^2 + bx + c = 0$ , gdzie zmienne  $a$ ,  $b$  i  $c$  to liczby rzeczywiste wprowadzane z klawiatury. Dla zmiennych  $a$ ,  $b$ ,  $c$ ,  $x1$  oraz  $x2$  należy przyjąć format wyświetlania ich z dokładnością do dwóch miejsc po kropce.

## wskazówki do zadania

W pierwszej części programu sprawdzamy, czy wartość współczynnika  $a$  jest równa zero.

Jeśli  $a = 0$ , powinien zostać wyświetlony komunikat typu: *Niedozwolona wartość współczynnika a...* oraz program powinien zakończyć swoje działanie, natomiast dla  $a$  różnego od zera program powinien kontynuować swoje działanie zaczynając od oczekiwania na wprowadzenie wartości  $b$  i  $c$ , następnie program powinien sprawdzić czy występują pierwiastki równania kwadratowego. Aby tego dokonać należy obliczyć *delta*, według dobrze znanego wzoru matematycznego, który wygląda następująco:

$$\text{delta} = b * b - 4 * a * c;$$

Jeśli  $\text{delta} < 0$ , to dane równanie kwadratowe nie posiada pierwiastków rzeczywistych.

W przypadku, gdy  $\delta = 0$ , to dane równanie kwadratowe posiada jeden pierwiastek podwójny, który obliczamy przy użyciu następującego wzoru:

$$x_1 = -b / (2 * a);$$

Natomiast, jeśli  $\delta > 0$ , dane równanie kwadratowe posiada dwa pierwiastki rzeczywiste, które obliczamy przy użyciu następujących wzorów:

$$x_1 = (-b - \sqrt{\delta}) / (2 * a);$$

$$x_2 = (-b + \sqrt{\delta}) / (2 * a);$$

Przykładowo dla  $a = 1$ ,  $b = 5$  i  $c = 4$  wartość pierwiastków równania wynoszą odpowiednio  $x_1 = -4$  i  $x_2 = -1$ . Dla  $a = 1$ ,  $b = 4$  i  $c = 4$  trójmian ma z kolei jeden pierwiastek podwójny  $x_1 = -2$ . Dla  $a = 1$ ,  $b = 2$  i  $c = 3$  trójmian nie ma pierwiastków rzeczywistych.

2.2 Napisz program, który z wykorzystaniem instrukcji wyboru ***switch*** oblicza pierwiastki równania kwadratowego  $ax^2 + bx + c = 0$ , gdzie zmienne  $a$ ,  $b$  i  $c$  to liczby rzeczywiste wprowadzane z klawiatury. Dla zmiennych  $a$ ,  $b$ ,  $c$ ,  $x1$  oraz  $x2$  należy przyjąć format wyświetlania ich na ekranie z dokładnością do dwóch miejsc po kropce.

### wskazówki do zadania

Należy wprowadzić do programu zmienną pomocniczą *liczba\_pierwiastkow*.

2.3 Napisz program, który oblicza wartość  $x$  z równania  $ax + b = c$ . Wartości  $a$ ,  $b$  i  $c$  należą do zbioru liczb rzeczywistych i są wprowadzane z klawiatury. Dodatkowo należy zabezpieczyć program na wypadek sytuacji, kiedy wprowadzona wartość  $a$  jest równa zero. Dla

zmiennych  $a$ ,  $b$ ,  $c$  oraz  $x$  należy przyjąć format wyświetlania ich z dokładnością do dwóch miejsc po kropce.

2.4 Napisz program, w którym użytkownik zgaduje liczbę losową z przedziału od 0 do 9 generowaną przez komputer.

### wskazówki do zadania

Aby wygenerować liczby pseudolosowe w środowisku Visual C++, należy włączyć do programu plik nagłówkowy *random*.

```
#include <random>
```

Jednym ze sposobów użycia biblioteki *random* przedstawiają następujące linijki kodu:

*random\_device rd; // dany kod uruchamia generator liczb pseudolosowych o nazwie Mersenne Twister co pokazuje kolejna linia kodu*

```
mt19937 gen(rd());  
uniform_int_distribution<> dist(1, 9);  
// powyższa linia kodu zawęża generowanie liczb  
pseudolosowych do przedziału od 1 do 9
```

Powyższe linie kodu są odpowiedzialne za uruchomienie generatora liczb pseudolosowych o nazwie Mersenne Twister – w skrócie MT. Generator MT jest jednym, dużym rejestrem przesuwnym ze sprzężeniami zwrotnymi powodującymi rotację oraz mieszanie się bitów. Rejestr ma długość 19937 bitów (liczba Mersenne'a), które w pamięci zajmują 624 słowa 32-bitowe. Ostatnia linijka kodu zawęża generowanie liczb pseudolosowych do przedziału od 1 do 9, zgodnie z wymaganiami zadania.

2.5 Wykonuj wszystkie zadania rzetelnie i bez pomocy kolegi/koleżanki. Kolega/Koleżanka nie napisze za Ciebie egzaminu, nawet gdy

trafisz z nim/nią do tej samej grupy egzaminacyjnej, to każda próba jakiejkolwiek pomocy kończy się wyproszeniem Ciebie oraz osoby, która próbowała Tobie pomóc z egzaminu przez Egzaminatora oraz niezaliczeniem danej kwalifikacji zawodowej dla was dwóch lub nawet całej grupy egzaminacyjnej, w zależności od decyzji Egzaminatora.

Powodzenia ;)