



Podstawowa struktura programu w języku C++

```
#include <iostream>
using namespace std;
...           // deklaracje stałych
...           // deklaracje funkcji
int main ()
{
    // część wykonawcza programu
    return 0;
}
```

Zadanie 1. Wyprowadzanie danych

Napisz program wyprowadzający na ekran napis: „jestem na konferencji”. Zapisz program pod nazwą *zadanie1*.

Wskazówki: Prosty program w języku C++ ma postać:

```
Zadanie1.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      cout << "Jestem na konferencji";
8      return 0;
9  }
```

W języku C++ `cout` jest obiektem reprezentującym standardowe wyjście.

Zadanie 2. Deklarowanie zmiennych, wprowadzanie danych, wykonywanie obliczeń i wyprowadzanie wyników na ekran

Napisz program umożliwiający wprowadzenie dwóch liczb całkowitych a i b , obliczenie ich sumy ($suma$) i wyprowadzenie wyniku ($suma$) na ekran monitora. Skompiluj i uruchom program. Zapisz program pod nazwą *zadanie2*.

Wskazówki:

W pokazanym fragmencie programu zadeklarowano trzy zmienne typu całkowitego:

```
Zadanie2.cpp
1  #include <iostream>
2
3  using namespace std;
4  int a,b,suma;
5  int main()
6  {
```

Przykład wprowadzania danych: `cin >> a >> b;`

W języku C++ `cin` jest obiektem reprezentującym standardowe wejście.

W instrukcji przypisania używamy znaku „=” np.: `iloczyn=a*b;`

Zadanie 3. Dodawanie opisów

Do programu z zadania 2. dodaj opisy „wprowadz pierwsza liczbe”, „wprowadz druga liczbe”, „suma wynosi”.

Wskazówki: Umieszczenie `endl` lub znaków `"\n"` na końcu polecenia `cout` zaczyna nowy wiersz, np.: `cout << "wprowadz pierwsza liczbe:" << "\n";`

Zadanie 4. Stosowanie instrukcji warunkowej

Napisz program, który umożliwi wprowadzenie z klawiatury dwóch różnych liczb całkowitych (*liczba1*, *liczba2*) i wyprowadzenie na ekran większej z nich.

Wskazówki: Instrukcja warunkowa w języku C++ ma postać:

```
if (wyrażenie)
    instrukcja1;
else instrukcja2;
```

lub

```
if (wyrażenie) instrukcja1;
```

Na przykład:

```
13     if(liczba1>liczba2)
14         cout << "wieksza jest: " << liczba1;
15     else cout << "wieksza jest: " << liczba2;
16
```

Poniżej przykład realizacji zadania 4. w języku Scratch (gimnazjum):

The image shows a Scratch project window titled "liczba większa". On the left, the stage area displays a cat character with a speech bubble that says "Podaj drugą liczbę". Below the stage, there are two variable monitors: "Liczba1" with a value of 45 and "Liczba2" with a value of 0. On the right, the script area contains the following code blocks: "kiedy kliknięto" (when clicked), "zapytaj Podaj pierwszą liczbę i czekaj" (ask for first number), "ustaw Liczba1 na odpowiedź" (set Liczba1 to answer), "zapytaj Podaj drugą liczbę i czekaj" (ask for second number), "ustaw Liczba2 na odpowiedź" (set Liczba2 to answer), "jeżeli Liczba1 > Liczba2 to" (if Liczba1 > Liczba2 then), "powiedz Większa liczba to: przez 2 s" (say "Większa liczba to: przez 2 s"), "powiedz Liczba1 przez 2 s" (say Liczba1 przez 2 s), "w przeciwnym razie" (otherwise), "powiedz Większa liczba to: przez 2 s" (say "Większa liczba to: przez 2 s"), "powiedz Liczba2 przez 2 s" (say Liczba2 przez 2 s).

Zadanie 5. Stosowanie instrukcji iteracyjnej for

Zmodyfikuj program z zadania 4, aby można było sprawdzić dziesięć par różnych liczb.

Wskazówki: Instrukcja iteracyjna w języku C++ ma postać:

```
for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;
```

Fragment programu z zastosowaniem instrukcji `for`:

Szkoła ponadgimnazjalna

Grażyna Koba, *Programujemy w języku C++*

```
9      for (i=0;i<10;i++)
10     {
11     cout << "Podaj pierwsza liczbe: ";
12     cin >> liczba1;
13     cout << "Podaj druga liczbe: ";
14     cin >> liczba2;
15     if(liczba1>liczba2)
16         cout << "wieksza jest: " << liczba1 << "\n";
17     else cout << "wieksza jest: " << liczba2 << "\n";
18     }
```

Zadanie 6. Stosowanie funkcji – definiowanie i wywołanie

Napisz program, w którym zdefiniujesz funkcję *RoznicaKwadrat* z dwoma parametrami *a* i *b* obliczającą różnicę kwadratów dwóch liczb całkowitych. Wywołaj funkcję w programie głównym z parametrami aktualnymi *x* i *y*.

Wskazówki:

W języku C++ wszystkie podprogramy nazywane są funkcjami. Dzielimy je na dwie grupy: funkcje niezwracające wartości i zwracające wartość. Funkcja zwracająca wartość wylicza wartość i odsyła tę wartość do funkcji wywołującej. Mówimy, że funkcja zwraca wartość.

Funkcje zwracające wartość muszą zawierać instrukcję `return` z wartością, która ma być zwrócona do funkcji wywołującej. Zwracana wartość może być stałą, zmienną lub wyrażeniem. Na przykład:

```
return a*b;
return Wynik;
```

W języku C++ funkcję można wywołać, umieszczając wynik w zmiennej. Na przykład:

```
Wynik=RoznicaKwadrat();
```

Funkcję można wykorzystać jako element wyrażenia lub instrukcji. Na przykład:

```
cout << RoznicaKwadrat();
```

```
1  #include <iostream>
2  using namespace std;
3  int x, y;
4  int RoznicaKwadrat(int a, int b)
5  {
6      return x*x - y*y;
7  }
8
9  int main()
10 {
11     cin >> x >> y;
12     cout << RoznicaKwadrat(x, y);
13     return 0;
14 }
```

Definicja funkcji

Wywołanie funkcji

Uwaga: Parametr to wartość przekazywana funkcji, a **wartość zwracana** to wartość przekazywana z funkcji.

Środowiska programowania dostępne bezpłatnie w Internecie:

- **Scratch 2.0**
<https://scratch.mit.edu/scratch2download/>
- **Kompilator C++**
<https://sourceforge.net/projects/orwelldevcpp/>