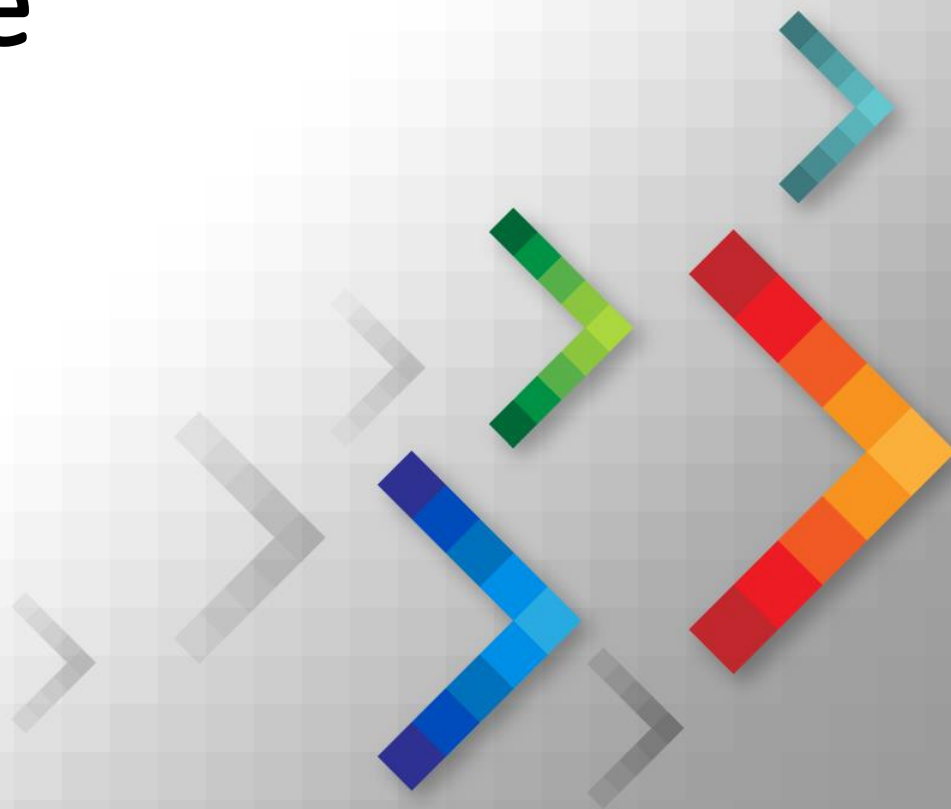


Programowanie w języku C++

Grażyna Koba



Kilka definicji:

Program komputerowy to ciąg instrukcji języka programowania, realizujący dany algorytm.

Język programowania to zbiór określonych instrukcji i zasad składni, używanych do zapisania tzw. **kodu źródłowego** programu.

Program może występować w dwóch postaciach:

- jako **program źródłowy** (postać zrozumiała dla programisty),
- jako **program wynikowy** (kod maszynowy, program wykonywalny) – zapisany w postaci ciągu instrukcji procesora, zrozumiały dla komputera.

Tłumaczenie programu źródłowego na kod maszynowy nazywamy **kompilacją**.

Aby utworzyć program:

1. Piszemy kod źródłowy programu (implementujemy program) – korzystając z edytora wbudowanego do środowiska programistycznego wybranego języka programowania
2. Zapisujemy program w pliku.
3. Kompilujemy program, zwykle korzystając z opcji **Compile** lub **Build**. Otworzy się okno z informacją o przebiegu kompilacji.
4. Jeśli program został skompilowany, uruchamiamy go, zazwyczaj korzystając z opcji **Run**. Program uruchomi się, zwykle w osobnym oknie.

```
#include <iostream>
```

Dołączenie biblioteki standardowej

```
using namespace std;
```

Informacja o korzystaniu z biblioteki standardowej

```
... // deklaracje stałych
```

```
... // deklaracje funkcji
```

```
int main ()
```

```
{ // część wykonawcza programu
```

```
    return 0;
```

```
}
```

Deklarowanie zmiennych

- W języku C++ małe i wielkie litery w nazwach instrukcji, zmiennych itp. mają różne znaczenie,
- W nazwach zmiennych można stosować litery (bez polskich znaków diakrytycznych), cyfry i znak podkreślenia. Nazwa nie może zaczynać się od cyfry.
- W języku C++ zmienne można deklarować w dowolnym miejscu, ale zawsze przed ich pierwszym użyciem.

C++

```
int liczba;  
int i, j, Suma;
```

C++

```
float Wartość;  
float a, b, X, Y;
```

Wprowadzanie i wyprowadzanie zmiennych

cin jest **obiektem** reprezentującym standardowe **wejście** programu.

Operator **>>** oznacza wprowadzenie danych do odpowiedniej zmiennej, podanej po prawej stronie.

C++

```
cin >> liczba;  
cin >> a >> b >> W;
```

cout jest obiektem reprezentującym **wyjście** programu.

Operator **<<** oznacza wyprowadzenie wartości podanej po prawej stronie..

C++

```
cout << P;  
cout << "S = " << a+b << "\n ";  
cout << "Suma wynosi:" << S << endl;
```

Instrukcja warunkowa

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

C++

określenie	C++
równy	==
różny	!=
mniejszy	<
mniejszy lub równy	<=
większy	>
większy lub równy	>=
alternatywa logiczna	
koniunkcja logiczna	&&
negacja logiczna	!

C++

```
a!=-1
x==0
```

```
w==5 || w==25
y>0 && y<100
```

Iteracja polega na wielokrotnym powtarzaniu tej samej operacji (ciągu operacji).

Iterację implementujemy, stosując tzw. **pętlę**. Z pętlą mamy do czynienia, gdy w pewnym kroku algorytmu wracamy do jednego z wcześniejszych kroków, co powoduje, że kroki te mogą zostać wykonane wiele razy.

<code>for (wyrażenie_początkowe; warunek; wyrażenie_pętli) instrukcja;</code>	C++
---	-----

```
for (i=0; i<n; i++)
{
    cin >> a;
    iloczyn*=a;
}
```


Instrukcja iteracyjna
`while`

```
while (warunek) instrukcja;
```

Działanie pętli **while**: najpierw sprawdzany jest warunek; jeśli jest on spełniony, to wykonywana jest *instrukcja*.

Wewnątrz bloku instrukcji powinna być zawsze umieszczona instrukcja, która zmienia wartość warunku – w przeciwnym wypadku pętla nigdy się nie zakończy.

W szczególnej sytuacji, gdy warunek od razu jest niespełniony, instrukcja może w ogóle nie zostać wykonana. Instrukcja może być pojedyncza lub złożona.

```
...  
cin >> a >> b;  
while(b!=0.0)  
{  
    iloras=a/b;  
    cout << "Iloras wynosi " << iloras << endl;  
    cin >> a >> b;  
}
```

i Instrukcje iteracyjne

W języku C++ wszystkie podprogramy nazywane są funkcjami. Dzielimy je na dwie grupy:

- funkcje niezwracające wartości (odpowiednik procedury w języku Pascal),
- funkcje zwracające wartość (odpowiednik funkcji w języku Pascal).

Definicja funkcji obejmuje:

- nagłówek (zawierający nazwę, typ zwracanej wartości i ewentualnie listę parametrów),
- treść (zawierającą instrukcje).

C++

```
opis_typu nazwa_funkcji (lista_parametrów_formalnych)
{
    instrukcje;
    return wartość;
}
```



Dziękuję za uwagę

Grażyna Koba
grazyna.koba@migra.pl