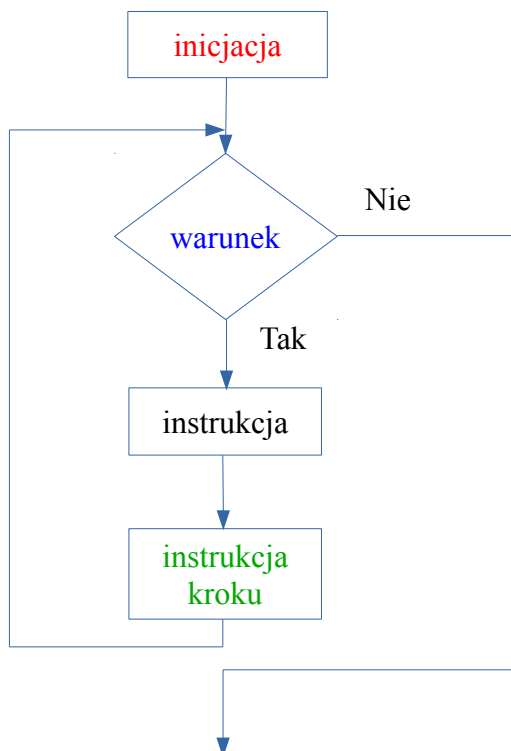


Pętle w C++

1. Pętla for



```
for (inicjacja ; warunek ; instrukcja kroku)
    instrukcja;
```

```
for (int i=1 ; i<=10 ; i++)
    cout << "*" ; // wypisze 10 gwiazdek
```

i++ inkrementacja (zwiększanie o 1)

i-- dekrementacja (zmniejszenie o 1)

```
int i;
for (i=1 ; i<=10 ; i=i+1)
    cout << i; // wypisze liczby od 1 do 10
```

```
int i,x;
x=1;
for (i=x ; i<=10 ; i=i+1)
    cout << i; // wypisze liczby od 1 do 10
```

Niebezpieczeństwa związane z pętlą for:

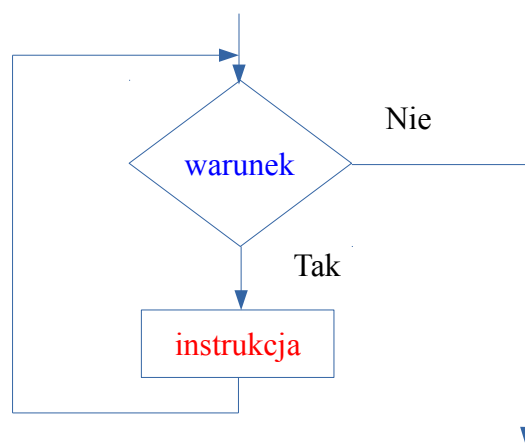
```
for (int i=1 ; i=0 ; i++)
    cout << "*" ; // ta pętla nie wykona się, ponieważ od
                  // początku warunek jest niespełniony
```

```
for (int i=1 ; i>10 ; i--)
    cout << i; // to jest przykład nieskończonej pętli, tzw.
                // zapętłonej, ponieważ „i” dąży do minus
                // nieskończoności, a warunkiem końca pętli
                // jest i>10, co nigdy nie nastąpi
```

Przykład pętli for, która wypisze sumę 5 dowolnych podanych liczb

```
...  
for (int i=1 ; i<=5 ; i++) {  
    cout << "Podaj " << i << "liczbę: ";  
    cin >> a;  
    s=s+a;  
}  
cout << "Suma pięciu liczb wynosi: " << s;  
...
```

2. Pętla while



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

```
k=1;  
while (k<=10)  
{  
    cout << "*" << endl;  
    k++;  
} // pętla wypisze 10 "*"
```

Pętle while należy rozumieć następująco:

Dopóki (lub podczas gdy) **warunek** jest **prawdziwy**, spełniony **wykonuj instrukcję**.

Trzeba zwrócić uwagę na to, że warunek jest sprawdzany przed wykonaniem instrukcji, stąd płynie wniosek, że pętla może nie wykonać się w ogóle, np. gdy w przykładzie obok „k” na początku będzie równe powiedzmy 11.

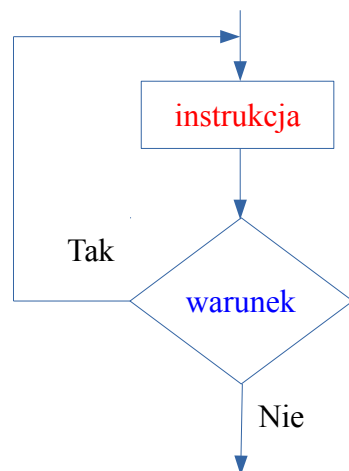
Ważne jest też to, że programista sam musi pamiętać o zmianie „k”. Jeśli zapomni o tym, to pętla może się zapętlić.

Co istotne pętla while wykona tylko jedną instrukcję, wobec czego niezbędne są nawiasy klamrowe {}

Przykład: chcemy obliczyć sumę nieokreśloną ilość liczb. Sumowanie kończymy, gdy użytkownik poda 0.

```
liczba=1;
k=0;
suma=0;
while (liczba !=0)
{
    cout << "Podaj liczbę: ";
    cin >> liczba;
    suma=suma+liczba;
    k++;
}
cout << "Suma " << k << " liczb równa się: " << suma << endl;
```

3. Pętla do ... while



```
do
    instrukcja;
whlie (warunek);
```

```
k=1;
do {
    cout << "*" ;
    k++;
}
while (k<=10);
```

Różnica między pętlą while, a do ... while polega na tym, że w tej drugiej warunek końca pętli sprawdzany jest po wykonaniu instrukcji, dlatego też, mamy pewność, że pętla do ... while wykona się przynajmniej jeden raz. Na przykład k=11 spowoduje, że program wypisze jedną gwiazdkę. Jest to subtelna różnica, ale niekiedy bardzo ważna.