Wstęp do programowania - wykład 7

dr Piotr Jastrzębski

Pętla for

```
for(w1; w2; w3)
{
    instrukcje;
}
```

- w1 - wyrażenie inicjujące
- w2 - warunek logiczny
- w3 - wyrażenie modyfikujące

```cpp
#include <iostream>
using namespace std;
int main()
{
    for(int i=1;i<5;i++)
    {
        cout << i <endl;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    for(i=1;i<5;i++)
    {
        cout << i <endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    for(int i=5;i>0;i--)
    {
        cout <<i <<endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    for(int i=1;i*i<100;i+=2)
    {
        cout << i <<endl;
    }
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    for(;;)
    {
        cout << "endless loop!";
    }
    return 0;
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    for(int n = 0; n < 10; ++n, cout << n <<endl)
        ;
    return 0;
}
```

Pętla while i do ...while

```
do {
   /* instrukcje do wykonania w pętli */
} while (warunek);
```

```
while (warunek) {
   /* instrukcje do wykonania w pętli */
}
```

```cpp
#include <iostream>
using namespace std;
int main()
{
    int i =0;
    while (i<5)
    {
        cout << i <<endl;
        i++;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int i =0;
    while (i<5)
    {
        cout <<i <<endl;
        if (i>2)
        {
            break;
        }
        i++;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int i =0;
    while (i<5)
    {
        i++;
        if (i==2)
        {
            continue;
        }
        cout <<i <<endl;
    }
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main()
{
    int i =0;
    do
    {
        cout <<i <<endl;
        i++;
    }
    while (i<-2);
    return 0;
}
```

Zasięg zmiennej

- globalne - zadeklarowane poza `main`, dostępne dla wszystkich funkcji.
- lokalne - zadeklarowane w innym miejscu

```cpp
#include <iostream>
using namespace std;

int a = 1; /* zmienna globalna */
int b = 3; /* zmienna globalna */
int main()
{
    int a = 2;          /* to już zmienna lokalna */
    cout << a << endl;
    cout << b << endl;
}
```

Czas życia

Czas życia to czas od momentu przydzielenia dla zmiennej miejsca w pamięci (stworzenie obiektu) do momentu zwolnienia miejsca w pamięci (likwidacja obiektu).

```
#include <iostream>
using namespace std;

int main()
{
    int a = 10;
    {           /* otwarcie lokalnego bloku */
        int b = 10;
        cout << a << " " << b << endl;
    }           /* zamknięcie lokalnego bloku */
    cout << a << " " << b << endl;  /* BŁĄD: b juz nie ist
}
```

Stałe

Stała, różni się od zmiennej tylko tym, że nie można jej przypisać innej wartości w trakcie działania programu. Wartość stałej ustala się w kodzie programu i nigdy ona nie ulega zmianie.

```
const typ nazwa_stałej=wartość;
```

```cpp
#include <iostream>
using namespace std;

int main()
{
    const int a = 7;
    a++;
}
```

#define

`#define` - dyrektywa preprocessora.

Linia pozwalająca zdefiniować stałą, funkcję, słowo kluczowe lub makro, które będzie potem podmienione w kodzie programu na odpowiednią wartość lub może zostać użyte w instrukcjach warunkowych dla preprocesora.

```
#define NAZWA_STALEJ WARTOSC
```

```
#define NAZWA_STALEJ
```

```
#include <iostream>
#define SIX 1+5
#define NINE 8+1

using namespace std;
int main()
{
    cout << SIX << "*" << NINE
        << "=" << SIX * NINE << endl;
    return 0;
}
```

```cpp
#include <iostream>
#define for10 for(int i=0;i<10;i++)
using namespace std;

int main()
{
    for10
    {
        cout << i << endl;
    }
    return 0;
}
```