

## 2.7. Pętle (instrukcje iteracyjne)

Instrukcje iteracyjne powodują powtarzanie ciągu poleceń, dopóki spełniony jest określony warunek logiczny. Po wykonaniu ciągu poleceń określoną liczbę razy lub jeśli podany warunek nie jest spełniony, wykonanie pętli powinno się zakończyć. W języku C++ występują następujące rodzaje pętli: `for`, `while`, `do...while`.

### 2.7.1. Pętla `for`

Instrukcja `for` służy do definiowania pętli, jeśli został podany licznik wykonań pętli i warunek, który musi być spełniony, aby kolejny raz ją wykonać. Składnia instrukcji jest następująca:

```
for (wyrażenie początkowe; wyrażenie warunkowe; wyrażenie modyfikujące)
{
    blok instrukcji;
}
```

- *wyrażenie początkowe* — określa wartość początkową zmiennej, która jest używana jako licznik pętli;
- *wyrażenie warunkowe* — określa warunek, który musi być spełniony, aby pętla została wykonana kolejny raz;
- *wyrażenie modyfikujące* — modyfikuje zmienną, która jest licznikiem.

Pętłę `for` wykorzystuje się wtedy, gdy znana jest liczba wykonywanych powtórzeń.

W najprostszej postaci instrukcja `for` może służyć do wyświetlenia kolejnych liczb od 1 do wartości wprowadzonej przez użytkownika lub do obliczania sumy tych liczb.

#### Przykład 2.48

```
#include <iostream>
using namespace std;
int main()
{
    int zakres, i;
    cout << "Program wyświetli kolejne liczby od 1 do podanej liczby."
    << endl;
    cout << "Podaj liczbę końcową zakresu: ";
    cin >> zakres;
    for (i = 1; i <= zakres; i++)
        cout << i << endl;
    return 0;
}
```

**Przykład 2.49**

```

#include <iostream>
using namespace std;
int main()
{
    int zakres, suma, i;
    cout << "Program liczy sume kolejnych liczb od 1 do podanej liczby\n";
    cout << endl;
    cout << "Podaj liczbe koncowa zakresu: ";
    cin >> zakres;
    suma = 0;
    for (i = 1; i <= zakres; i++)
    {
        suma += i;
    }
    cout << "Suma liczb od 1 do " << zakres << " wynosi " << suma << endl;
    return 0;
}

```

W podanym przykładzie w bloku `for` występuje jedna instrukcja: `suma += i;` wykonywana w każdym przebiegu pętli i powoduje zwiększanie wartości zmiennej `suma` o 1. Pętla wykonywana jest dopóki, dopóty spełniony jest warunek `i <= zakres`, w którym wartość zmiennej `zakres` jest ustawiana przez użytkownika i określa zakres sumowanych liczb.

Modyfikacja licznika w instrukcji `for` może polegać na jego zwiększaniu (inkrementacja) lub zmniejszaniu (dekrementacja). Można na nim wykonywać mnożenie, dzielenie i bardziej zaawansowane działania.

**Przykład 2.50**

```

#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i = 0; i <= 20; i+=2)
    {
        cout << i << endl;
    }
    return 0;
}

```

Program wyświetli kolejne liczby parzyste od 0 do 20.

### Przykład 2.51

```
#include <iostream>
using namespace std;
int main()
{
    int i;
    for (i = 20; i >= 0; i--)
    {
        cout << i << endl;
    }
    return 0;
}
```

Program wyświetli liczby od 0 do 20 w odwrotnej kolejności.

Wartość początkowa i wartość końcowa zmiennej wykorzystywanej jako licznik mogą być nieustalone podczas pisania kodu programu i mogą zostać ustalone dopiero podczas jego wykonywania.

### Przykład 2.52

```
#include <iostream>
using namespace std;
int main()
{
    int i, zakres1, zakres2;
    cout << "Program wyświetli kolejne liczby parzyste." << endl;
    cout << "Podaj początkowy zakres liczb: ";
    cin >> zakres1;
    cout << "Podaj końcowy zakres liczb: ";
    cin >> zakres2;
    for (i = zakres1; i <= zakres2; i+=2)
    {
        cout << i << endl;
    }
    return 0;
}
```

W uproszczonej postaci pętla może zostać pozbawiona wyrażenia modyfikującego.