

## Цикли.

Повтарят блок от код определен/неопределен брой пъти.

- Цикъл с **предусловие**

1. Проверява се **<условието>**, ако е **вярно** - стъпка **2**.  
Ако **не е вярно** - тялото се пропуска (**цикъла приключва**).
2. Изпълнява се **<тялото>** на цикъла.
3. Връща се на стъпка **1**.

Синтаксис:

```
while (<условие>) {  
    <тяло>  
}
```

// Забележка: Фигурните скоби, при всички типове цикли,  
// не са задължителни когато тялото на цикъла се състои от  
// само една "команда" (израз, завършващ с ; ) .

- Цикъл със **следусловие**

1. Изпълнява се **<тялото>** на цикъла.
2. Проверява се **<условието>**, ако е **вярно** - стъпка **1**.  
Ако условието **не е вярно** - **цикъла приключва**.

Синтаксис:

```
do {  
    <тяло>  
} while (<условие>);
```

- Цикъл **for**

1. Изпълнява се **<инициализирането на променливи>**.
2. Проверява се **<условието>**, ако е **вярно** - стъпка **3**.  
Ако **не е вярно** - тялото се пропуска (цикъла приключва).
3. Тялото на цикъла се изпълнява.
4. Изпълнява се **<промяната/стъпката>**.
5. Връща се на стъпка **2**.

Синтаксис:

```
for (<инициализация>; <условие>; <промяна/стъпка>) {  
    <тяло>  
}
```

## Контрол над изпълнението на цикъла.

- `break`;

При достигане на `break` в тялото на цикъл, то се прескача целия код в тялото на цикъла, който се намира след `break` и цикъла приключва.

- `continue`;

При достигане на `continue` в тялото на цикъл, кода в тялото на цикъла след `continue` се пропуска и изпълнението на цикъла се връща на `<условието>` при `while` и `do while`, и на `<промяната>` при `for`.

*Примери:*

```
for (int i = 1; i < 10; i++)  
    cout << i << ' ';
```

// Отпечатва на конзолата: 1 2 3 4 5 6 7 8 9

```
int num;  
long long sum = 0;  
do {  
    cin >> num;  
    sum += num;  
} while (num != 0);  
cout << sum << endl;
```

// Чете цели числа от конзолата до въвеждането на 0

// и извежда сумата на всичките числа