

# Циклы в Java

---

**Циклы** позволяют выполнять однотипное действие несколько раз. Каждое повторение цикла называется **итерацией**.

### Цикл с предусловием `while`

**Условие** будет проверяться перед каждым выполнением тела цикла, если оно истинно, тело цикла будет выполняться. Так будет происходить, пока условие истинно, когда условие станет ложным, программы выйдет из цикла.

**Если условие изначально ложно**, тела цикла не будет выполнено ни разу.

**Если условие всегда истинно**, цикл будет продолжаться бесконечно.

Синтаксис,

```
while(условие) { // проверка условия  
    тело цикла выполняется, если условие истинно  
} // если условие ложно, программа выходит из цикла и продолжает работу  
код после цикла;
```

## Цикл с предусловием `while` (пример)

```
int count = 3;
while (count > 0) {
    System.out.println(count);
    count--;
}
```

Первая проверка условия: `count` = 3, значит условие истинно и тело цикла выполнится.

**Итерация 1** (первое выполнение тела цикла):

1. в консоль будет выведено: 3
2. уменьшение значения `count` на 1, значит `count` будет равен 2

**Вторая проверка условия:** `count` = 2 (после уменьшения в теле цикла), значит условие истинно и тело цикла выполнится

**Итерация 2** (второе выполнение тела цикла):

1. в консоль будет выведено: 2
2. уменьшение значения `count` на 1, значит `count` будет равен 1

**Третья проверка условия:** `count` = 1 (после уменьшения в теле цикла), значит условие истинно и тело цикла выполнится

**Итерация 3** (второе выполнение тела цикла):

1. в консоль будет выведено: 1
2. уменьшение значения `count` на 1, значит `count` будет равен 0

**Четвертая проверка условия:** `count` = 0 (после уменьшения в теле цикла), значит **условие ложно** и **тело цикла не будет выполнено**, программа выйдет из цикла.

## Цикл с постусловием do...while

**Сначала будет выполняться тело цикла, а затем проверяться условие**, если оно истинно, тело цикла будет выполняться еще раз. Так будет происходить, пока условие истинно, когда условие станет ложным, программы выйдет из цикла.

**Если условие изначально ложно**, тела цикла выполняется один раз, т.к. условие проверяется после выполнения тела цикла.

**Если условие всегда истинно**, цикл будет продолжаться бесконечно.

Синтаксис,

```
do {
```

```
    тело цикла выполнится первый раз в любом случае,  
    далее будет выполняться, если условие истинно
```

```
} while (условие); // проверка условия, если условие ложно, программа выходит из  
    цикла и продолжает работу
```

```
код после цикла;
```

## Цикл с постусловием do...while (пример)

```
int count = 2;  
do { // тело цикла:  
    System.out.println(count); // вывод в консоль: значение count  
    count; // уменьшение значения count на 1  
} while (count > 0); // проверка условия: условие истинно, пока значение count больше 0  
// условие станет ложным, когда значение count будет равно 0
```

**Итерация 1** (первое повторение тела цикла):

1. в консоль будет выведено: 2
2. уменьшение значения **count** на 1, значит **count** будет равен 1

**Первая проверка условия:** **count** = 1, значит условие истинно и тело цикла выполнится.

**Итерация 2** (второе повторение тела цикла):

1. в консоль будет выведено: 1
2. уменьшение значения **count** на 1, значит **count** будет равен 0

**Вторая проверка условия:** **count** = 0, значит условие ложно и тело цикла выполняться не будет, программа выйдет из цикла

## Цикл for

состоит из выполнения трех операций

Синтаксис,

```
for ( [ начало-инициализация ]; [ условие ]; [ шаг ] ) {  
    тело цикла выполняется, если условие истинно  
}
```

[ ] - квадратные скобки в описании используются для обозначения необязательных параметров

1. **начало-инициализация** - выполняется один раз, при заходе в цикл. Обычно это выражение инициализирует один или несколько счётчиков. Также используется для объявления переменных.
2. **условие** - проверяется при входе в цикл и далее перед каждой итерацией, если оно истинно, **тело цикла** выполняется, если ложно, программа выходит из цикла. Если **условие** пропущено, оно считается истинным.
3. **шаг** - выполняется после выполнения тела цикла на каждой итерации, но перед проверкой условия. Обычно, обновление счетчика (уменьшение/увеличение).

## Цикл for (пример)

```
int count = 3;

for (int i = 0; i < count; i++) {
    System.out.println(count);
    count--;
}
```

### Начало цикла:

int i = 0 - инициализация счетчика

### Первая проверка условия: $i < \text{count}$ ,

значение  $i$  равно 0, а значение **count** равно 3, получаем  $0 < 3$ , значит условие истинно и тело цикла выполнится

### Итерация 1 (первое выполнение тела цикла):

1. в консоль будет выведено: 3
2. уменьшение значения **count** на 1, значит **count** будет равен 2

### Первое обновление счетчика (шаг):

$i++$  - значение  $i$  станет равно 1

### Вторая проверка условия: $i < \text{count}$ ,

значение  $i$  равно 1(после первого обновления счетчика), а **count** равно 2 (после уменьшения в теле цикла), получаем  $1 < 2$ , значит условие истинно и тело цикла выполнится.

### Итерация 2 (второе выполнение тела цикла):

1. в консоль будет выведено: 2
2. уменьшение значения **count** на 1, значит **count** будет равен 1

### Второе обновление счетчика (шаг):

$i++$ ; значение  $i$  станет равно 2

### Третья проверка условия: $i < \text{count}$ ,

значение  $i$  равно 2(после второго обновления счетчика), а значение **count** равно 1 (после уменьшения в теле цикла), получаем  $2 < 1$ , значит условие ложно и тело не выполнится, программа выйдет из цикла

# Директивы break и continue

## Прерывание цикла break

позволяет выйти из цикла в любой момент и продолжить выполнение кода после цикла.

Синтаксис,

```
while(условие) { // проверка условия
```

```
    тело цикла выполняется, если условие истинно
```

```
    if (условие2) { break; } // если условие2 истинно, программа выходит  
                             из цикла благодаря директиве break
```

```
}
```

```
код после цикла;
```



# Директивы break и continue

## Следующая итерация continue

прекращает выполнение текущей итерации цикла.

Синтаксис и пример,

```
for (int i = 0; i < 10; i++) {  
    if (i % 2 == 0) continue; // если i - четное число,  
                             // завершает текущую итерацию,  
                             // цикл переходит к проверки условия i < 10  
    System.out.println(i); // выводит значение i  
}
```

