

Ветвления и циклы в Java

Евгений Беркунский,
ст. преподаватель кафедры ИУСТ,
Национальный университет
кораблестроения

eugeny.berkunsky@gmail.com

<http://berkut.homelinux.com>



Три конструкции, реализующие ветвление:

- Условный оператор **if... else**
- Оператор выбора **switch...case**
- Тернарный условный оператор **?:**

Условный оператор if.. else

```
if (условие) {  
    последовательность операторов 1  
} else {  
    последовательность операторов 2  
}
```

Условный оператор if.. else

Например:

```
if (a > b) {  
    max = a; min = b;  
} else {  
    min = a; max = b;  
}
```

Оператор выбора switch...case

```
switch (значение) {  
    case const1:  
        последовательность операторов 1;  
        break;  
    case const2:  
        последовательность операторов 2;  
        break;  
    .....  
    default:  
        последовательность операторов;  
}
```

Оператор выбора switch...case

Например:

```
switch (n) {  
    case 1:  
        System.out.println("Мало");  
        break;  
    case 2: case 3:  
        System.out.println("Норма");  
        break;  
  
    default:  
        System.out.println("Столько не бывает");  
}
```

Тернарный оператор ?:

Используется непосредственно в выражениях.

Например:

```
int x = a > b ? a : b;
```

Или так:

```
System.out.println(a>b ? a : b);
```

Три конструкции, реализующие циклы:

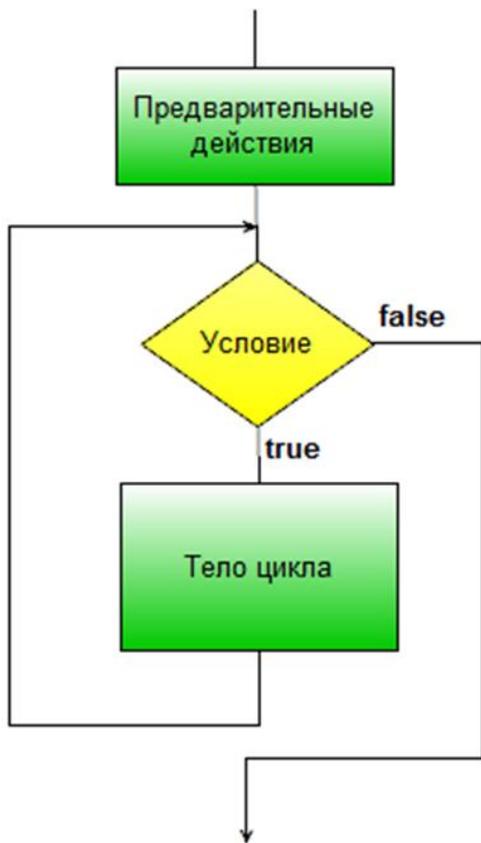
- Цикл **while...**
- Цикл **do...while**
- Цикл **for**

Кроме того, есть форма цикла **for** для итераций по массивам и коллекциям (рассмотрим позже)

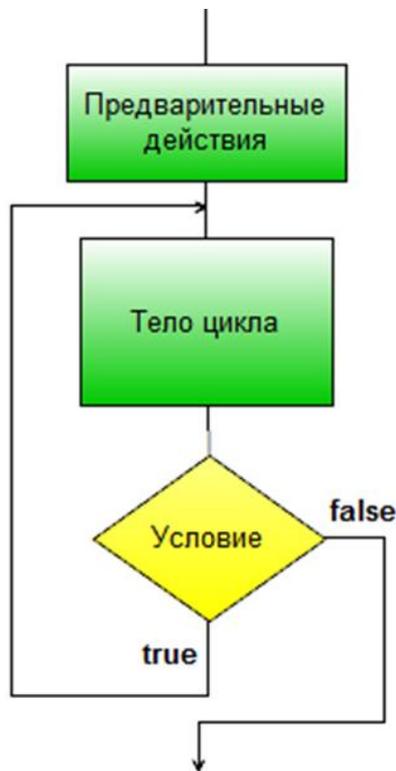


Циклы while и do...while

Цикл while



Цикл do...while



```
.....  
while (условие) {  
    действия в цикле  
}
```

```
.....  
do {  
    действия в цикле  
} while (условие);
```

Цикл for

```
for ( /*выражение1*/; /*выражение2*/; /*выражение3*/ ) {  
    /*один оператор или блок операторов*/;  
}
```

Выражение 1 - объявление (и) или инициализация, ранее объявленной, переменной-счетчика, которая будет отвечать за истинность условия в цикле

```
int counter = 0;
```

```
counter = 9;
```

```
int counter;
```

Выражение 2 - это условие продолжения цикла for, оно проверяется на истинность.

```
counter < 10; // условие истинно пока count меньше 10
```

Выражение 3 изменяет значение переменной-счетчика. Без него цикл считается бесконечным, так как изменение содержимого переменной count выполняться не будет, и если изначально условие было истинным, то цикл будет бесконечным, иначе программа даже не войдет в цикл.

```
counter++ //
```

Цикл for

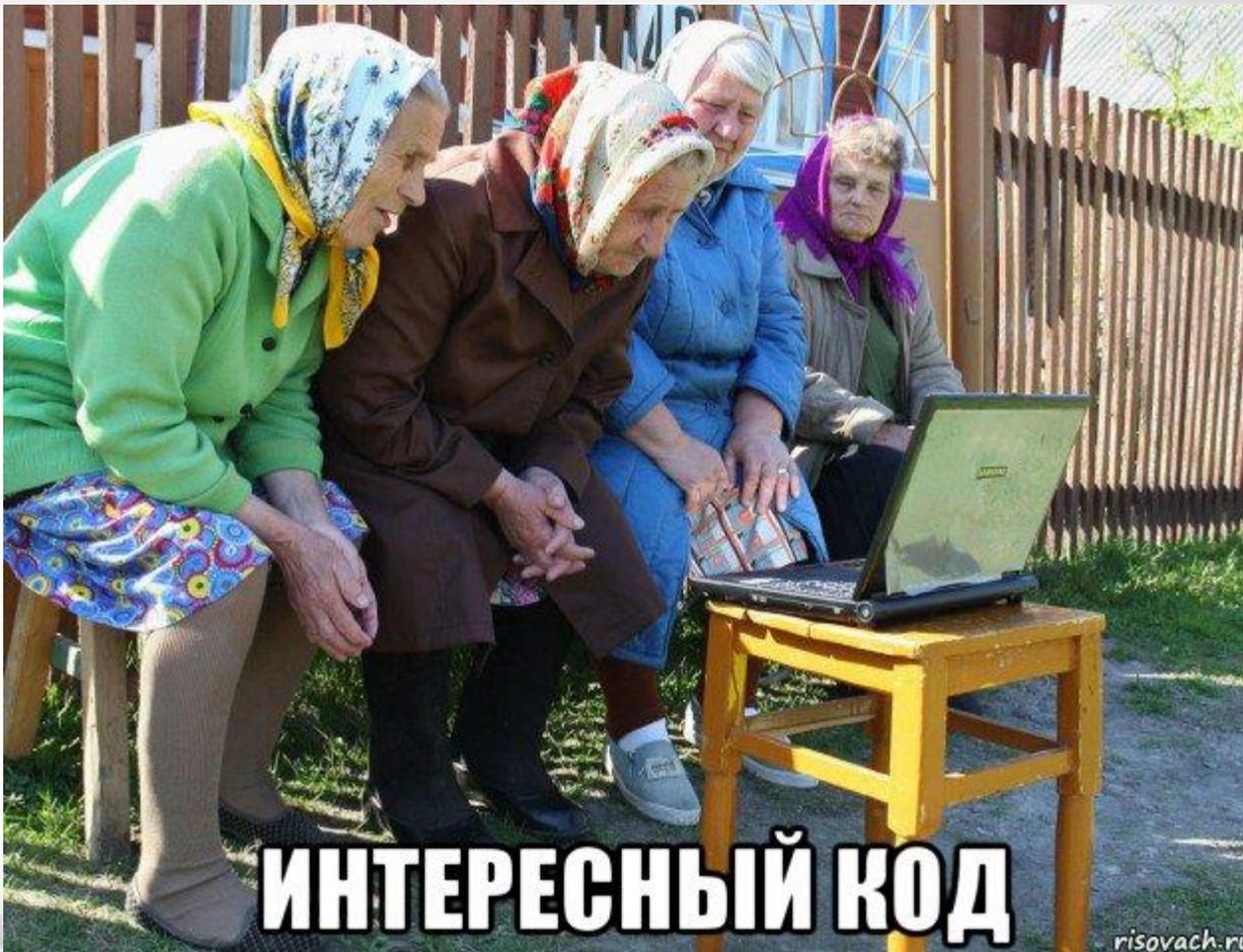
```
for (int counter = 0; counter < 15; counter++) {  
    .....  
}
```

Шаг в цикле for может быть отличным от единицы,
а точнее, любым целым(!) числом





Демонстрація



ИНТЕРЕСНЫЙ КОД

risovach.ru



Вопросы?



http://berkut.homelinux.com/?item=z_contact