

# Циклы и массивы

Автор: Юлий Слабко

# Рассматриваемые вопросы



- Понятие цикл
- Типы циклов
- Циклы (for, while, for-each, do - while)
- Операторы continue и break
- Массивы
- Одномерные массивы (создание, инициализация)
- Сортировка массива
- Многомерные массивы

# Циклы в java

**Цикл** — это многократно повторяющийся фрагмент программы.

В Java существует два типа циклов

- тип «пока»
- тип «n-раз»



Первый тип «пока» предназначен для повторения какого-то действия до тех пор, пока выполняется некоторое условие. Пример: увеличивать число на 5 до тех пор, пока оно не станет трёхзначным.

Второй тип «n-раз» предназначен для повторения каких-то действий заранее известное количество раз. Пример: умножить число само на себя 4 раза.

# Цикл тип “пока”

Представителями цикла типа “пока” являются циклы **while** и **do...while**

## Общий вид цикла **while**

```
while (<условие>) {  
    оператор;  
    .....  
}
```

## Общий вид цикла **do...while**

```
do {  
    оператор;  
    .....  
} while (<условие>;
```

# Цикл while

Цикл повторяется до тех пор, пока условие верно.

В случае, если условие изначально не верно, цикл ни разу не выполнится.

## Общий вид цикла

[инициализация счетчика]

```
while (<условие>) {
```

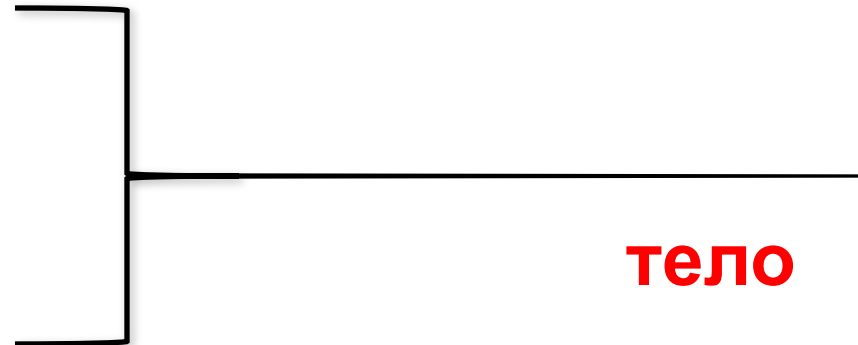
```
  оператор;
```

```
  .....
```

```
  цикла
```

```
  [увеличение (уменьшение) счетчика]
```

```
}
```



# Пример цикла while

```
int i = 1;
while (i < 5) {
    System.out.println("i = " + i);
    i++;
}
```

---

Вывод:

```
i = 1
i = 2
i = 3
i = 4
```

# Цикл do...while

Обязательно выполнится хотя бы один раз.

## Общий вид цикла

[инициализация счетчика]

```
do {  
    оператор;  
    .....  
    [увеличение (уменьшение) счетчика]  
} while (<условие>);
```

# Пример do...while

```
int n = 5;  
do {  
    System.out.println("Sample : " + n);  
    n--;  
} while (n > 0);
```

---

Вывод:

```
Sample : 5  
Sample : 4  
Sample : 3  
Sample : 2  
Sample : 1
```



# Пример

```
int sum = 0;
int i = 0;
while (i < 10) {
    if (i % 2 == 0) {
        sum += i;
    }
    i++;
}
System.out.print("sum = " + sum);
```



Чему равно значение переменной *sum*?

# Цикл For



Представителем цикла типа “n-раз” являются цикл *for*.

## Общий вид цикла for

```
for (инициализация; условие; итерация) {  
    оператор;  
    .....  
}
```

Алгоритм выполнения цикла:

1. Инициализация счетчика
2. Проверка условия
3. Выполнение тела цикла
4. Итерация
5. Повторение шагов 2-5

# Пример цикла For

```
for (int i = 0; i < 5; i++){  
    System.out.println("Count is: " + i);  
}
```

---

Вывод:

```
Count is : 0  
Count is : 1  
Count is : 2  
Count is : 3  
Count is : 4
```

# Continue и break

# Оператор break

По мере необходимости выполнение цикла можно немедленно прекратить его. Для этой цели используется выражение **break**. Если оно встречается в цикле, то цикл завершается и управление передается выражению, следующему после цикла.

**Задача:** Найти первое число из отрезка [17,28], которое нацело делится на 13;

```
int i = 17;
while (i <= 28){
    if (i % 13 == 0){
        break;
    }
    i++;
}
System.out.println("First number is : " + i);
```

# Оператор continue



Существует возможность преждевременно завершить очередную итерацию цикла, нарушив нормальный ход выполнения команд. Это позволяет сделать оператор **continue**. Данный оператор вызывает принудительный переход на следующую итерацию цикла, причем все последующие операторы игнорируются.

```
for (int i = 1; i < 5; i++) {  
    if (i == 3) {  
        System.out.println("continue!");  
        continue;  
    }  
    System.out.println("i = " + i);  
}
```

# Массивы

# Массив

**Массив** — это конечная последовательность упорядоченных по индексу элементов одного типа, доступ к каждому элементу в которой осуществляется по этому индексу.

**Размер или длина массива** — это общее количество элементов в массиве. Размер массива задаётся при создании массива и не может быть изменён в дальнейшем

Массивы бывают:

- одномерными
- многомерными (двух, трех ...)



# Одномерный Массив

Объявление массивов:

```
тип [] имя;
```

```
тип имя[];
```

Примеры:

```
int[] array;
```

```
int array[];
```

Инициализация массивов:

```
имя = new тип[размер];  
int[10];
```

Примеры:

```
array = new
```

Начальная инициализация массива:

```
int[] array = {1,2,3,4,5};
```

# Одномерный Массив

Доступ к элементам массива:

```
array[0]; array[1];
```

Первый индекс – 0



Вычисление длины массива:

```
array.length – размер массива array
```

# Пример

```
int[] array = new int[10];
Random rand = new Random();
for (int i = 0; i < array.length; i++) {
    array[i] = rand.nextInt(10);
}
System.out.println("initial array:");
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}

for (int i = 0; i < array.length; i++) {
    array[i] += 10;
}
System.out.println("new array:");
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
```

# Сортировка массивов

- 1.** Сортировка выбором
- 2.** Сортировки с помощью обменов
  - а) Пузырьковая сортировка
  - б) Шейкерная сортировка
- 3.** Сортировка с помощью включения
- 4.** Сортировка слиянием
- 5.** Сортировка с помощью разделения

[Реализация алгоритмов](#)

[Сортировки](#)  
[Визуализация алгоритмов](#)

[Сортировки](#)

# Сортировка выбором

```
int[] array = {8, 5, 2, 6, 9, 3, 1, 4, 0, 7};
```

```
for (int i = 0; i < array.length; i++) {
```

```
    int minIndex = i;
```

```
    for (int j = i + 1; j < array.length; j++) {
```

```
        if (array[j] < array[minIndex]) {
```

```
            minIndex = j;
```

```
        }
```

```
    }
```

```
    int temp = array[i];
```

```
    array[i] = array[minIndex];
```

```
    array[minIndex] = temp;
```

```
}
```

Нахождение индекса  
минимального  
элемента

Меняем текущий элемент  
с минимальным

```
for (int i = 0; i < array.length; i++) {
```

```
    System.out.print(array[i] + " ");
```

```
}
```

Вывод отсортированного  
массива

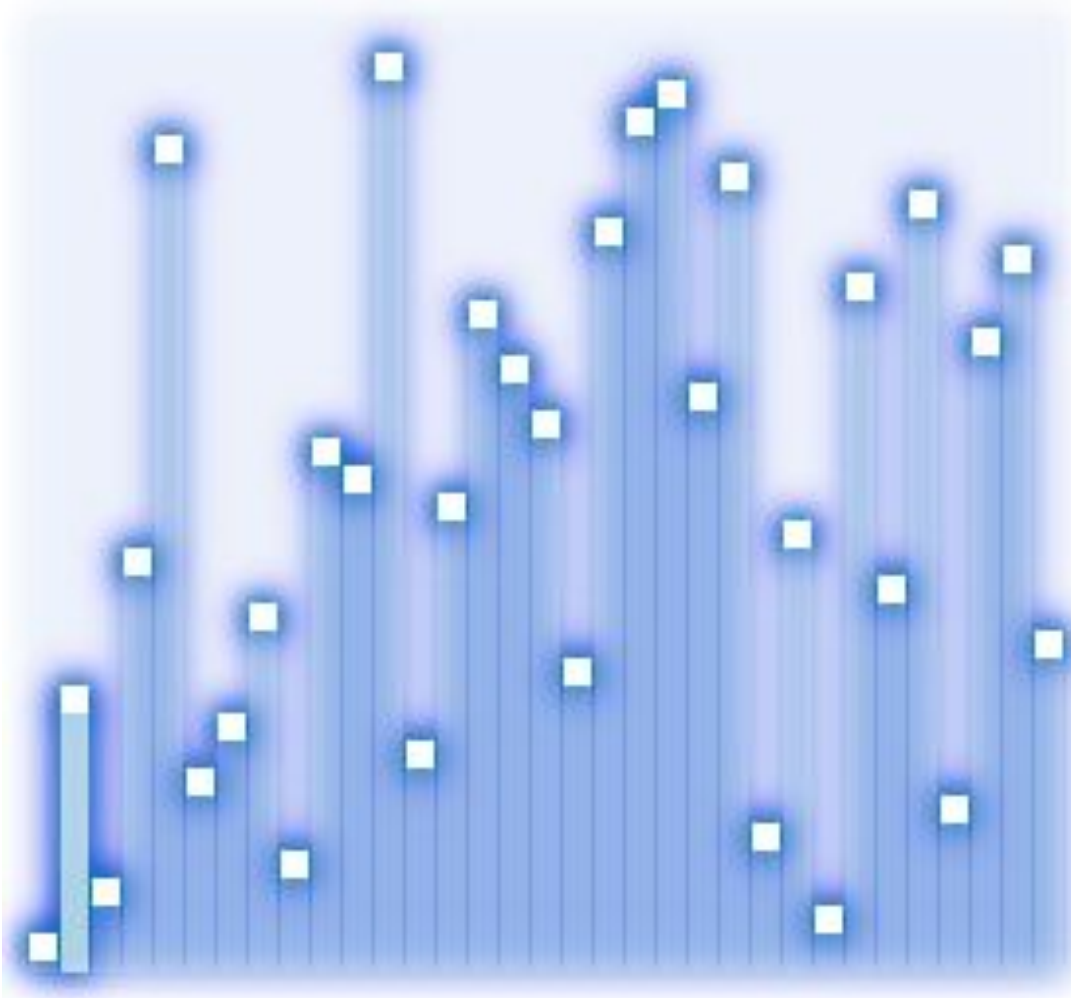
8
5
2
6
9
3
1
4
0
7

# Пузырьковая сортировка

5 4 3 1 2 | 

```
int[] array = {5, 4, 3, 1, 2};
for (int i = array.length - 1; i > 0; i--) {
    for (int j = 0; j < i; j++) {
        if (array[j] > array[j + 1]) {
            int temp = array[j];
            array[j] = array[j + 1];
            array[j + 1] = temp;
        }
    }
}
for (int i = 0; i < array.length; i++) {
    System.out.print(array[i] + " ");
}
```

# Шейкерная сортировка



# Многомерные массивы

Объявление и инициализация многомерного массива

```
int[][] array = new int[3][3];
```

Начальная инициализация массива

```
int[][] array = {  
    {1, 2, 3}, {4, 5, 6}, {7, 8, 9}  
};
```

Доступ к элементам многомерного массива

`array[i][j]` – где *i* и *j* индексы



# Пример

```
int[][] array = {
    {1, 1, 1, 1}, {1, 1, 1, 1}, {1, 1, 1,
1}
};
int sum = 0;
for (int i = 0; i < array.length; i++) {
    for (int j = 0; j < array[0].length; j++)
    {
        sum += array[i][j];
    }
}
System.out.println(sum);
```

# Вопросы



# Спасибо за внимание