

# Основы программирования на языке Java. Уровень 1.

## Занятие 3

Степулёнок Денис Олегович — [denis.stepulenok@oracle.com](mailto:denis.stepulenok@oracle.com)  
Солодкая Анастасия Сергеевна — [a.s.solodkaya@gmail.com](mailto:a.s.solodkaya@gmail.com)

19 октября 2016 года

# Программа (агенда) занятия 3 I

- Оперативная память компьютера
- Представление массива данных в памяти
- Объявление и инициализация массивов
- Алгоритм программы генерации массива со случайными числами
- Реализация программы
- Алгоритм поиска минимума/максимума в массиве
- Реализация алгоритмов поиска; вложенность циклов
- Простейший алгоритм сортировки массива
- Реализация алгоритма сортировки

# Оперативная память компьютера I

**Оперативная память** (Random Access Memory, RAM, память с произвольным доступом; ОЗУ (оперативное запоминающее устройство); оперативка) — энергозависимая часть системы компьютерной памяти, в которой во время работы компьютера хранится выполняемый машинный код (программы), а также входные, выходные и промежуточные данные, обрабатываемые процессором.

# Представление массива данных в памяти I

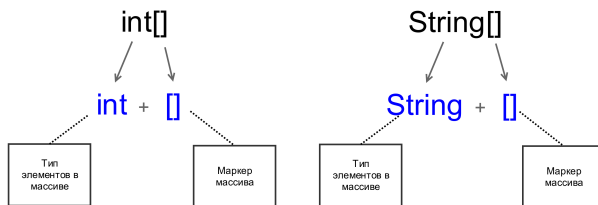
**Массив** — пронумерованная последовательность величин одинакового типа, обозначаемая одним именем.

Элементы массива располагаются в последовательных ячейках памяти, обозначаются именем массива и индексом.

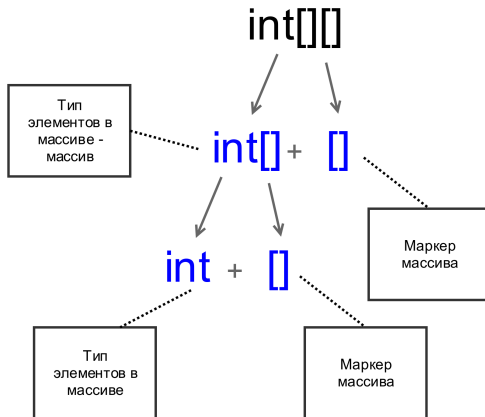
Каждое из значений, составляющих массив, называется элементом массива.

# Из чего состоит объявление массива? I

Тип элементов в массиве + [].



# Объявление многомерного массива I



# Инициализация I

Используем **new** - потому что массивы считаются **объектами**.

**Синтаксический сахар** - можно убрать **new** в случае, когда мы объявляем массив "на месте".

`new int[5]`

`new int[] {1, 7, 8, 2, 1}`

сокращение



`{1, 7, 8, 2, 1}`

`new int[5][2]`

`new int[5][]`

`new int[][] { {1, 3}, {6, 1, 4 }, {15} }`

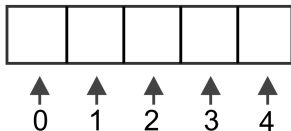
сокращение



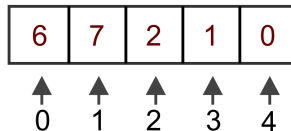
`{{1, 3}, {6, 1, 4 }, {15} }`

# Инициализация массива int "на месте" I

Длина 5  
Элементы от 0 до 4



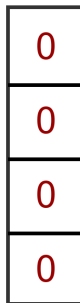
```
int[] t = {6, 7, 2, 1, 0};
```





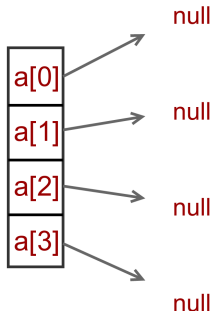
# Инициализация массива примитивов "по умолчанию" |

```
int[] a = new int[4]
```



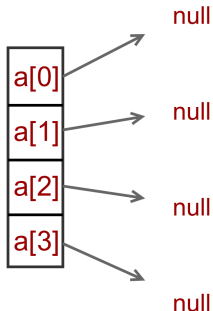
# Инициализация массива объектов "по умолчанию" |

```
String[] a = new String[4]
```

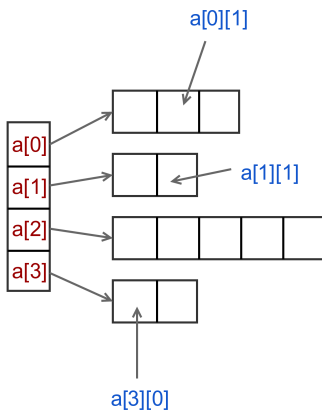


# Инициализация массива массивов "по умолчанию" I

Массив массивов  
`int[][] a = new int[4][]`



# Инициализация массива массивов "по умолчанию" I

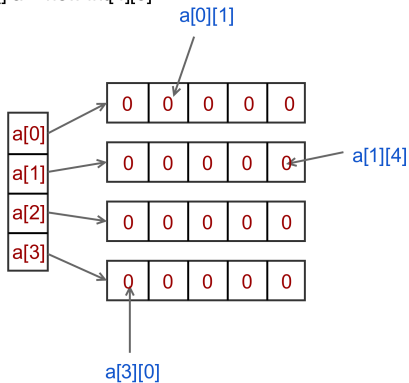


Массив массивов

```
int[][] a = new int[4][];  
a[0] = new int[3];  
a[1] = new int[2];  
a[2] = new int[5];  
a[3] = new int[2];
```

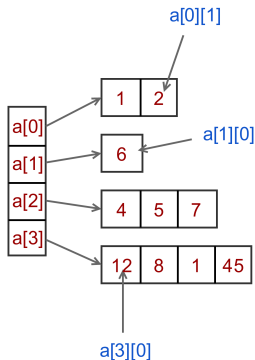
# Инициализация массива массивов "по умолчанию" I

Массив массивов  
`int[][] a = new int[4][5]`



# Инициализация массива массивов "на месте" I

Массив массивов  
`int[] a = { {1, 2}, {6}, {4, 5, 7}, {12, 8, 1, 45} }`



# Объявление и инициализация массивов I

```
public class D_Arrays { /// Массивы. Arrays: foreach, многомерные
    public static void main(String[] args) {
        // Инициализация
        int[] a1 = new int[]{1, 2, 3, 10};
        // Сокращённая форма инициализации
        int a2[] = {1, 2, 3, 10};
        // Массив - можно сразу не инициализировать
        int[] a3;
        a3 = new int[100];
        // Пробегает по всем элементам массива
        for (int i = 0; i < a1.length; i++) System.out.println("a1[" + i + "] = " + a1[i]);
        // Если нам не важны индексы, мы можем использовать цикл "foreach"
        for (int i : a2) System.out.print(i + " ");
        System.out.println(); // Просто перевод строки
        // == Двухмерный массив ==
        int[][] array2D = {
            {1, 2, 3},
            {4, 5, 6},
            {7, 8, 9}
        };
        int x = array2D[0][1]; // Элемент массива с индексом 0 1
        array2D[1][1] = 231; // Новое значение
    }
}
```

# Работа с массивами I

- Цикл **for-each**
- Класс **Arrays**:
  - **Arrays.toString(...)**, **Arrays.deepToString(...)**
  - **Arrays.equals(...)**, **Arrays.deepEquals(...)**
  - **Arrays.copyOf(...)**, **Arrays.copyOfRange(...)**
  - **Arrays.binarySearch(...)**, **Arrays.sort(...)**
- **System.arraycopy(...)**



# Зачем нам заполнять массив случайными числами? I

Обсуждение: алгоритмы для массивов и тестирование

# Алгоритм программы генерации массива со случайными числами I

Обсуждение:

- Создаём генератор случайных чисел
- Создаём массив для случайных чисел
- Заполняем массив случайными числами в цикле
- Выводим массив на экран

# Реализация программы I

```
import java.util.Random;

public class GenRandom {
    public static void main(String[] args) {
        // Создаём генератор случайных чисел
        Random gen = new Random();
        // Создаём массив для случайных чисел
        int a[] = new int[20];
        a[0] = 10; // Записываю 10
        // Заполняем массив случайными числами в цикле
        for (int i = 0; i < a.length; i++) {
            a[i] = gen.nextInt(100);
        }
        // Выводим массив на экран
        for (int x : a) System.out.print(x + " ");
    }
}
```

```
javac -encoding utf8 GenRandom.java
java GenRandom > GenRandom.txt
```

18 3 95 10 83 74 70 58 29 81 9 50 38 69 19 10 89 79 78 55

# Реализация алгоритмов поиска I

```
import java.util.Random;

public class FindMinimum {
    public static void main(String[] args) {
        Random gen = new Random();
        int a[] = new int[8];
        for (int i = 0; i < a.length; i++) a[i] = gen.nextInt() % 100;
        // Выводим массив на экран
        for (int x : a) System.out.printf(x + " ");
        System.out.println();
        // Поиск минимума
        int min = a[0];
        for (int i = 1; i < a.length; i++)
            if (a[i] < min)
                min = a[i];
        // Выводим найденный минимум
        System.out.println("min = " + min);
    }
}
```

```
-72 -91 75 80 9 71 -8 -65
min = -91
```

# Поиск элемента в массиве I

```
public class FindValue {  
    public static void main(String[] args) {  
        int a[] = {10, 102, 1 - 2};  
        // Выводим массив на экран  
        for (int x : a) System.out.printf(x + " ");  
        System.out.println();  
        int value = 102; // Значение для поиска  
        // Поиск элемента  
        // Сложность:  $O(n)$  - линейная  
        // Количество операций пропорционально количеству элементов в массиве  
        for (int x : a)  
            if (x == value) {  
                System.out.println("Found " + value); // Нашли!  
                return;  
            }  
        System.out.println(value + " not found!"); // Не нашли  
    }  
}
```

```
10 102 -1  
Found 102
```

# Простейший алгоритм сортировки массива I

```
import java.util.Arrays;
import java.util.Random;

public class Sort { // Сортировка
    public static void main(String[] args) {
        // Создаем массив и заполняем случайными числами
        Random gen = new Random();
        int a[] = new int[8];
        for (int i = 0; i < a.length; i++) a[i] = gen.nextInt() % 100;
        // Выводим массив на экран
        for (int x : a) System.out.print(x + " ");
        System.out.println();
        // Сортировка "пузырьком"
        for (int i = 0; i < a.length; i++)
            for (int j = i + 1; j < a.length; j++)
                if (a[i] > a[j]) { // Если не на своих местах => меняем местами
                    int temp = a[i];
                    a[i] = a[j];
                    a[j] = temp;
                }
        // Выводим массив на экран
        for (int x : a) System.out.print(x + " ");
        System.out.println();
        // На практике самостоятельно не реализуют сортировку, а делают так:
```

# Простейший алгоритм сортировки массива II

```
        Arrays.sort(a);  
    }  
}
```

```
43 31 -42 49 -98 72 46 41  
-98 -42 31 41 43 46 49 72
```