

IT ШКОЛА SAMSUNG

Модуль 1. Основы программирования

Урок. Массивы

The Samsung logo, consisting of the word "SAMSUNG" in white capital letters inside a blue oval shape.

SAMSUNG

```
int [] a, b, c;
```



Допустимо также объявление массивов в стиле языка C/C++, а именно, с записью квадратных скобок **после** имени массива:

```
int a[];
```

В этом случае объявление одновременно нескольких массивов будет выглядеть более громоздко:

```
int a[], b[], c[];
```

Однако применение такого синтаксиса позволяет одновременно

объявить как массив, так и простую переменную:

```
int a[], i;
```

```
имя_массива = new тип_массива [размер_массива]
```

Например:

```
a = new int [10];
```

Объявление массива можно совместить с его созданием, используя следующий синтаксис:

```
тип_массива имя_массива[] = new тип_массива [размер_массива]
```

Например:

```
int a[] = new int [10];
```

```
тип_массива имя_массива[] = new тип_массива[] {список_выражений};
```

Например, если имеются ранее объявленные и инициализированные переменные x и y , то можно следующим образом создать и инициализировать массив a :

```
int a[] = new int [] {3, 11, x, 2*x, y - x};
```

```
тип_массива имя_массива[ ] = {список_выражений};
```

Например:

```
int a[] = {3, 11, x, 2*x, y - x};
```

После создания массива его размер сохраняется в свойстве **length**, которое рекомендуется использовать в различных алгоритмах обработки массивов. Обращение к этому свойству имеет синтаксис:

```
имя_массива.length
```

Например, для массива **a**, объявленного в предыдущем примере, оператор

```
System.out.println(a.length);
```

выведет на экран значение **5**.

Элементы массива нумеруются, начиная с нулевого значения, и номер элемента называется его индексом. Таким образом, первому элементу массива соответствует значение индекса **0**, второму – значение индекса **1**, элементу с порядковым номером **k** – значение индекса **k-1**. Для доступа к отдельным элементам массива используется следующий синтаксис:

```
имя_массива [выражение_целого_типа]
```



*В случае, если в ходе выполнения **Java** программа пытается выйти за границы массива (обратиться к элементу массива с индексом вне диапазона от **0** до размерности массива минус **1**), то возникнет ошибка - исключение [ArrayIndexOutOfBoundsException](#)*

Поэтому необходимо внимательно отслеживать возможный диапазон значений, которые может принять индекс массива.

*Как сделать программу более устойчивой и обрабатывать, возникающие по ходу выполнения программы исключения, более подробно будет рассмотрено в одной из тем Модуля **2**.*

```
int maxRange = 100;
for (int i = 0; i < a.length; i++) {
    a[i] = (int) (Math.random() * (maxRange + 1)); //Целое число 0..max
    System.out.print("\t" + a[i]);
}
```

В общем случае, чтобы получить число из диапазона `[minRange; maxRange]` можно использовать следующее выражение:

```
(int) (minRange + Math.random() * (maxRange - minRange + 1));
```

```
for (тип_массива имя_переменной: имя_массива) тело_цикла
```

```
int sum = 0;  
for (int x: a)  
    sum += x;
```



```
public class myArray {  
  
    public static void main(String[] args) {  
        //TODO Auto-generated method stub  
  
        System.out.println("Массив, заполненный случайными  
числами");  
        // Объявление массива, определение максимума  
  
        int [] a;  
        int maxRange = 100;  
        // Создание массива  
  
        a = new int [10];  
        // Заполнение массива случайными числами  
        for(int i = 0; i < a.length; i++) {  
            a[i] = (int) (Math.random() * (maxRange + 1));  
            System.out.print("\t" + a[i]);  
  
        }  
        System.out.println();  
        System.out.println("\nМассив, созданный списком  
инициализации");  
        int x = 5;  
        // Объявление и создание массива  
        int b[] = {3, 11, x, 2 * x, a[1] - a[2]};  
        for(int y: b)  
            System.out.print("\t" + y);  
        System.out.println();  
  
    }  
}
```


СПАСИБО

Массивы

