

# Onedimensional Arrays

Single variable

1

Array: Indexes

0

1

2

3

4

Values

1

3

8

23

99

# Контрольные вопросы

- Что такое цикл?
- Когда обычно применяют цикл `for`?
- Какие есть особенности у цикла `for`?
- Как работает `break` в циклах?
- Чем отличается использование `continue` в цикле `while` и `for`?
- Основное правило вложенных циклов

# Одна переменная – одно значение

Для хранения данных в памяти можно использовать переменные. Но каждая переменная способна одновременно хранить лишь один элемент информации. Чтобы сохранить второй элемент, необходимо создать ещё одну переменную... Но что же делать, если нужно будет хранить целое множество однотипных элементов?

# Использовать массивы!



Для наглядности представьте себе собранные в один ряд пустые коробки. В каждую коробочку можно положить что-нибудь однотипное, например, котов. Теперь, даже не зная их по именам, вы можете выполнить команду **Накормить кота из 3 коробки**. Сравните с командой **Накормить Рыжика**. Чувствуете разницу? Вам не обязательно знать котов по именам, но вы всё равно сможете справиться с заданием. Завтра в этих коробках могут оказаться другие коты, но это не составит для вас проблемы, главное знать номер коробки, который называется **индексом**.

# Понятие массива

- Массив – это множество однотипных переменных, последовательно расположенных в памяти (т.е. рядом, друг за другом)
- Значения переменных массива доступны по одному общему имени
- При этом каждая переменная в массиве является самостоятельной единицей под названием **элемент**
- Каждый элемент имеет свой порядковый номер - **индекс**. По индексу можно обращаться к конкретному элементу массива
- Индексы массива начинаются с нуля (индекс - смещение от начала).



# Синтаксис создания массива

```
тип[] имя = new тип[количество_элементов];
```

или

```
тип имя[] = new тип[количество_элементов];
```

Создание ссылочной переменной (без массива):

```
int cats[]; // первый вариант
```

```
int[] cats; // второй вариант
```

# Примеры кода

```
int[] cats;  
cats = new int[8];  
// все элементы массива будут равны 0
```

```
int[] cats = new int[8];
```

```
int count = 8; // count = sc.nextInt();  
int[] cats = new int[count];
```

```
String cats[] = new String[8];  
// все элементы массива будут равны null
```

# Явная инициализация массива

```
int[] cats = new int[] {5, 4, 7, 3};
```

```
int[] cats = {5, 4, 7}; // упрощённая форма записи
```

```
String catNames[] = new String[] {"Рыжик",  
"Барсик", "Том"};
```

```
int[] cats, dogs, chipmunks; // три ссылки!
```

```
int cats[], dogs[], chipmunks[], amount;
```



# Индексация элементов массива



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

Элементы располагаются по возрастанию адресов. Формула, по которой виртуальная машина определяет нужный адрес:

**базовый адрес массива + размер базового типа \* индекс**

Если указать неправильный индекс, программа попытается получить полный доступ к содержимому ячейки памяти, которая ей по сути не принадлежит. В результате этого произойдет ошибка на этапе выполнения.

# Обращение к элементам

cats[3] = 5; // четвертому по счёту элементу массива присвоено значение 5

cats[5] = 7; // шестому по счёту элементу массива присвоено значение 7

System.out.println(cats[3]); // отображение на экране консоли четвертого элемента

cats[6] = 7; // при выходе за границы массива **выбрасывается исключение**  
**ArrayIndexOutOfBoundsException**

# Количество элементов

// количество элементов массива может быть явно запрошено через свойство **length**.

```
int count = 10;
```

```
int[] ar = new int[count];
```

```
int size = ar.length; // 10
```

```
System.out.println("Размер массива: " + size);
```

# Вопрос

Сколько места занимает в памяти массив такого вида?

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

# Размер массива в байтах

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

Заголовок	8 байт + 4 байта = 12 байт
Примитивы типа int	4 байта x 10 шт. = 40 байт
Выравнивание для кратности 8	4 байта
Итого	56 байт

Структура заголовка объектов в Java:

<https://habrahabr.ru/post/134102/>

# Инициализация циклом

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

```
ar[0] = 10;
```

```
ar[1] = 20;
```

```
ar[2] = 30;
```

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

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

```
    ar[i] = (i + 1) * 10;
```

```
}
```

# Практика: gismeteo

Климат Одессы												
Показатель	Янв.	Фев.	Март	Апр.	Май	Июнь	Июль	Авг.	Сен.	Окт.	Нояб.	Дек.
Абсолютный максимум, °C	15,1	18,6	24,1	29,4	33,3	35,6	39,3	38	32,4	30,5	26	16,3
Средний максимум, °C	2,2	2,7	6,6	13	19,5	24	27	26,5	21	15	8,4	3,7
Средняя температура, °C	-0,5	-0,2	3,5	9,4	15,6	20	22,6	22,3	17,2	11,6	5,7	1,1
Средний минимум, °C	-2,8	-2,6	1	6,6	12,1	16,3	18,5	18,2	13,5	8,6	3,2	-1,2
Абсолютный минимум, °C	-26,2	-28	-16	-5,9	0,3	5,2	7,5	7,9	-0,8	-13,3	-14,6	-19,6
Норма осадков, мм	34	37	32	27	36	49	45	38	43	34	41	35

Упражнение: написать программу, которая рассчитает среднегодовую температуру (среднюю, максимальную и минимальную), по предоставленным данным. Между какими двумя месяцами года происходит максимальный перепад средних температур?

# Преобразование массива в строку

```
import java.util.Arrays;

public class Program {

    public static void main(String[] args) {
        int[] ar = {1, 3, 5, 7};
        String result = Arrays.toString(ar);
        System.out.println(result);
    }
}
```

Main

```
"C:\Program Files\Java\jdk1.8.0\bin\java" ...
[1, 3, 5, 7]
```



# Заполнение массива

```
public class Program {  
  
    public static void main(String[] args) {  
        int[] ar = new int[5];  
        Arrays.fill(ar, 27);  
        System.out.println(Arrays.toString(ar));  
    }  
}
```

Main

```
"C:\Program Files\Java\jdk1.8.0\bin\java" ...  
[27, 27, 27, 27, 27]
```

# Копирование данных

```
3 import java.util.Arrays;
4
5 /**
6  * @author Сама
7  */
8 public class JavaApplication13 {
9
10     public static void main(String[] args) {
11         int[] source = new int[]{11, 22, 33, 44, 55, 66, 77};
12         int[] destination = new int[5];
13         System.arraycopy(source, 2, destination, 0, 3);
14         System.out.println(Arrays.toString(destination));
15     }
```

javaapplication13.JavaApplication13 >

Документация Javadoc

Вывод - JavaApplication13 (run) X

run:

[33, 44, 55, 0, 0]

СБОРКА УСПЕШНО ЗАВЕРШЕНА (общее время: 0 секунд)

# Сортировка данных в массиве

```
public class Program {  
  
    public static void main(String[] args) {  
        int[] ar = {2, 46, 38, 1, 116, 14, 20};  
        Arrays.sort(ar);  
        System.out.println(Arrays.toString(ar));  
    }  
}
```

Main

```
"C:\Program Files\Java\jdk1.8.0\bin\java" ...  
[1, 2, 14, 20, 38, 46, 116]
```

# Сравнение массивов

```
public static void main(String[] args) {  
    int[] ar = {2, 46, 38, 1};  
    int[] br = {2, 47, 38, 1};  
    int[] cr = {2, 46, 38};  
    int[] dr = {46, 38, 1, 2};  
    int[] er = {2, 46, 38, 1};  
    System.out.println(Arrays.equals(ar, br));  
    System.out.println(Arrays.equals(ar, cr));  
    System.out.println(Arrays.equals(ar, dr));  
    System.out.println(Arrays.equals(ar, er));  
}
```

ain

false

false

false

true

# Пример кода

<https://git.io/vKUXM>

# Алгоритмы поиска

<https://git.io/vKUXQ>