



Java

Массивы

Лекция #6 А

Пустовалова О.Г.
доцент. каф. мат.мод.
ИММИКН ЮФУ

Содержание

- **Объявление и инициализация массива. Длина массива**
- **Одномерные массивы**
- **Двумерные массивы**
- **Неровные массивы**
- **Примеры**

Одномерные массивы

Массивы

Массив — это структура данных, в которой хранятся величины одинакового типа.

Доступ к отдельному элементу массива осуществляется по целочисленному индексу.

```
int[] a = new int[100];
```

Длина массива не обязательно должна быть постоянной.

Массивы

Объявить массив можно двумя способами:

```
int[] a; или int a[];
```

Создание массива:

```
int[] a = new int[100];
```

При создании массива чисел все его элементы **инициализируются нулями.**

Массивы значений типа `boolean` инициализируются логическим значением **false**.

А массивы **объектов** — **пустым значением null**, указывающим на то, что массив пока еще не содержит ни одного объекта.

Создание массива строк

Каждый элемент массива будет иметь значение `null`:

```
String[] names = new String[10];
```

Для того, чтобы создать массив из **пустых символьных строк**, его придется специально заполнить пустыми строками:

```
String[] names = new String[10]; // null  
for (int i = 0; i < 10; i++) names[i] = "";
```

Выход за пределы массива

Если, создав массив, состоящий из 10 элементов, вы попытаетесь обратиться к элементу а [10] (или любому другому элементу, индекс которого выходит за пределы от 0 до 9), выполнение программы прервется, поскольку будет сгенерировано исключение в связи с выходом индекса массива за допустимые пределы.

```
String[] names = new String[10]; // null
for (int i = 0; i < 10; i++) names[i] = "";

System.out.println(names[10]);
```

Array index is out of bounds [more...](#) (Ctrl+F1)

Количество элементов в массиве

```
for (int i = 0; i < a.length; i++)  
    System.out.println(a[i]);
```

После создания массива изменить его размер нельзя.

У списочного массива можно изменять длину.

Цикл for each для массивов

```
int[] a = new int[10]; // null
for (int i = 0; i < 10; i++) a[i] = i+1;

for (int element : a)
    System.out.print(element) ;
```

Переменная цикла в стиле **foreach** перебирает **элементы** массива, а не значения индекса.

Массив нулевого размера

При необходимости можно создать **массив нулевого размера**.

Такой массив может оказаться полезным при написании метода, возвращающего массив, который оказывается в некоторых случаях пустым.

Массив нулевой длины объявляется следующим образом:

```
int[] b = new int[0];  
System.out.println(b.length) ;
```

0

Инициализация массивов

```
int[] c = { 2, 3, 5, 7, 11, 13};
```

//в данном случае операция new не требуется

Пример. Одномерный массив

```
public class Main {  
    public static void main(String args[]) {  
  
        int month_days[];  
        month_days = new int[12];  
  
        month_days[0] = 31;  
        month_days[1] = 28;  
        month_days[2] = 31;  
        month_days[3] = 30;  
        month_days[4] = 31;  
        month_days[5] = 30;  
        month_days[6] = 31;  
        month_days[7] = 31;  
        month_days[8] = 30;  
        month_days[9] = 31;  
        month_days[10] = 30;  
        month_days[11] = 31;  
        System.out.println("April has " + month_days[3] + " days.");    }  
}
```

Пример. Одномерный массив

```
public class Main {
```

April has 30 days.

```
    public static void main(String args[]) {
```

```
        int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };
```

```
        System.out.println("April has " + month_days[3] + " days.");
```

```
    }
```

```
}
```

Многомерные массивы

Многомерные массивы

Для доступа к элементам многомерного массива применяется несколько индексов.

Такие массивы служат для хранения таблиц и более сложных упорядоченных структур данных.

Объявление двумерного массива:

```
double[][] a;
```

Выделение памяти и инициализация (нулями) массива:

```
a = new double[2][3];
```

Многомерные массивы

Инициализация без использования **new**.

```
int[][] magicSquare =  
    {  
        {16, 3, 2, 13},  
        {5, 10, 11, 8},  
        {9, 6, 7, 12},  
        {4, 15, 14, 1}  
    };
```

Обращение к элементам двумерного массива

```
System.out.println(magicSquare[0][0]) ;
```

Пример. Двумерный массив. Инициализация массива

```
public class Main {  
  
    public static void main(String args[]) {  
        double m[][] = {  
            {0 * 0, 1 * 0, 2 * 0, 3 * 0},  
            {0 * 1, 1 * 1, 2 * 1, 3 * 1},  
            {0 * 2, 1 * 2, 2 * 2, 3 * 2},  
            {0 * 3, 1 * 3, 2 * 3, 3 * 3}  
        };  
        int i, j;  
  
        for (i = 0; i < 4; i++) {  
            for (j = 0; j < 4; j++)  
                System.out.print(m[i][j] + " ");  
            System.out.println();  
        }  
    }  
}
```

```
0.0 1.0 2.0 3.0  
0.0 2.0 4.0 6.0  
0.0 3.0 6.0 9.0
```

Вывод элементов двумерного массива «на скорую руку»

```
int[][] magicSquare =  
{  
    {16, 3, 2, 13},  
    {5, 10, 11, 8},  
    {9, 6, 7, 12},  
    {4, 15, 14, 1}  
};
```

```
System.out.println(Arrays.deepToString(magicSquare));
```

```
[[16, 3, 2, 13], [5, 10, 11, 8], [9, 6, 7, 12], [4, 15, 14, 1]]
```

Метод **deepToString()** из **java.util.Arrays** позволяет выводить многомерные массивы

Вывод элементов двумерного массива с помощью for each

```
for (int[] row : magicSquare)

    for (int value : row)

        System.out.print(value+" ");
```

```
16 3 2 13 5 10 11 8 9 6 7 12 4 15 14 1
```

Цикл в аиле for each не обеспечивает автоматического перебора элементов двумерного массива.

Он лишь перебирает **строки**, которые, в свою очередь, являются одномерными массивами.

Пример. Двумерный массив

```
public class Main {  
  
    public static void main(String args[]) {  
        int twoD[][]= new int[4][5];  
        int i, j, k = 0;  
  
        for(i=0; i<4; i++)  
            for(j=0; j<5; j++) {  
                twoD[i][j] = k;  
                k++;  
            }  
  
        for(i=0; i<4; i++) {  
            for(j=0; j<5; j++)  
                System.out.print(twoD[i][j] + " ");  
            System.out.println();  
        }  
    }  
}
```

```
0 1 2 3 4  
5 6 7 8 9  
10 11 12 13 14  
15 16 17 18 19
```

Неровные массивы

Неровные массивы

В Java легко формируются **неровные**, "рваные", массивы, т.е. такие массивы, у которых **разные строки имеют разную длину**.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

1 6 15 20 15 6 1

Пример. Двумерный неровный массив. Вторая размерность

```
public class Main {  
    public static void main(String args[]) {  
  
        int twoD[][] = new int[4][];  
        twoD[0] = new int[1];  
        twoD[1] = new int[2];  
        twoD[2] = new int[3];  
        twoD[3] = new int[4];  
  
        int i, j, k = 0;  
        for(i=0; i<4; i++)  
            for(j=0; j<i+1; j++) {  
                twoD[i][j] = k;  
                k++;  
            }  
        for(i=0; i<4; i++) {  
            for(j=0; j<i+1; j++)  
                System.out.print(twoD[i][j] + " ");  
            System.out.println();  
        }  
    }  
}
```

```
0  
1 2  
3 4 5  
6 7 8 9
```



Спасибо за внимание!