
Мультимедийный курс

Программирование на Java

Часть 1

Лекция 3.2

МАССИВЫ В JAVA

1. Одномерные массивы
2. Многомерные массивы
3. Массивы-параметры
4. Анонимные массивы

Ссылочные типы :

массивы, строки, классы

При работе со ссылочными типами :

- ❖ операторы "=" и "==" присваивают и сравнивают **ссылки, а не значения**
- ❖ значение *простого типа не может быть преобразовано к ссылочному типу и, наоборот*

null - специальное значение (константа), представляющее **пустую ссылку**

Чтобы воспользоваться массивом, необходимо:

1) объявить переменную массива

```
<тип> <имя>[];
```

или

```
<тип>[] <имя>;
```

2) выделить память под массив

```
<имя> = new <тип> [<число_элементов>];
```

Можно совмещать объявление с выделением памяти:

```
<тип>[] <имя> = new <тип> [<число_элементов>];
```

Примеры объявления и размещения массивов



```
float[] f;
```

```
f = new float[300];
```

```
byte[] buffer = new byte[1024];
```

```
byte buffer[] = new byte[1024];
```

Инициализация - осуществляется после размещения массива в памяти

- по умолчанию
- при помощи **списка инициализации** (в этом случае операция *new* не используется)

```
float m1 []={23.2f, 3, 45};
```

Нумерация элементов массива
начинается с 0

Возможности

- Количество элементов массива **не может быть изменено**
- Переменной можно **присвоить ссылку на другой массив того же типа**

```
float m1[]={23.2f,3,45};
```

```
float mas_c[]={333,444,555,666};
```

```
...
```

```
m1=mas_c
```

Возможности

- **Размер массива** хранится в поле *length* класса массива, (только для чтения)

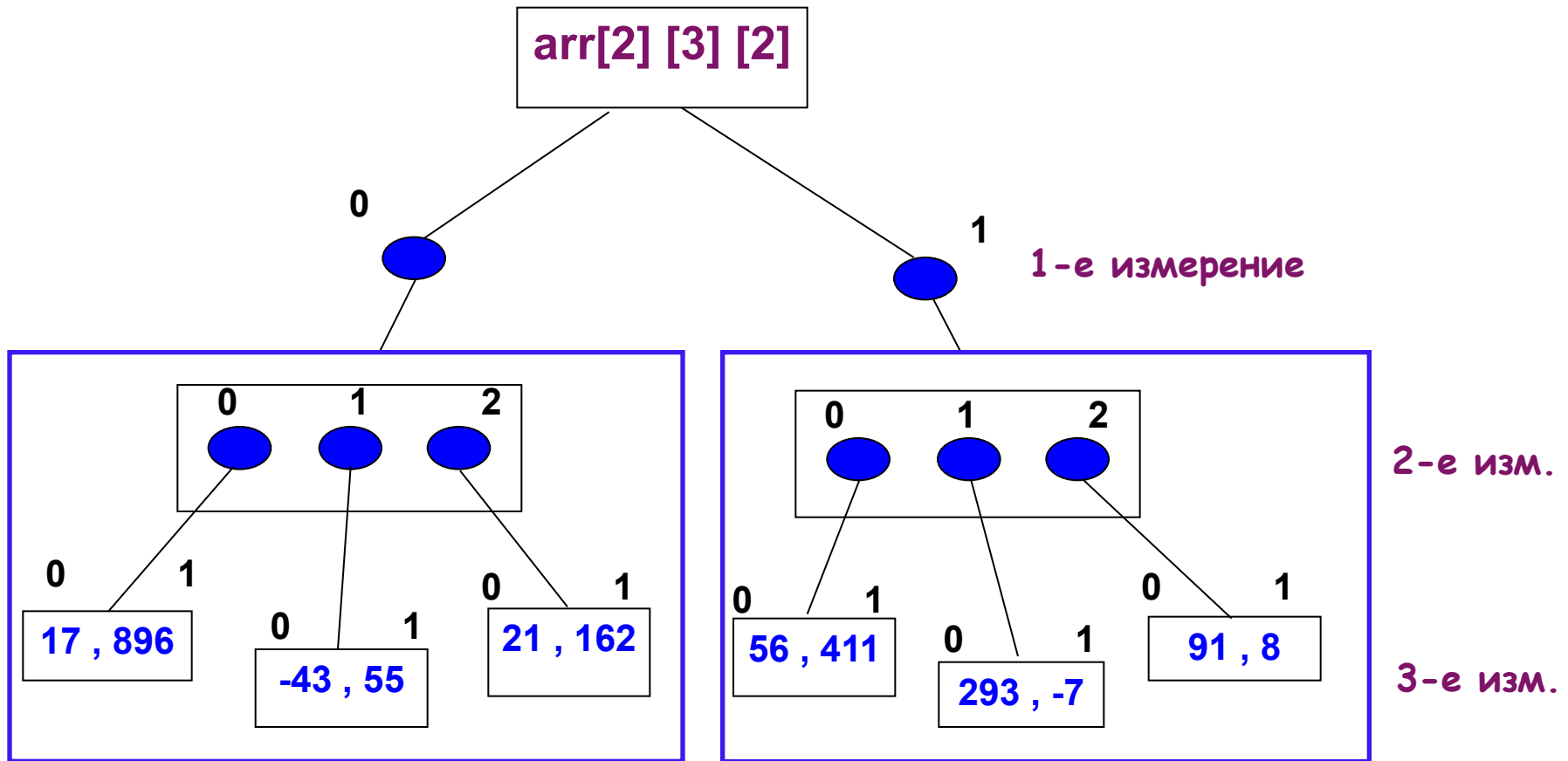
```
for( int i = 0; i < m.length; i++)  
m[i] = i;
```

- При нарушении границ массива формируется **исключение** `ArrayIndexOutOfBoundsException`

Многомерные массивы

Многомерный массив – это массив массивов.

Его элемент - ссылка на массив на размерность меньше



Объявление и выделение памяти

```
byte m2[ ][ ] = new byte [256][16];
```

- ✓ Поэтапное задание размерностей, начиная с левой

```
byte m3[ ][ ][ ] = new byte [2][ ][ ];
```

...

```
m3[0] = new byte [3][ ];
```

```
m3[1] = new byte [5][2];
```

- ✓ Для подмассивов одного уровня можно задавать разные размерности

```
m3[0][0] = new byte [1];
```

```
m3[0][1] = new byte [3];
```

Это позволяет организовывать "непрямоугольные"
массивы

Явная инициализация: при помощи
вложенных списков инициализации

```
byte m3[ ][ ][ ] = {{{17, -2}, {6, 9}, {63, 90}}, {{33, 46},  
{-24, 55}, {111, 123}}}; // размерность [2][3][2];
```

Объявление, размещение и инициализация
«треугольного» массива

```
int m[ ][ ] = {{1}, {2, 3}, {4, 5, 6}};
```

Массивы могут быть параметрами и возвращаемыми значениями методов

```
static int[ ] inversion (int[ ] a)  
{  
    int len = a.length;  
    int[] b = new int[ len ];  
    for (int i = 0; i < len; i++)  
// Инвертирование порядка элементов  
    b[i] = a[ len - i - 1];  
    return b;  
}
```

Анонимные массивы – массивы без имени

Аргументы при вызове методов:



```
int[ ] x = inversion(new int[ ] {1,2,3,4,5});
```