Назначение лекции

Вы познакомитесь с особенностями объявления и работы с массивами в Java.

Новые слова

**Прямоугольный массив** – двухмерный массив, у которого строки содержат одинаковое количество элементов.

**Непрямоугольный массив** – двухмерный массив, у которого строки содержат разное количество элементов.

# Напомним о структуре «Массив»

Массив – это структура хранения данных. Массив имеет имя. Массив состоит из элементов, обычно одного типа (например, целые числа, символы или строки). Каждому элементу соответствует индекс (целое, положительное число).

Указывая в программе имя массива и индекс элемента можно записывать или читать требуемый элемент.

В языке Java в одной программе не разрешается менять размер массива (количество элементов). Если возникает такая необходимость, то создается новый массив с другой размерностью, куда переписывеются элементы «старого» массива.

Массив хранится в оперативной памяти компьютера и при его выключении стирается. Для сохранения массива требуется запись на внешний носитель, например, флешку.

Пример массива: улица населенного пункта.

Чуй – имя массива

31, 10 – индексы (дом и квартира)

# Объявление массива

Объявим **одномерный** массив из целых чисел:

**int[] arr = new int [5];**

Читается эта запись следующим образом:

объявляется массив с именем **err** для целых чисел и в памяти выделяется место (new) для пяти таких элементов.

Объявление массива из трех строк:

**String[] name =new String [3];**

Объявим **двухмерный** массив из целых чисел:

**int [] [] arr = new int [5] [3];**

Здесь объявлен массив **arr** из 5 строк и в каждом по 3 элемента. Подобный массив называется **прямоугольный**, т.к. в каждой строке одинаковое количество элементов.

Аналогично объявляются трех и более-мерные массивы.

В Jave разрешены и **не** **прямоугольные** массивы. В этом случае строки могут иметь разное количество элементов. Например,

**int [] [] arr = new int [2] [];**

где, второй индекс не определен. Для их определения требуется дополнение:

**arr[0] = new int [2];**

**arr[1] = new int [3];**

где, первая строка будет состоять из двух элементов, а вторая – из трех.

Примечание. Индексы элементов, как и в С++, начинаются с нуля.

# Примеры работы с одномерным массивом

Пример 1. Массив из 5 целых чисел

**int [] arr = new int [5];**

**arr [2] = 10;**

**arr [0] = 5;**

**System.out.println (arr [0] + " " + arr [2]);**

Результат:

5 10

Пример 2. Выдача элементов массива в цикле

**int [] arr = new int [5];**

**arr [0] = 3;**

**arr [1] = 12;**

**arr [2] = 0;**

**arr [3] = 21;**

**arr [4] = 2;**

**for (int i=0;i<5;i++) System.out.println (arr [i]);**

Результат:

3

12

0

21

2

Пример 3. Выдача элементов массива строк в цикле

**String[] name =new String [3];  
name[0]="Касымов";  
name[1]="Азамат";  
name[2]="Омурбекович";  
for (int i=0;i<3;i++) System.*out*.println(name[i]);**

Результат:

Касымов

Азамат

Омурбекович

В том случае, если количество элементов массива небольшое, то можно инициализировать массив при объявлении:

**String[] name ={"Касымов","Азамат","Омурбекович"};  
for (int i=0;i<3;i++) System.*out*.println(name[i]);**

Пример 4. Выдача элементов двухмерного массива целых чисел в цикле.

Для выдачи двухмерного массива потребуется два цикла. Один вложен в другой. Например,

**int [] [] mas=new int [2] [3];  
  
mas[0][0]=3;  
mas[0][1]=5;  
mas[0][2]=0;  
  
mas[1][0]=21;  
mas[1][1]=7;  
mas[1][2]=1;  
  
for (int i=0;i<2;i++)  
 for (int j=0;j<3;j++) System.*out*.println(mas[i][j]);**

Результат:

3

5

0

21

7

1

Пример 5. Выдача элементов двухмерного не прямоугольного массива целых чисел в две строки.

**int [] [] mas=new int [2] [];  
mas[0]=new int [2];  
mas[1]=new int [3];  
  
mas[0][0]=3;  
mas[0][1]=5;  
  
mas[1][0]=21;  
mas[1][1]=7;  
mas[1][2]=1;  
  
for (int j=0;j<2;j++) System.*out*.print(mas[0][j]+" ");  
System.*out*.print("\n");  
for (int j=0;j<3;j++) System.*out*.print(mas[1][j]+" ");**

Результат:

3 5

21 7 1

Примечание. Необходимо обратить внимание на методы **println и print.** Первая выполняет переход на следующую строку, после каждого обращения, а вторая этого не делает

Конец лекции