гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Компютърно образователен център Progress Палитра от възможности!

Лекция 7.1

Използване на масиви в Java – 1-ва част

- **Въведение в масиви**
- **Приложение на масиви**
- **Деклариране на масив, инициализиране и реферирane на отделни стойности от масива**
- **Примери**
- **Задачи**

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

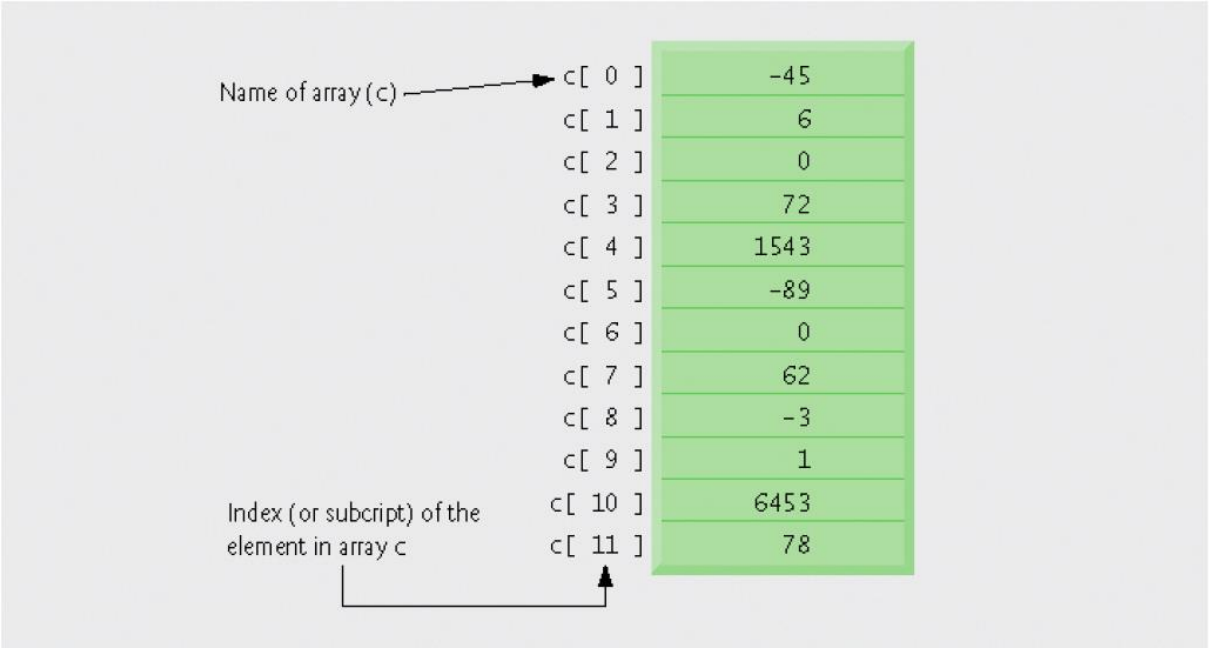
Въведение

Масиви

- Структури от данни т.е. съвкупност от логически свързани елементи от определен тип данни
- Масивите съдържат данни от един и същ тип
- Масивите са референтен тип данни
- Броят на елементите определя дължината на масива
  - Веднъж създаден, дължината на масива остава **неизменна**

Определение

- Масивът е подредена група от променливи(наричани елементи или компоненти) съдържащи стойности от един и същ тип данни
- Елементите могат да са от **примитивен** или от **референтен** тип
- - За достъп до определен елемент на масива и съответно до неговата стойност се използва позиция(наричана още *index*)
- Масивите са референтен тип данни – представят се от променлива, която реферира обект от тип масив в Heap паметта



Графика на масив от целочислен тип и дължина 12

- Фигурата изобразява логическо представяне на масив от цели числа(целочислен масив), реферирaн(именуван) с променливата **c**
- Този масив има 12 елемента
- Програмно, всеки от тези елементи се реферира като се използва името на масива, следвана от позицията(индекса) на този елемент в масива, заградена в квадратни скоби([])
- Първият елемент на всеки масив има индекс нула и се нарича нулев елемент
- Така, елементите на масива се означават като
  c[0] , c[1], c[2] и тн.
  Последният елемент в този масив има индекс 11, което е с 1 по-малко от дължината на масива с.

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Индекси

- Наричат се още subscript
- Задават позицията на даден елемент в квадратни скоби
- Винаги е положително число или резултат от пресмятане на целочислен израз
- Изменя се между 0 и **дължината на масива минус единица**

```
a = 5;
b = 6;
c[ a + b ] += 2;
```

Обичайни грешки при програмиране

Използване на променлива от тип long за индекс води до грешка при компилация.

Всеки индекс трябва да е от тип int или тип, който може да се сведе неявно до int, а именно byte, short или char, но не и long

Използване на отрицателни стойности за индекс или стойности по-големи или равни на дължината на масива води до грешка при изпълнение на програмата и нейното прекратяване.

Дължина на масив

За масива c :

- С променливата **c** е означено **името** на масива
- c.length връща дължината на масива **c**
- c има 12 елемента(c[0] , c[1] , … c[11])
- За пресмятане на сумта от първите три елемента на масива c и присвоим резултата в променливата sum, трябва да напишем на Java следната команда
  sum = c[0] + c[1] + c[2];

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Деклариране и създаване на масиви

- Масивите са обекти от референтен тип
  - Съхраняват се в Heap паметта, където заемат непрекъсната област
  - Създават се динамично с ключовата дума new
    - Пример за масив с елементи от примитивен тип

    ```
    int c [] = new int[ 12] ;
    ```
  - Еквивалентно на

    ```
    int c[] ;
    c = new int[ 12 ];
    ```

- Пример за масив с елементи от рефентен тип

  ```
  String b[] = new String [ 100 ] ;
  ```

Обичайна грешка при програмиране :
Задаването в квадратни скоби на броя на елементите на масив в неговата декларация(т.е., **int c[12];**) е синтактична грешка.

- При деклариране на масив може да се групира типа на данните с квадратните скоби с цел да се укаже,че всички променливи в тази декларация са променливи, реферирващи масиви от същия тип данни.
  Например декларацията

  ```
  double [] array1, array2;
  ```

  Указва, че array1 и array2 са „масив от double” променливи.
  Горната декларация е еквивалентна на:

  ```
  double array1[];
  double array2[];
  ```

  а също на

  ```
  double [] array1;
  double [] array2;
  ```

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net

Примери за използване на масиви

Основни съпки в използването на масиви

- Деклариране на масиви
- Създаване на масиви
- Инициализиране на елементите на даден масив
- Обработка на елементите на масив

Създаване и инициализиране на масив – пример

- Деклариране на масив
- Създаване на масив
- Инициализиране на елементите на масив

```
1  // Fig. 7.2: InitArray.java
2  // Creating an array.
3
4  public class InitArray
5  {
6     public static void main( String args[] )
7     {
8        int array[]; // declare array named array
9
10       array = new int[ 10 ]; // create the sp
11
12       System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings
13
14       // output each array element's value
15       for ( int counter = 0; counter < array.length; counter++ )
16          System.out.printf( "%5d%8d\n", counter, array[ counter ] );
17    } // end main
18 } // end class InitArray
```

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

Декларирane **array** като масив от **int** елементи

Създаване на **10** **int** елемента за **array**; всеки **int** се инициализира на **0** по подразбиране

array.length връща дължината на array

Всеки **int** елемент на масива е инициализиран на **0** по подразбиране

array[counter] връща **int** стойността на елемента на позиция **index** в масива **array**

Outline

InitArray.java

Ред 8
Декларира array като масив от **int** елементи

Ред 10
Създава 10 **int** елементи за array; всеки **int** елемент е инициализиран на 0 по подразбиране

Ред 15
array.length връща дължината на array

Ред 16
array[counter] връща **int** стойността на index място в масива array

- Използване на инициализиращ списък
  - Списък със стойности, разделени със запетая и оградени с фигурни скоби({ })
    ```
    int n[] = { 10, 20, 30, 40, 50};
    ```
  - Създава и инициализира масив от пет елемента
  - Индексът на масива може да взима стойности
    0, 1, 2, 3, 4
  - Няма нужда от ключовата дума **new**

```java
// Fig. 7.3: InitArray.java
// Initializing the elements of an array with an array initializer.

public class InitArray
{
   public static void main( String args[] )
   {
      // initializer list specifies the value for each element
      int array[] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };

      System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings

      // output each array element's value
      for ( int counter = 0; counter < array.length; counter++ )
         System.out.printf( "%5d%8d\n", counter, array[ counter ] );
   } // end main
} // end class InitArray
```

Декларира `array` като масив от **`int`** елементи

Компилаторът използва инициализиращ списък за да създаде **`array`**

```
Index   Value
    0      32
    1      27
    2      64
    3      18
    4      95
    5      14
    6      90
    7      70
    8      60
    9      37
```

Пресмятане на израз и съхраняване на резултата в променливите на масив

- Пример : Инициализираме елементите на масив с дължина с четни числа и табулираме стойностите на елементите на масива

```
1  // Fig. 7.4: InitArray.java
2  // Calculating values to be placed into elements of an array.
3
4  public class InitArray
5  {
6     public static void main( String args[] )
7     {
8        final int ARRAY_LENGTH = 10; // declare c
9        int array[] = new int[ ARRAY_LENGTH ]; //
10
11       // calculate value for each array element
12       for ( int counter = 0; counter < array.length; counter++ )
13          array[ counter ] = 2 + 2 * counter;
14
15       System.out.printf( "%s%8s\n", "Index", "Value" ); // column headings
16
17       // output each array element's value
18       for ( int counter = 0; counter < array.length; counter++ )
19          System.out.printf( "%5d%8d\n", counter, array[ counter ] );
20    } // end main
21 } // end class InitArray
```

```
Index   Value
    0       2
    1       4
    2       6
    3       8
    4      10
    5      12
    6      14
    7      16
    8      18
    9      20
```

Outline

Декларирането на константа ARRAY_LENGTH с използване на `final` атрибут

Деклариране и създаване на `array` с 10 `int` елемента

Използване на индекс в `array` за присвояване на стойност на елемент от array

InitArray.java

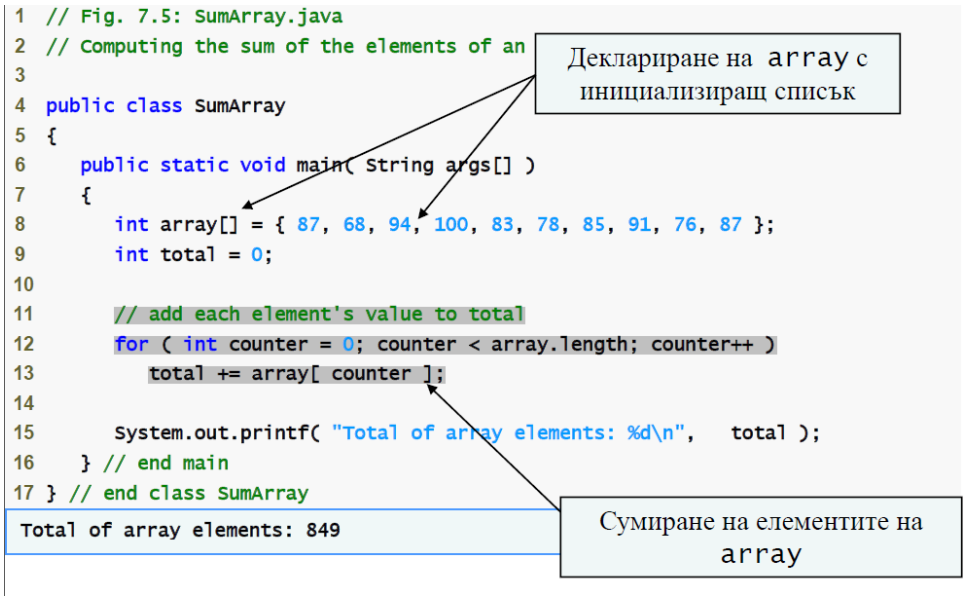
Ред 8
Деклариране на константа

Ред 9
Деклариране и създаване на array с 10 int елемента

Ред 13
Използване на индекс в array

Program output

- Сумиране на елементите на масив
  - Елементите на масив може да представляват един ред от стойности получени по определен начин
  - Често се налага сумирането на тези стойности



Outline

SumArray.java

Ред 8
Деклариране на array с инициализиращ списък

Редове 12-13
Сумиране на елементите на array

Program output

гр. София, пл. Славейков 11 ет.5 тел. : +359 897 91 93 96

office@progressbg.net www.progressbg.net