

Лекция 4

Масиви

Християн Димитров

Съдържание

- Въведение
- Създаване и инициализация
- Индексация
- Обхождане
- Предаване като параметър
- Многомерни масиви
- Списък аргументи с променлива дължина
- Command-Line аргументи

Създаване

- Масивите се създават с ключовата дума **new**

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

- Еквивалентно на:

```
int c[]; // declare array variable  
c = new int[ 12 ]; // create array
```

- Могат да се създават и масиви от обекти

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

Инициализация

- Инициализация по подразбиране

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

- Всички елементи на масива приемат стойност по подразбиране в зависимост от типа си – 0 за числени типове, false за типа boolean, null за референтни типове
- Инициализация със списък елементи

```
int n[] = { 10, 20, 30, 40, 50 };
```

- Списъкът се подава във фигурни скоби, разделен със запетаи
- Не се използва ключовата дума new

Индексация

- Индекс

`c[ a + b ]`

- Задава се в квадратни скоби след името на масива
 - Трябва да е положително цяло число или израз с такъв резултат
 - Индексирането започва от 0
 - `c.length` връща размера на масива
- Внимание!
 - Стойности от тип `long` не могат да се използват като индекси на масив – само `int`, `byte`, `short` или `char`
 - Опит за достъп извън границите на масива води до възникване на `ArrayIndexOutOfBoundsException`

Обхождане

- С брояч

```
for ( int i = 0; i < c.length; i++ ) {  
    // c[ i ];  
}
```

- С for за обхождане на масиви и колекции

```
for ( int number : c ) {  
    // number;  
}
```

- Директен достъп до елементите на масива
- Не може да се променят елементите
- Няма достъп до индекса на елемента

Пример

```
public class EnhancedForTest
{
    public static void main( String args[] )
    {
        int array[] = { 87, 68, 94, 100, 83, 78, 85, 91, 76, 87 };
        int total = 0;

        // add each element's value to total
        for ( int number : array )
        {
            total += number;
        }

        System.out.printf( "Total of array elements: %d\n", total );
    }
}
```

Предаване като параметър

- В списъка с аргументи се задава само името на масива без квадратните скоби

```
int hourlyTemperatures[] = new int[ 24 ];  
modifyArray( hourlyTemperatures );
```

- Забележка:
 - Тъй като масивите са референтен тип, методът не получава копие на масива, а референция към инстанцията, която сме подали като аргумент
 - Промените, извършени върху масива, се запазват и след излизане от обхвата на метода

Многомерни масиви

- Многомерните масиви са най-общо казано масиви от масиви
- Най-често използваните многомерни масиви са двумерните масиви, поради факта, че с тях лесно се представят таблици – редове и колони
- Забележка:
 - Не е задължително дължината на редовете в двумерните масиви да е една и съща

Многомерни масиви

- Създаване на многомерни масиви

- С фиксирани размери

```
int b[][] = new int[ 3 ][ 4 ];
```

- С редове с различна дължина

```
int b[][] = new int[ 2 ][ ]; // create 2 rows  
b[ 0 ] = new int[ 5 ]; // create 5 columns for row 0  
b[ 1 ] = new int[ 3 ]; // create 3 columns for row 1
```

- Инициализация със списък елементи

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

- Ред 0 има елементи 1 и 2, а ред 1 – 3, 4 и 5

Многомерни масиви

- Обхождане на многомерни масиви

- С брояч

```
for ( int row = 0; row < array.length; row++ )
{
    for ( int column = 0; column < array[ row ].length; column++ )
    {
        // array[ row ][ column ];
    }
}
```

- С for за обхождане на масиви и колекции

```
for ( int row[] : array )
{
    for ( int column : row )
    {
        // column;
    }
}
```

Списък аргументи с променлива дължина

- Задава се като многоточие (...) в списъка с аргументи
- Броят на аргументите е неизвестен
- Може да се задава само веднъж в списъка с аргументи на даден метод
- Трябва да се намира в края на списъка с аргументи
- Представява масив от елементи от един и същи тип

Пример

```
public class VarArgsTest
{
    // calculate average
    public static double average( double... numbers )
    {
        double total = 0.0; // initialize total

        // calculate total using the enhanced for statement
        for ( double d : numbers )
        {
            total += d;
        }

        return total / numbers.length;
    }
}
```

Пример

```
public static void main( String args[] )
{
    double d1 = 10.0;
    double d2 = 20.0;
    double d3 = 30.0;
    double d4 = 40.0;

    System.out.printf( "d1 = %.1f\nd2 = %.1f\nd3 = %.1f\nd4 = %.1f\n\n",
                       d1, d2, d3, d4 );

    System.out.printf( "Average of d1 and d2 is %.1f\n",
                       average( d1, d2 ) );
    System.out.printf( "Average of d1, d2 and d3 is %.1f\n",
                       average( d1, d2, d3 ) );
    System.out.printf( "Average of d1, d2, d3 and d4 is %.1f\n",
                       average( d1, d2, d3, d4 ) );
}
}
```

Command-Line аргументи

- Предават се след името на класа при изпълнение от командния ред

```
java MyClass a b
```

- Получават се като аргумент на `main` метода

```
String args[]
```

- Брой на аргументите

```
args.length
```

- Първи аргумент

```
args[ 0 ]
```

- За разлика от C++, където нулевият аргумент е името на изпълняваната програма



Въпроси?