CSC 3215: Object Oriented Programming - 1 (JAVA)

Arrays in Java

Dr. Kamruddin Nur

Associate Professor
Dept. of Computer Science, AIUB
kamruddin@aiub.edu

February, 2018

Contents

Arrays

In Java,

- An array is a **container object** that holds a fixed number of **values** of a **single data type**
- Array **length** is fixed
- Each item in an array is called an **element**
- Each element is accessed by its numerical **index**.

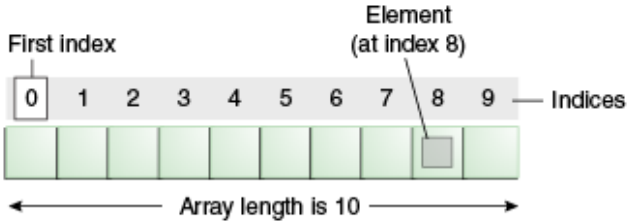

Figure 1: Arrays in Java, figure source: docs.oracle.com/javase/tutorial/

Array Declaration

Syntax:

- `dataType[] arrayName;` (preferred)
- `dataType arrayName[];` (more prevailing)

```
1
2  // array declaration for an array of int data type
3
4  int[] arrayOfIntegers;
5
6  //Similarly, for other data types
7  byte[] arrayOfBytes;
8  short[] arrayOfShorts;
9  long arrayOfLongs[];
10 float arrayOfFloats[];
11 double[] arrayOfDoubles;
12 boolean[] arrayOfBooleans;
13 char arrayOfChars[];
14 String arrayOfStrings[];
```

Array Instantiation (Creation)

```
1
2  // create an array for 10 integer data
3
4  arrayOfIntegers = new int[10];
5
6  //Similarly, for other data types
7  arrayOfBytes = new byte[10] ;
8  arrayOfShorts = new short[10];
9  arrayOfLongs = new long[10];
10 arrayOfFloats = new float[10];
11 arrayOfDoubles = new double[10];
12 arrayOfBooleans = new boolean[10];
13 arrayOfChars = new char[10];
14 arrayOfStrings = new String[10];
```

Array Declaration and Instantiation

```
1
2  // declare and instantiate(create) an array for 10 integer data
3
4  int arrayOfIntegers[] = new int[5];
5
6  //Similarly, for other data types
7  byte arrayOfBytes[] = new byte[10] ;
8  short arrayOfShorts[] = new short[10];
9  long arrayOfLongs[] = new long[10];
10 float arrayOfFloats[] = new float[10];
11 double arrayOfDoubles[] = new double[10];
12 boolean arrayOfBooleans[] = new boolean[10];
13 char arrayOfChars[] = new char[10];
14 String arrayOfStrings[] = new String[10];
```

Array Initialization

Syntax:

- `arrayName[index] = value;`

```
1
2  // declare and instantiate(create) an array for 10 integer data
3
4  arrayOfIntegers[0] = 100;
5  arrayOfIntegers[1] = 101;
6  arrayOfIntegers[2] = 110;
7  arrayOfIntegers[3] = 111;
8
9  //Similarly, for other data types
10 arrayOfBytes[0] = 127
11 arrayOfShorts[0] = 32767;
12 arrayOfLongs[0] = 1000000000001L;
13 arrayOfFloats[0] = 100.50f;
14 arrayOfDoubles[0] = 100.50d;
15 arrayOfBooleans[0] = false;
16 arrayOfChars[0] = 'a';
17 arrayOfStrings[0] = "A string array element.";
```

Declaration, Instantiation and Initialization

```
1
2  // declare, instantiate and initilize an array for integer data
3
4  int intArray[] = {100, 101, 110, 111, 200, 300, 500};
5
6  char charArray[] = { 'H', 'e', 'l', 'l', 'o',',',' ',
7                       'W','o', 'r', 'l', 'd', '!'};
```

2 Array Manipulation
- Printing
- Insertion
- Deletion
- arraycopy()

Printing a Java Array

- Using loop - for example, using **for**:

```
1
2   //Declaring and instantiating/creating an
        array
3  int marks[] = new int[100];
4
5  //initializing/inserting/populating marks
6  marks[0] = 75;
7  marks[1] = 85;
8  marks[2] = 60;
9  marks[3] = 55;
10 marks[99] = 70;
11
12 //printing all array elements
13 System.out.println("Marks:");
14 for(int i = 0; i<marks.length;i++){
15   if(marks[i] > 0){
16    System.out.println(marks[i]);
17   }
18 }
```

Printing a Java Array

- Using loop - for example, using **for**:

```
1
2   //Declaring and instantiating/creating an
        array
3  int marks[] = new int[100];
4
5  //initializing/inserting/populating marks
6  marks[0] = 75;
7  marks[1] = 85;
8  marks[2] = 60;
9  marks[3] = 55;
10 marks[99] = 70;
11
12 //printing all array elements
13 System.out.println("Marks:");
14 for(int i = 0; i<marks.length;i++){
15   if(marks[i] > 0){
16    System.out.println(marks[i]);
17   }
18 }
```

Program Output:

```
Marks:
75
85
60
55
70
```

Printing a Java Array *(cont'd)*

- Using loop - for example, using **`for each`**
- Alternative syntax of for loop (enhanced form of for loop) to iterate through items of arrays/collections
- Good when iterating through all items

```
1  char[] vowels = {'a', 'e', 'i', 'o', 'u'};
2  // foreach loop
3  for (char item: vowels) {
4  System.out.println(item);
5  }
```

Printing a Java Array *(cont'd)*

- Using loop - for example, using **`for each`**
- Alternative syntax of for loop (enhanced form of for loop) to iterate through items of arrays/collections
- Good when iterating through all items

```
1  char[] vowels = {'a', 'e', 'i', 'o', 'u'};
2  // foreach loop
3  for (char item: vowels) {
4  System.out.println(item);
5  }
```

Program Output:

```
a
e
i
o
u
```

Insertion

- Inserting numbers using *java.util.Random*;

```
1  int size = 5;
2  int randInt[] = new int[size];
3  float randFloat[] = new float[size];
4  double randDouble[] = new double[size];
5  long randLong[] = new long[size];
6
7  Random rand = new Random();
8
9  //inserting random value
10 for (int j=0; j < size; j++){
11  randInt[j] = rand.nextInt(50);
12  randFloat[j] = rand.nextFloat();
13  randDouble[j] = rand.nextDouble();
14  randLong[j] = rand.nextLong();
15 }
16 //printing these random number arrays
17 System.out.println("Inserted Random Numbers:");
18 System.out.println("Int \tFloat \t\tDouble \t\t\tLong");
19 for (int k=0; k < size;k++){
20  System.out.println(randInt[k]+"\t"+randFloat[k]+"\t"+randDouble[k]+
21                     "\t"+randLong[k]);
22 }
```

Insertion *(Cont'd)*

Program Output:

```
Inserted Random Numbers:
Int     Float           Double                  Long
6       0.33926302      0.9062140275485454      6158664939164313664
48      0.102960765     0.05608131029463659     7327332171887205453
45      0.47820288      0.036026334496322754    -482456665847366683
22      0.18870878      0.8443508782378616      3930692005384318630
0       0.5421631       0.9409076018027932      3760271761023181683
```

Deletion

- Simplest way, set the *defaultvalue* or *null* !

```
1  public void deleteBook(Book book){
2  for(int i = 0; i < listOfBook.length; i++){
3  if(listOfBook[i] == book){
4  listOfBook[i] = null;
5  System.out.println("Message: Book deleted.");
6  break;
7  }
8  }
9
10 }//end of deleteBook
```

Calling deleteBook() method and showing info:

```
lib.deleteBook(b1);
lib.deleteBook(b3);
lib.showLibInfo();
```

Deletion *Cont'd*

- Using *java.util.ArrayList*

```
1  /*Requires
2  import java.util.List;
3  import java.util.ArrayList;
4  */
5  List al = new ArrayList();
6  al.add(10);
7  al.add(20);
8  al.add(1);
9  al.add(2);
10
11 al.remove(new Integer(1));
12 al.remove(new Integer(2));
13
14 System.out.println("Modified ArrayList : " + al);
```

Program Output:

```
Modified ArrayList : [10, 20]
```

Deletion *Cont'd*

- Using *java.util.Iterator*, recommended when iterating over elements

```
1  import java.util.List;
2  import java.util.ArrayList;
3  import java.util.Iterator;*/
4  List al = new ArrayList();
5  al.add(10);
6  al.add(20);
7  al.add(1);
8  al.add(2);
9  // Remove elements smaller than 10 using
10 // instead of Iterator.remove()
11 Iterator itr = al.iterator();
12 while (itr.hasNext()){
13 int x = (Integer)itr.next();
14 if (x < 20){ itr.remove(); }
15 }
16
17 System.out.println("Modified ArrayList : "+ al);
```

Program Output:

```
Modified ArrayList : [20]
```

Copying a Java Array

- Using `arraycopy` method of System class
- `arraycopy(Object src, int srcPos,Object dest, int destPos, int length)`

```
1
2  char charArray[] = { 'H', 'e', 'l', 'l', 'o',',',' ',
3                       'W','o', 'r', 'l', 'd', '!'};
4  char copyTo[] = new char[7];
5
6  //arraycopy(Object src, int srcPos,Object dest, int destPos, int
       length)
7  System.arraycopy(charArray, 7, copyTo, 0, 5);
8
9  System.out.println(new String(copyTo));
```

Program Output:

```
World!
```

3 Multidimensional Array
Declaration, Instantiation, and Initialization

Multidimensional Array

```
1  // Declaration and Instantiation
2  int[][] twoDimArray = new int[3][4]; //3 rows, 4 cols
3  int twoDimArray2[][] = new int[3][4];
4  int[] matrix[] = new int[4][2]; //4 rows, 3 cols
5
6  //Declaration, Instantiation, and Initialization
7  int[][] twoDimArray4 ={ {1,2}, {1,2}, {1,2}, {1,2}, {1,2} };
8  double m[][] = {
9  { 1, 2, 3, 4 },
10 { 5, 6, 7, 8 },
11 { 9, 10, 11, 12 },
12 { 13, 14, 15, 16 }
13 };
14 //Initialization
15 matrix[0][0] = 1;
16 matrix[0][1] = 2;
17 matrix[1][0] = 1;
18 matrix[1][1] = 2;
19 matrix[2][0] = 1;
20 matrix[2][1] = 2;
21 matrix[3][0] = 1;
22 matrix[3][1] = 2;
```

Multidimensional Array *(Cont'd)*

- length

```
1
2  int[][] twoDArray = { {1,2}, {1,2}, {1,2}, {1,2}, {1,2} };
3  System.out.println("twoDArray Length: "+ twoDArray.length);
4                  //returns number of rows
5  System.out.println("twoDArray[0] Length: "+ twoDArray[0].length);
6  System.out.println("twoDArray[1] Length: "+ twoDArray[1].length);
7                  //returns number of cols
```

Thanks

Thanks for your time and attention!

kamruddin@aiub.edu