

# LABORATORY ASSIGNMENT № 7

Dr. Duncan, CSC 1350, Louisiana State University

11/05/2019

## Some Array Utility Methods

### Learning Objectives

1. Allocating and Using One-Dimensional Arrays,
2. Implementing Static Methods with Arrays as Parameters, and
3. Implementing Array-returning Static Methods

In chapter 7 of the textbook, we discussed algorithms that may be used to copy an array and generate the string representation of the contents of an array using a separator. We discussed the *copyOf*, *arraycopy* and *toString* static methods, all of which are implemented in the standard Java API (Application Programmer's Interface). In today's lab session, you will implement your own version of each method in a class called *ArrayUtil*. You will then write a program *ArrayUtilTester*, consisting of only the *main* method, to test the methods in the *ArrayUtil* class. To configure the Netbeans project correctly, you must add *ArrayUtilTester* to the project before *ArrayUtil*. See the specifications on these methods below.

In Java *null* is a special literal. It can be assigned to any reference variable but not a primitive type (eg: *int*, *char*, *double*, *boolean*, etc). Reference variables are those associated with objects, including arrays. To test whether an object reference is *null*, use the Boolean expression **objRef == null**, where *objRef* is the name of the reference variable. You will need to use this while implementing the methods below.

### The ArrayUtil Class

The *ArrayUtil* class will consist of the following methods:

Listing 1: The *toString* Method

```
/**
 * Returns a string representation of the contents of the specified array.
 * The string representation consists of a list of the array's elements,
 * enclosed in square brackets ("[]"). Adjacent elements are separated by
 * the characters ", " (a comma followed by a space). Elements are converted
 * to strings. Returns "null" if array is null.
 * @param array the array whose string representation to return
 * @return a string representation of array
 */
public static String toString(int[] array)
```

## Listing 2: The copyOf Method

```
/**
 * Copies the specified array, truncating or padding with zeros
 * (if necessary) so the copy has the specified length.
 * @param original the array to be copied
 * @param newLength the length of the copy to be returned
 * @return a copy of the original array, truncated or padded with
 * zeros to obtain the specified length
 * @throws IllegalArgumentException when:
 * 1. newLength is negative
 * 2. if original is null
 */
public static int[] copyOf(int[] original, int newLength)
```

## Listing 3: The arraycopy Method

```
/**
 * Copies an array from the specified source array, beginning at
 * the specified position, to the specified position of the
 * destination array.
 * @param src the source array.
 * @param srcPos starting position in the source array.
 * @param dest the destination array.
 * @param destPos starting position in the destination data.
 * @param length the number of array elements to be copied.
 * @throws IllegalArgumentException when
 * 1. if either src or dest is null.
 * 2. copying would cause access of data outside array bounds when
 * any of the following is true:
 * length < 0, srcPos < 0, destPos < 0, srcPos+length > src.length,
 * destPos+length > dest.length
 */
public static void arraycopy(int[] src, int srcPos, int[] dest, int destPos, int length)
```

## The ArrayUtilTester Program

Create a project *ArrayUtilTester* containing the *ArrayUtilTester* class. Neither the *ArrayUtil* nor the *ArrayUtilTester* class should use standard Java API methods. The main method should perform the following tasks:

- **Initial Version:** The *ArrayUtilTester* class will contain only the main method. Add the *ArrayUtil* class to the project. Then implement the *toString* and *copyOf* methods in the *ArrayUtil* class as described above. Add code to the main method that prompts the user for the size of an integer array, *first*. The main method then allocates *first*. It prompts the user for the entries of *first* and then stores the entries in *first*. It then calls the *copyOf* method in *ArrayUtil* to create an array, *second*, that is twice the size of *first* and another array *third* that is half the size of *first*. It uses *first* as an argument to *copyOf* when creating *second* and *third*. It then uses the *toString* method to display the arrays *first*, *second* and *third* as shown in the sample runs.

- **Final Version:** Add the implementation of the *arraycopy* method as described above to the *ArrayUtil* class. Then add code in the main method to do the following. Create an array *fourth* that is twice the size of *first*. The main method then prompts the user for the index in *fourth* at which to copy the contents of *first* into *fourth*. Using the *arraycopy* method, the main method copies the contents of *first* into *fourth* beginning at that index. The main then creates another array, *fifth*, that is twice the size of *first*. Using *arraycopy*, it copies the contents of *first* twice into *fifth* so that the copies of *first* are side-by-side in *fifth*. It then displays *first*, *fourth* and *fifth* as shown in the sample runs.

Note: If the length of *first* is an odd number, due to integer division half its length will be half of 1 less than its length. For example, if *first* has a length of 7, an array that is half its length will have a length of  $7/2 = 3$ .

## Other Requirements

Remove all Netbeans auto-generated comments. Include the Javadoc documentation for each method in *ArrayUtil*, as shown on this handout. Write header comments for both *ArrayUtil* and *ArrayUtilTester* using this Javadoc documentation template:

```
/**
 * Explain the purpose of this class; what it does <br>
 * CSC 1350 Lab # 7
 * @author YOUR NAME
 * @since DATE THE CLASS WAS WRITTEN
 * @see ArrayUtil (ADD THIS TAG ONLY IN THE ArrayUtilTester Javadoc HEADER COMMENTS)
 */
```

Here are sample program interactions:

Listing 4: Sample Run

```
Enter the size of the first array -> 7
Enter the entries of the array separated by spaces:
2 3 5 7 11 13 17

first: [2, 3, 5, 7, 11, 13, 17]
second: [2, 3, 5, 7, 11, 13, 17, 0, 0, 0, 0, 0, 0, 0]
third: [2, 3, 5]

Enter an index of 'fourth' at which to copy 'first' -> 4

first: [2, 3, 5, 7, 11, 13, 17]
fourth: [0, 0, 0, 0, 2, 3, 5, 7, 11, 13, 17, 0, 0, 0]
fifth: [2, 3, 5, 7, 11, 13, 17, 2, 3, 5, 7, 11, 13, 17]
```

#### Listing 5: Sample Run

```
Enter the size of the first array -> 5
Enter the entries of the array separated by spaces:
1 3 5 7 9

first: [1, 3, 5, 7, 9]
second: [1, 3, 5, 7, 9, 0, 0, 0, 0, 0]
third: [1, 3]

Enter an index of 'fourth' at which to copy 'first' -> 8

Exception in thread "main" java.lang.IllegalArgumentException: Invalid ↵
    argument
    at ArrayUtil.arraycopy(ArrayUtil.java:75)
    at ArrayUtilTester.main(ArrayUtilTester.java:35)
```

#### Listing 6: Sample Run

```
Enter the size of the first array -> 6
Enter the entries of the array separated by spaces:
1 1 2 3 5 8

first: [1, 1, 2, 3, 5, 8]
second: [1, 1, 2, 3, 5, 8, 0, 0, 0, 0, 0, 0]
third: [1, 1, 2]

Enter an index of 'fourth' at which to copy 'first' -> 3

first: [1, 1, 2, 3, 5, 8]
fourth: [0, 0, 0, 1, 1, 2, 3, 5, 8, 0, 0, 0]
fifth: [1, 1, 2, 3, 5, 8, 1, 1, 2, 3, 5, 8]
```

### Submitting Your Work for Grading

Navigate your way to the ...\\NetBeansProjects\\ArrayUtilTester\\src\\arrayutiltester folder using Windows file explorer. You should find *ArrayUtil.java* and *ArrayUtilTester.java*, files containing your source code for the program. Click one of the files and then hold down the shift key and click the other so that both files are selected. Right-click the selected files and create a compressed (zipped) folder containing a copy of each file. Rename the zip file *PAWSID\_lab07.zip*, where *PAWSID* is the prefix of your LSU/Tiger email address - the characters left of the @ sign. Double-click the zip file to verify that both *ArrayUtil.java* and *ArrayUtilTester.java* are included the zip file. If the zip file does not contain both files, delete the zip file and repeat the steps. Upload the zip file to the digital drop box on Moodle.