Software and Programming II (SP2) — Lab sheet 3

Arrays and ArrayLists

2024/25

Based upon exercises from Java for Everyone, 2e, Chapters 5 and 6.
Most of these you have probably encountered during Software and Programming I.

---

1. Perform each of the following tasks:
   (a) Create an array x of doubles with an initialiser list (see also the code for Question 2) that contains the following values:
   8, 4, 5, 21, 7, 9, 18, 2, and 100.
   (b) Print the number of items in the array by using an expression of the form `x.length`.
   (c) Print the first array item, `x[0]`.
   (d) Print the last array item. Be careful to choose the right index.
   (e) Print the expression `x[x.length - 1]`.
   (f) Use a standard `for` loop to print all the values in the array without labels.
   (g) Use a standard `for` loop to print all the values in the array with labels ("`x[0]`", "`x[1]`", . . .) to indicate what each element is.
   (h) Use a standard `for` loop to print all the values in the array in reverse order with labels to indicate what each element is.
   (i) Use an *enhanced* `for` loop to print all the values in the array without labels.

2. Write a method that is passed an array, x, of doubles and an integer rotation amount, n. The method creates a new array with the items of x moved forward by n positions. Elements that are rotated off the array will appear at the end. For example, suppose x contains the following items in sequence:

   ```
   1 2 3 4 5 6 7
   ```

   After rotating by 3, the elements in the new array will appear in this sequence:

   ```
   4 5 6 7 1 2 3
   ```

   Array x should be left unchanged by this method. Use the following code to help you get started. Be sure to test your program with different number of rotations.

```
public class Arrays2 {
    public static void main(String[] args) {
        double[] x = {8, 4, 5, 21, 7, 9, 18, 2, 100};
        System.out.println("Before rotation:  ==============================");
        for (int i = 0; i < x.length; i++) {
            System.out.println("x[" + i + "]:  " + x[i]);
        }
       // x = rotate(x, 3);
        System.out.println("After rotation:  ==============================");
        for (int i = 0; i < x.length; i++) {
            System.out.println("x[" + i + "]:  " + x[i]);
        }
    }

    // Your method goes here.
}
```

3. Create a class `CustomerLister` with a `main` method that instantiates an array of `String` objects called `customerName`. The array should have room for five `String` objects. Use an *initialiser list* to put the following names into the array:

   ```
   Cathy
   Ben
   Jorge
   Wanda
   Freddie
   ```

   Print the array of names.

4. Array lists are objects that, like arrays, provide you the ability to store items sequentially and recall them by index. Working with array lists involves invoking `ArrayList` methods, so we will need to develop some basic skills.

   The API documentation for the `ArrayList` class is available at
`https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/ArrayList.html`

   Now let's start with the code below:

   ```
   import java.util.ArrayList;

   public class ArrayListRunner {
      public static void main(String[] args) {
          ArrayList<String> names = new ArrayList<>();
          System.out.println(names);
      }
   }
   ```

   The main method imports `java.util.ArrayList` and creates an `ArrayList` that can hold strings. It also prints out the `ArrayList` and, when it does, we see that the list is empty: `[]`.

   Complete the following tasks by adding code to this skeleton program. If you are asked to print a value, provide a suitable label to identify it when it is printed.

(a) Invoke add() to enter the following names in sequence:
Alice, Bob, Connie, David, Edward, Fran, Gomez, Harry.
Print the ArrayList again.

(b) Use get() to retrieve and print the first and last names.

(c) Print the size() of the ArrayList.

(d) Use size() to help you print the last name in the list.

(e) Use set() to change Alice to Anna.
Print the ArrayList to verify the change.

(f) Use the alternate form of add() to insert Doug after David.
Print the ArrayList again.

(g) Use an *enhanced* for loop to print each name in the ArrayList.

(h) Create a second ArrayList called names2 that is built by calling the ArrayList constructor that accepts another ArrayList as an argument. Pass names to the constructor to build names2. Then print the ArrayList.

(i) Call names.remove(0) to remove the first element. Print names and names2. Verify that Anna was removed from names, but not from names2.

5. Write a program to

- read numbers from the user,
- put each number in an ArrayList, and
- print various information about the entered numbers.

Specifically, keep reading and adding numbers to your list until they enter -1 (that is, a loop that keeps getting numbers until you read -1).

Then output

(a) number of items entered before -1,

(b) the average of the input numbers,

(c) the standard deviation (see https://en.wikipedia.org/wiki/Standard_deviation#Discrete_random_variable) of the even numbers, and

(d) the sum of the odd numbers.

For example, this should be what happens when you run your program and enter:

```
Enter a number: 1
Enter a number: 2
Enter a number: 75
Enter a number: 26
Enter a number: -1

# of items: 4
Average: 26
Standard deviation of even numbers: 12.00
Sum of odd numbers: 76
```