

Welcome to ArrayList and Objects Lab

Learning Outcomes

By the end of this lab:

- Write, extend and debug code that uses ArrayLists.



Exercise A: Debug NumberList

Fix the bugs in [NumberList.java](#), noting specific changes you make.

- Review the actual output, including error messages, and make an hypothesis about where you think a problem may be.
- Insert a print statement, or view in the debugger, to verify whether a specific line of code is reached and confirm the hypothesized contents of variables at that line.
- Based on the result, make a new hypothesis and repeat.

Actual Output

```
initial:
[]
after removing contiguous duplicates:
[]
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index: 9, Size: 0
    at java.util.ArrayList.rangeCheck(Unknown Source)
    at java.util.ArrayList.get(Unknown Source)
    at NumberList.main(NumberList.java:49)
```

Expected Output

```
initial:
[6, 9, 8, 4, 5, 5, 5, 7, 9, 9]
after removing contiguous duplicates:
[6, 9, 8, 4, 5, 7, 9]
last element: 9
```

Review Questions

1. Describe the clues provided by the actual output and the exception message shown in the example.
java.lang.IndexOutOfBoundsException
2. Describe the process you went through to narrow down where the bugs were and what you did to fix them.
 - a. Describe how to fix the bug to produce the initial list.
For the for loop `i < size` and not `i < list.size()`.
 - b. Describe how to fix the bug that doesn't remove all the duplicates.

After the duplicate is removed make sure that we decrement `i` by 1 because all the elements in the array will be shifted to the left.

```
import java.util.ArrayList;
import java.util.Random;

public class NumberList {

    /**
     * Creates an ArrayList of size, populated with numbers
     * from 1 to 10 generated using rnd.nextInt().
     *
     * @param rnd A provided random number generator.
     * @param size The size of the list to create.
     * @return The list of numbers.
     */
    public static ArrayList<Integer> createList(Random rnd, int size) {
        ArrayList<Integer> list = new ArrayList<>(size);
        for (int i = 0; i < size; ++i) {
            list.add( rnd.nextInt(10) + 1 );
        }
        return list;
    }

    /**
     * The resulting list should not have the same number
     * twice in a row. Checks and removes duplicate values.
     *
     * Example:
     * [3,4,3,3] should result in [3,4,3]
     *
     * @param list A list of numbers
     */
    public static void remove2InARow(ArrayList<Integer> list) {
        for (int i = 0; i < list.size()-1; ++i) {
            if ( list.get(i) == list.get(i+1) ) {
                list.remove(i);
                i=i-1;
            }
        }
    }

    public static void main(String[] args) {
        final int COUNT = 10;
        Random rand = new Random(1); //don't change seed

        ArrayList<Integer> list = createList( rand, COUNT );
        System.out.println("initial:\n" + list);

        remove2InARow(list);
        System.out.println("after removing contiguous duplicates:\n" + list);
    }
}
```

```
System.out.println("last element: " + list.get(list.size() - 1));  
  
}
```



Exercise B: Hashtag Count Chart

Complete [Counter.java](#). The program asks a user for a hashtag and then a list of counts for that hashtag each day. The program then shows some summary statistics and a bar chart.

Expected Output

```
Enter a hashtag: happy
Enter count of #happy tweets per day (negative to end) .
200 213 43 25 19 100 -3
For 6 days minimum=19, maximum=213, average=100.000000
*****
***** (200)
***** (213)
***** (43)
***** (25)
***** (19)
***** (100)
A * represents 4 tweets.
```

PROGRAM CODE:

```
import java.util.ArrayList;
import java.util.Scanner;

public class Counter {

    public static void main(String[] args) {
        ArrayList<Integer> dailyCounts = new ArrayList<>();
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a hashtag: ");
        String hashtag = input.nextLine().trim();
        hashtag = hashtag.startsWith("#") ? hashtag : "#" + hashtag;

        System.out.println("Enter count of " + hashtag + " tweets per day (negative
to end).");
        ArrayList<Integer> days = new ArrayList<>(7);

        for (int i = 0; i < 7; i++) {
```

```

        days.add(input.nextInt());
    }

    //determine total number of days of data
    //determine minimum
    int minimum = days.get(0);
    int sum = 0;
    for (int i = 0; i < 6; i++) {
        sum += days.get(i);
        if (days.get(i) < minimum) {
            minimum = days.get(i);
        }
    }

    //determine maximum
    int maximum = days.get(0);
    for (int i = 0; i < 6; i++) {
        if (days.get(i) > maximum) {
            maximum = days.get(i);
        }
    }

    //determine average
    double average = (double) sum / 6;

    //draw a bar chart where maximum / 50 is represented by one *.
    int onestar = maximum / 50;
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < days.get(i) / onestar; j++) {
            System.out.print("*");
        }
        System.out.println("(" + days.get(i) + ")");
    }
    System.out.println("A * represents " + onestar + " tweets.");
}
}

```



Exercise C: Count per Hashtag Chart

Ask the user for hashtags. Count the number of each using parallel ArrayLists. Then show some summary statistics and a bar chart. Some starting code: [CountEachHashtag.java](#)

Expected Output

```

Enter a list of hashtags ('end' to end):
love instagood photooftheday fashion beautiful happy love happy
photooftheday love happy happy fashion
summer selfie summer happy art friends followme love happy
end

```

```

11 hashtags minimum=1, maximum=6, average=2.000000
**** #love(4)
* #instagood(1)
** #photooftheday(2)
** #fashion(2)
* #beautiful(1)
***** #happy(6)
** #summer(2)
* #selfie(1)
* #art(1)
* #friends(1)
* #followme(1)
A * represents 1 use of tag.

```

Exercise TA: Demonstration and Discussion

This exercise is utilized to determine 3 points of the lab grade. We suggest reviewing these before you discuss these with your TA.

1. Demonstrate and Discuss Exercise A: Debug Number List, particularly the review questions.
2. Demonstrate and Discuss Exercise B: Hashtag Count Chart
3. Demonstrate and Discuss Exercise C: Count per Hashtag Chart



Exercise D: More Trace and Explain

Draw a diagram showing memory for each to help explain what is happening. *With your partner, trace and explain each of the following (also found in [traceExplain2.txt](#)). Check your understanding using your IDE.*

1. Using the following definitions:

capacity: the length of the elementData array within ArrayList¹.

size: the value returned by the ArrayList size() method. In other words, the number of elements currently in the elementData array within the ArrayList.

What are the initial capacity and resulting size of each of the following?

```

ArrayList<Integer> list1;

ArrayList<Integer> list2 = new ArrayList<Integer>();

ArrayList<Integer> list3 = new ArrayList<Integer>(1000);

ArrayList<Integer> list4 = new ArrayList<Integer>();
list4.add( 98 );
list4.add( 98 );

```

2. Determine how many instances/objects of Integer and ArrayList are created explicitly (new) and

¹ As a user of the ArrayList we only have control of setting the initial capacity. If adding an element to an ArrayList will exceed the capacity of the internal array, then the add method will create a new, longer array and copy all the elements over replacing the current internal array. To a user of the ArrayList, it automatically grows.

implicitly (autoboxing). Drawing a diagram may help.

```
Integer num = new Integer(1);
Integer num2 = 2;
Integer num3 = 3 + num2;
ArrayList<Integer> nums = new ArrayList<>();
for ( int i = 14; i > 10; i--) {
    nums.add( i);
}
System.out.println( nums);
int num4 = nums.get(0);
```

Did you identify 8 instances created?

```
3. ArrayList<Character> chars = new ArrayList<>();
   chars.add('A');
   chars.add('B');
   chars.set(1, 'C');
   chars.add(0, 'D');
   chars.remove(1);
   for ( int i = 0; i < chars.size(); i++) {
       System.out.print( chars.get(i) + " ");
   }

4. ArrayList<Character> chars = new ArrayList<>();
   chars.add('A');
   chars.set(0, 'B');
   chars.add('C');
   chars.add(0, 'D');
   chars.remove(1);
   ArrayList<Character> chars2 = new ArrayList<>();
   chars2.add('D');
   chars2.add('C');
```

`System.out.println(chars.equals(chars2));`

The equals method being called is in which class?

5. What is the capacity and size of the list?

```
ArrayList<Integer> list = new ArrayList<>(4);
for ( int i = 0; i < 3; i++) {
    list.add( i);
}
```

`System.out.println(list.toString());`

May be helpful to review [ArrayList constructors](#)

6. Draw a diagram showing all references.

```
ArrayList<ArrayList<String>> words = new ArrayList<>();
words.add(new ArrayList<String>());
words.add(new ArrayList<String>());
words.add(new ArrayList<String>());
```

```

words.get(0).add("The");
words.get(1).add("best");
words.get(1).add(0, "thing");
words.add(words.get(0));
words.get(2).add("about");
words.add(1, words.get(0));
words.get(1).add(1, "a");
words.get(2).add(1, "boolean");

```

7. Does methodB add true to the list in main?

```

public static void methodB(ArrayList<Boolean> items) {
    items = new ArrayList<Boolean>();
    items.add( true);
    return;
}

public static void main(String []args) {
    ArrayList<Boolean> list = new ArrayList<>();
    methodB( list);
    System.out.println( list);
}

```

8. When the Scanner instance is advanced in readInt is that change in Scanner's internal pointer also changed when accessed from main?

```

public class Trace {
    public static int readInt(Scanner input) {
        if (input.hasNextInt()) {
            return input.nextInt();
        } else {
            input.nextLine();
            return -1;
        }
    }

    public static void main(String[] args) {
        Scanner scnr = new Scanner("1 5 4 2 3");
        int num = readInt(scnr);
        System.out.println("num: " + num);
        System.out.println("next: " + scnr.nextInt());
    }
}

```

Additional Learning Materials

When you have mastered everything in this lab, *and previous labs*, then you are welcome to learn from additional learning resources available on the web and beyond this course:

<https://codingbat.com/java>

<https://www.khanacademy.org/computing/computer-programming>

<https://techdevguide.withgoogle.com/>
<https://code.org/>
<http://programmingbydoing.com/>

Note: Team Lab is focused on pair programming and open discussion and so it is not appropriate to work on individual programming assignments.

Contributions by Marc Renault, Yien Xu, Connor Waity.
Lab © 2018-2020 Jim Williams

EX. A

```
import java.util.ArrayList;
import java.util.Random;

public class NumberList {

    /**
     * Creates an ArrayList of size, populated with numbers
     * from 1 to 10 generated using rnd.nextInt().
     *
     * @param rnd A provided random number generator.
     * @param size The size of the list to create.
     * @return The list of numbers.
     */
    public static ArrayList<Integer> createList(Random rnd, int size) {
        ArrayList<Integer> list = new ArrayList<>(size);
        for ( int i = 0; i < size; ++i) {
            list.add( rnd.nextInt(10) + 1);
        }
        return list;
    }

    /**
     * The resulting list should not have the same number
     * twice in a row. Checks and removes duplicate values.
     *
     * Example:
     * [3,4,3,3] should result in [3,4,3]
     *
     * @param list A list of numbers
     */
    public static void remove2InARow(ArrayList<Integer> list) {
        for ( int i = 0; i < list.size() - 1; ++i) {
```

```

        if ( list.get(i) == list.get(i + 1)) {
            list.remove(i);
        }
    }
}

public static void main(String[] args) {
    final int COUNT = 10;
    Random rand = new Random(1); //don't change seed

    ArrayList<Integer> list = createList( rand, COUNT);
    System.out.println("initial:\n" + list);

    removeInARow(list);
    System.out.println("after removing contiguous duplicates:\n" + list);

    System.out.println("last element: " + list.get( list.size() - 1));
}
}

```

Ex. B

```

import java.util.ArrayList;
import java.util.Scanner;

public class Counter {

    public static void main(String[] args) {
        ArrayList<Integer> dailyCounts = new ArrayList<>();
        Scanner input = new Scanner(System.in);

        System.out.print("Enter a hashtag: ");
        String hashtag = input.nextLine().trim();
        hashtag = hashtag.startsWith("#") ? hashtag : "#" + hashtag;

        System.out.println("Enter count of " + hashtag + " tweets per day (negative to
end).");

        ArrayList<Integer> days = new ArrayList<>(7);

        for (int i = 0; i < 7; i++) {
            days.add(input.nextInt());
        }
    }
}

```

```

    }

    //determine total number of days of data
    //determine minimum
    int minimum = days.get(0);
    int sum = 0;
    for (int i = 0; i < 6; i++) {
        sum += days.get(i);
        if (days.get(i) < minimum) {
            minimum = days.get(i);
        }
    }

    //determine maximum
    int maximum = days.get(0);
    for (int i = 0; i < 6; i++) {
        if (days.get(i) > maximum) {
            maximum = days.get(i);
        }
    }

    //determine average
    double average = (double) sum / 6;

    //draw a bar chart where maximum / 50 is represented by one *.
    int onestar = maximum / 50;
    for (int i = 0; i < 6; i++) {
        for (int j = 0; j < days.get(i) / onestar; j++) {
            System.out.print("*");
        }
        System.out.println("(" + days.get(i) + ")");
    }
    System.out.println("A * represents " + onestar + " tweets.");
}
}

```