More Java Objects - ArrayLists

Marco Arsenault

During assignment 1

- I mentioned Java has arrays (similar to lists in python)
- Very limited functionality
- Immutable (once created cannot add to it or remove)
- Simple in implementation but not great in practice

Adding an element to a list in python

```
lst = [4, 2, 6]

lst.append(10)

## lst is now [4, 2, 6, 10]
```

Adding an element to an array in Java

```java
public class Main {
    public static void main(String[] args) {

        int[] arr = {4, 2, 6};

        // make a new array to fit the new element
        int[] newArr = new int[arr.length + 1];

        // loop over the old array copying each element to the new array
        for(int i = 0; i < arr.length; i++) {
            newArr[i] = arr[i];
        }

        // add the 10 in the last position of the new array
        newArr[newArr.length - 1] = 10;
        // update arr to be the value of newArr
        arr = newArr;

    }
}
```

Arrays suck

- Thankfully we have created a class that can make life easier
- In Intellij type in ArrayList and the click the autocomplete option it gives
- It should automatically add `import java.util.ArrayList;` at the top of the file. If not then just add the import yourself.

ArrayList - Disclaimer

- First thing to note: We are using ArrayLists because they are used in processing. Some external resources will suggest declaring a List over an ArrayList. This will work in vanilla Java but not in processing to let's stick with ArrayLists

Small Caveat of ArrayLists...

- Only works on Objects
- int , double, float, etc are primitive types and therefore are not Objects…
- Dont worry Java has a wrapper class (Think of it as a class to add functionality to the base int) for each
- The classes are the full names capitalized. Integer, Double, Float, Character, etc

ArrayList Syntax

- The data type is ArrayList<Type>
- The Type **MUST** be an object.
- Example: `ArrayList<Integer> fancyArr = new ArrayList<>();`

Initializing an ArrayList

```
ArrayList<String> places = new ArrayList<String>();

places.add("Buenos Aires");

places.add("Córdoba");

places.add("La Plata");
```

Initializing an ArrayList V2

```
ArrayList<String> places = new ArrayList<String>(Arrays.asList("Buenos Aires", "Córdoba", "La Plata"));
```

Slide 4 revised

```java
import java.util.ArrayList;
import java.util.Arrays;

public class Main {
    public static void main(String[] args) {

        ArrayList<Integer> lst = new ArrayList<Integer> (Arrays.asList(4,2,6));
        lst.add(10);

    }
}
```

ArrayList Methods

All the fancy things we can do with ArrayLists

https://www.w3schools.com/java/java_ref_arraylist.asp

Super important methods

add(type) - adds to the list

remove(int index) - removes from the list

size() - The objects version of .length

get(int index) - how we access the list

sort() - I hope I dont need to explain this one

Java Example Summing up a random list

```java
import java.util.ArrayList;
import java.util.concurrent.ThreadLocalRandom;

public class Main {
    public static void main(String[] args) {
        // make an empty list
        ArrayList<Integer> lst = new ArrayList<Integer> ();
        int listSize = 10;
        int min = 1;
        int max = 100;
        int randomNum;
        // create a random list of list size numbers from min to max
        for (int i = 0; i < listSize; i++) {
            randomNum = ThreadLocalRandom.current().nextInt(min,  bound: max + 1);
            lst.add(randomNum);
        }

        int sum = 0;
        // loop over the random arary and sum it all up
        for (int i = 0; i < lst.size(); i++) {
            sum += lst.get(i);
        }
        System.out.println("The random sum is " + sum);
    }
}
```

Array List practice

Make a new class called ArrayPractice and write 3 methods.

// findMax(ArrayList<String>) -> int

// purpose: takes in an arraylist and returns the length of the longest word

// reverse(ArrayList<int>) -> ArrayList<int>

// purpose: takes in a list of ints and returns that same list reversed (not sorted in reversed order). That is {4,7,2,9} becomes {9,2,7,4}

// sumAttribute (ArrayList<Your object type>) -> double

// purpose: take in a list of the object you have been making in class. This methods should sum up the values of a single numeric attribute from your class. For example for cats we could sum up all their ages or weights, if you are doing shoes you could sum up the price, etc.