Java Class

Arrays

12th June 2020

OVERVIEW

The world of variables is a crazy one, we have our basics (int, double, boolean, String) and then custom variables such as Scanner, Random, or the ones you developed like Car and Clock. Now, what if we want to store all these in one neat, little package? Enter: ***The Array***

WHAT IS AN ARRAY?

An array is a programming structure (commonly known as a data structure) that allows us to hold multiple of a certain data type in a list. Think of it as a bullet point list of objects, like a shopping list.

HOW DO I MAKE AN ARRAY?

Creating your own array is very simple, no extra libraries are needed! You just need to follow the common syntax:

```
<datatype>[] <nameofvariable> = new <datatype>[<numberofslots>];
```

Now, this just looks like a load of gunk right? Let's see what happens if we want to make an array of 20 integers!

```
int[] myIntArray = new int[20];
```

Boom! Now we have an array of 20 integers, but be warned! There are currently 20 0's in this array. When an array is created, it gets filled with 0's, so we need to change these numbers to numbers we want. (NOTE: This occurs for doubles as well. If you make a boolean array, you get all false values and for a string array you get all null values).

Written by Angelo Nelson

HOW DO I MODIFY AN ARRAY?

Easy! We *index* into the array!

HOW DO I INDEX INTO AN ARRAY?

Stop asking so many questions

Indexing is quite simple, all indexing means is that we are getting an *index* and going to that spot in the array. It is very important to remember that an array starts at spot 0 and goes up to spot size-1. So if I wanted to get the first element in the array, I would write

```
myIntArray[0];
```

And to access the 26th element I would write:

```
myIntArray[25];
```

Pay close attention to how I wrote 25 instead of 26, that is because we start a 0, so everything is offset by 1.

Now that we can index, let's modify!

To change the first element in the array, we use something like this:

```
myIntArray[0] = 10;
```

This changes the first element to a 10. So if we print out the value of the first element, we will get 10 instead of 0. (think getter and setters but for arrays).

Now that we understand how to change an element in an array, let's go a bit further. We want to loop through all elements in the array and modify each of them!

HOW DO I LOOP OVER AN ARRAY?

I said stop asking questions

Quite simple once you understand a few things about arrays. First, each array has a built-in public variable named length that you can access to tell you the size of the array. See below:

```
myIntArray.length;
```

We have the length of the array, now we know how long to loop for. But which loop to use?

The For Loop!

Yes, our old friend, the for loop! Remember, a for loop is an array's best friend. Why? We will see.

Remember how we use a for loop when we know how many times to loop over something? Well, we have the length variable, so we can use the for loop! If the name of our array is myIntArray, a loop to go over each element in the array would look like this:

```
for(int i = 0; i < myIntArray.length;i++){
System.out.println(myIntArray[i]);
}
```

Notice how we are using myIntArray[i], this means each iteration of the loop we are getting the *ith* element in the list, starting at 0 going to the length of the array. This will print out the entire contents of the array. Now, what if we want to fill the array with numbers corresponding to their index value? We would do something like this:

```
for(int i = 0; i < myIntArray.length;i++){
myIntArray[i] = i;
}
```

VOCABULARY!

- Array
- Index
- Length of an array

PROJECT

Even and Odd, but this time it is an array!

We covered even and odd checkers, we covered even and odd loops, but now we will cover even and odd lists. You may be asking yourself, "Why is he having us do so much with even and odd numbers?". Well, the answer is: because it forces you to understand modulus and I think it's neat.

So for this project, you will:

1. **Create an integer array with 20 slots**
2. **Fill the array with the first 20 numbers (so 0-20)**
3. **Print out only the even contents of the array**
4. **Print out only the odd contents of the array**

Written by Angelo Nelson

HINT: Be sure you don't only select the odd indices, you need to select 0,1,2,3,....,length. You will need three loops, one to fill the array, and two to print (one to print even and one to print odd). I have given you the filling loop. You need to design the printing loop.