Kevin McAleer

Week 4: Research

Strings

Concat, written as "concat(String str)", concatenates elements into a single string as specified by the author. This allows manipulation of the element as a whole, such as concatenating a name, or an address, and can be useful for increasing efficiency.

Equals, written as "equals(Object anObject)", calculates if the second element is equal to the first, with either true or false as the result. This can be useful for simple sorting, such as comparing two passwords.

Length, written as "length()", calculates how long your string is. Returning the length as an integer value allows you to work with non-integer methods in integer format, such as calculating how many letters are in a password, which is a String method.

Repeat, written as "repeat(int count)", concatenates the string as many times as the author specifies. This would have helped me with a lot of my earlier work if I knew it existed, but could also be used for manipulating data for calculations such as remotely changing a bar based on a password's strength.

Lowercase, written as "toLowerCase()", changes any uppercase letters to lowercase, which because of the rigidity of Java, tremendously increases the efficiency of workflow. You do not need to stop and change everything manually to make sure you do not cause errors, the built in method takes care of it for you.

Arrays

Get, written as "get(Object array, int index)", allows the author to work with an element stored in an array. This is chosen based on the specified integer. This allows the author to interact with lists.

GetLength, written as "public static int getLength(Object array)", works similarly to the string version, and simply identifies the size of an array as an integer. Working with the length of an array allows for performing equations on the array, such as identifying how many names are in a que.

GetInt, written as "public static int getInt(Object array, int index)", gets the number stored at the identified index. This allows, again, for working with the element in equations. If you had a for loop set up as a counter to index into an array, for most practical equations, you would need to offset the number by +1.

Set, written as "set(Object array, int index, Object value)", allows the author to insert elements in arrays based on parameters specified in method. Set allows the manipulation of lists, and especially encapsulated lists, which assists in security.

SetInt, written as “public static void setInt(Object array, int index, int i)”, changes the element “index” to a new integer “i”. This allows change for elements without rewriting the array. This can be useful for identifying how many people are in line, and how many people are available to help, and how many lines are open.

Works cited:

1. https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/reflect/Array.html
2. https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/String.html