Charles Fehringer Week 3 Research

Sources:

String JavaDocs: https://docs.oracle.com/javase/7/docs/api/java/lang/String.html

Array JavaDocs: https://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html

Prompt:

1. Select five methods from the String JavaDocs and describe the following for each:
    a. What is the method signature?
    b. What does the method do?
    c. Why would this method be useful (how could you use it)?
2. Select five methods from the Array JavaDocs and describe the following for each:
    a. What is the method signature?
    b. What does the method do?
    c. Why would this method be useful (how could you use it)?

String Methods:

1. .charAt
    a. This takes an int and returns the char at that index in the string the method was called on. This is useful to extract the char at the position from the String. One example of using it is to get the first letter of a string and check if it is capitalized.
2. .contains
    a. This takes a char and will search the string for the char. If the string has the char, it will return true. Otherwise it will return false. This is useful to check for specific characters in the string, such as checking for new lines so that you can remove them.
3. .matches
    a. This takes a string and will check if it is the same as the string the method is called on. This is useful to check if two string are the same since the = operator will only check if the reference is the same, which will fail if the strings are different objects.
4. .replace
    a. This takes two chars, the old char and the new one. This will search the string and replace every old char with the new one. This can be useful if there is a character in the string that you have to replace with a different one.
5. .concat
    a. This takes another string and will merge the new string onto the end of the string the method was called on. This is useful for combining two string together, such as combining a first and last name together.

Array Methods:

1. .binarySearch
    a. This takes the array to be searched and the value to search for. It will do a binary search to find the value in the array. If it finds the value in the array, it will return its index and an int. This can be useful if you need a simple way to check if an array has the value you are looking for. It will return -1 if the value is not in the array.
2. .copyOf
    a. This takes an array to copy and an int for the length of the new array. This will then copy the provided array and return a new array with the values from the original array. It also will truncate or pad the new array to match the length provided. It will initialize empty values (usually 0 or false) if the new length is longer than the original array. This can be useful if you want a copy of an array that you can edit without changing the old values.
3. .copyOfRange
    a. This takes an array to copy as well as two ints (from and to) for the indexes to copy from. It will return a new array that has copied the values from the original array. From must be a valid index the from the original array, but the to argument can be used to extend pass the original array length. This is useful for copying a specific part of the array.
4. .equals
    a. This takes two arrays. This will then compare them to see if their values are the same. It will return a boolean with the result. This is useful for comparing because the = operator will only check the references to see if they are the same.
5. .fill
    a. This takes and array and the value to fill the array will. It will fill the whole array with the provided value. This can be useful to initialize and array or to clear the array so that it can be used again.