**Chapter 9 - The String Class**

**Checking Equality of Strings**

if (string1.equals(string2)) - Good idea. This checks if the text contained in these two different Strings is the same.

if (string1 == string2) - Bad idea. This only checks to see if these two strings reference the same object.

\*Be reminded that Strings are case-sensitive, so if s1 = "cat" and s2="CAT", they will **not** be seen as equal.

**AP String Methods** s1 = "hotdog" and s2 = "hamburger"

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Purpose** | **Syntax** |
| compareTo | Returns an integer value when comparing two strings based on the location of their characters in the ASCII table. | s1.compareTo(s2) (Returns -14) |
| length() | Returns the length of the String as an integer. | s1.length() (Returns 6) |
| substring(int) | Returns a new String that starts at the index of the provided integer through the end of the String. | s3 = s2.substring(3)  s3 = "burger" |
| substring(int, int) | Returns a new String start at first integer index, ending at the second integer index. | s3 = s2.substring(3, 6)  s3 = "bur" |
| indexOf(String) | Return the index (int) of the first occurrence of the provided String within this String. If not found, -1 is returned. | int x = s1.indexOf("dog")  x = 3; |
| concat(String) | Usually to essentially "glue" two String together. Alternately, you could just use the + operator to do the same thing. | s3=s1.concat(s2) or s3 = s1 + s3  s3 = "hotdoghamburger" |
| charAt(int) | Returns the character found at that index | st = "burger"  st.charAt(3) 🡪 ‘g’ |

**Other String Methods**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Purpose** | **Syntax** |
| toLowerCase() | Returns a new String will all character in lowercase. | s3=s1.toLowerCase() |
| toUpperCase() | Returns a new String will all character in uppercase. | s3=s1.toUpperCase() |
| trim() | New String with all leading and ending whitespace removed. | s3 = s2.trim() |
| replace(char1, char2) | Replaces all char1 occurrences with char2 | s2.replace("h","y") |
| replaceAll(String1, String2) | Replaces all String1 occurrences with String2 | s2.replaceAll("ham","horse") |
| toCharArray() | Builds an array of character from the String | st = “hello”;  char[] let = st.toCharArray() |

**Character and Substring Methods**

|  |  |  |
| --- | --- | --- |
| **Method Name** | **Purpose** | **Syntax** |
| indexOf(char) | Returns the index (int) of the first occurrence of the char in the String or -1 if not found. | int x = s1.indexOf("do");  x=3. |
| indexOf(char, int) | Returns the index (int) of the first occurrence of the char in the String starting after the (int) index. (or -1 if not found) | int y =s1.indexOf('o', 2);  y = 4; |
| lastIndexOf(char) | Returns the index (int) of the last occurrence of the char in the String or -1 if not found. | int x = s2.lastIndexof("xo");  x= -1; (not found) |
| lastIndexOf(char, int) | Returns the index (int) of the last occurrence of the char in the String starting after the (int) index. (or -1 if not found) | int y =s2.lastIndexOf('r', 5);  y = 8; |

**valueOf() Method** (Convert data types to String) – The *valueOf()* is an overloaded method. (many methods of same name with different parameter lists) We use it to convert a primitive data type, such as int, double, or char to a String.

int x = 17; String s1 = String.valueOf(x); s1 = "17";

**Integer.parseInt(String)** – Converts a String to an int.

String st = “123”; int num = Integer.parseInt(st); num = 123;