7. String Handling

Lesson 1:
Why Strings in a program?

7.1. Why Strings in a program?

7.1.1. Introduction to String Handling

- In Java strings are objects designed to represent a sequence of characters.

- Because character strings are commonly used in programs, Java supports the ability to declare String constants and perform concatenation of Strings directly without requiring access to methods of the String class.

- A Java String is read-only and once created the contents cannot be modified.

7.1. Why Strings in a program?

7.1.2. Immutable

- String objects are said to be **immutable**—which just means that they cannot be changed. This means that you cannot extend or otherwise modify the string that an object of type String represents.

- When you execute a statement that combines existing String objects, you are *always* creating a new String object as a result.

- When you change the string referenced by a String variable, you throw away the reference to the old string and replace it with a reference to a new one.

7.1. Why Strings in a program?

7.1.3. Arrays of Strings

- You declare an array of String objects with the same mechanism that you used to declare arrays of elements for the basic types. You just use the type String in the declaration. For example, to declare an array of five String objects, you could use the statement:
  - String[] names = new String[5];

- You could also declare an array of String objects where the initial values determine the size of the array:
  - String[] colors = {“red”, “orange”, “yellow”, “green”, “blue”, “indigo”, “violet”};