String vs StringBuilder

Java provides a mutable class that lets us change its text value.

This is the StringBuilder Class.

Creating Instances

Instantiating String Objects

```
String hello = "Hello";
String helloWorld = "Hello" + " World";
String badHello = new String("Hello"); // Valid code, but redundant
```

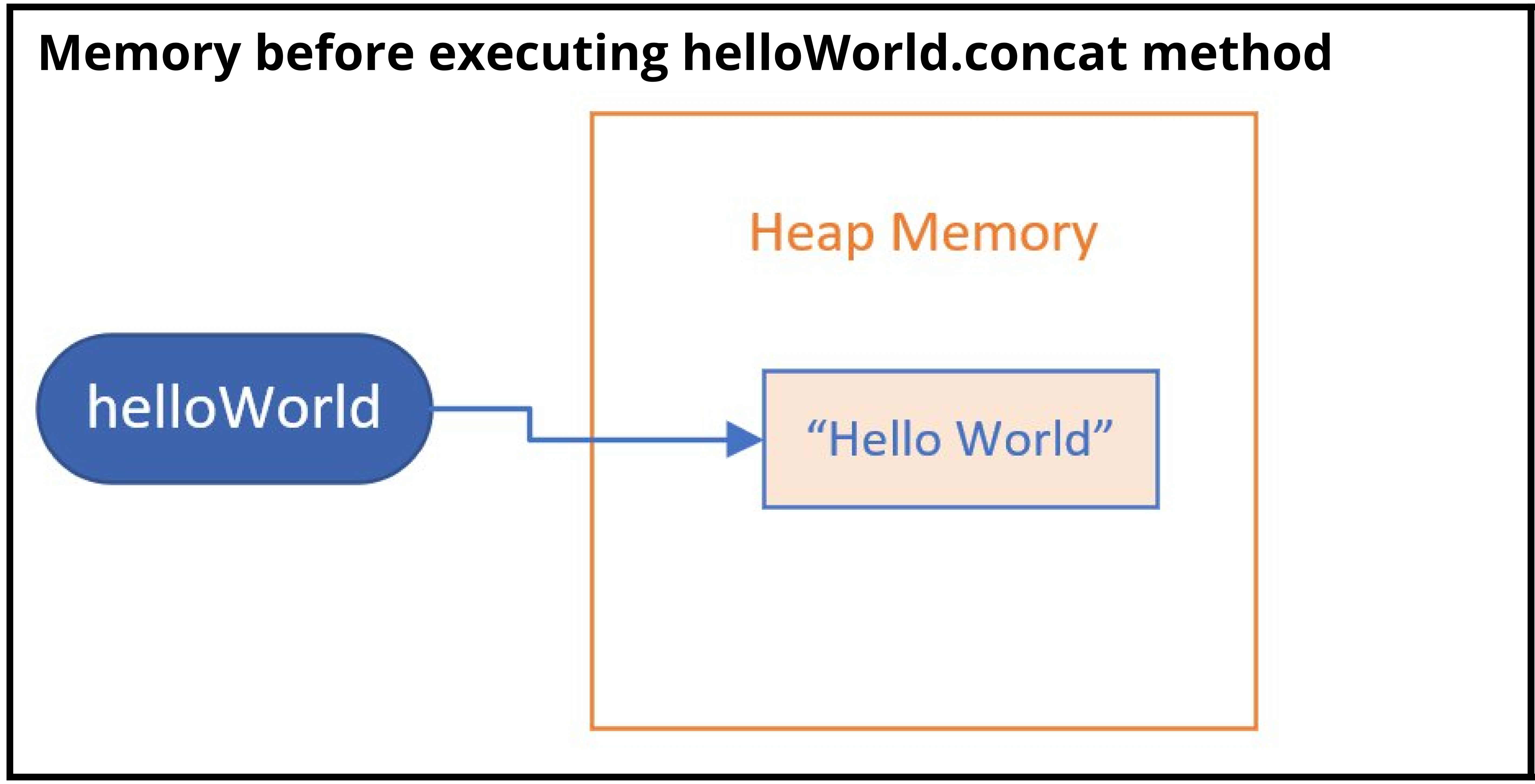
Instantiating StringBuilder Objects

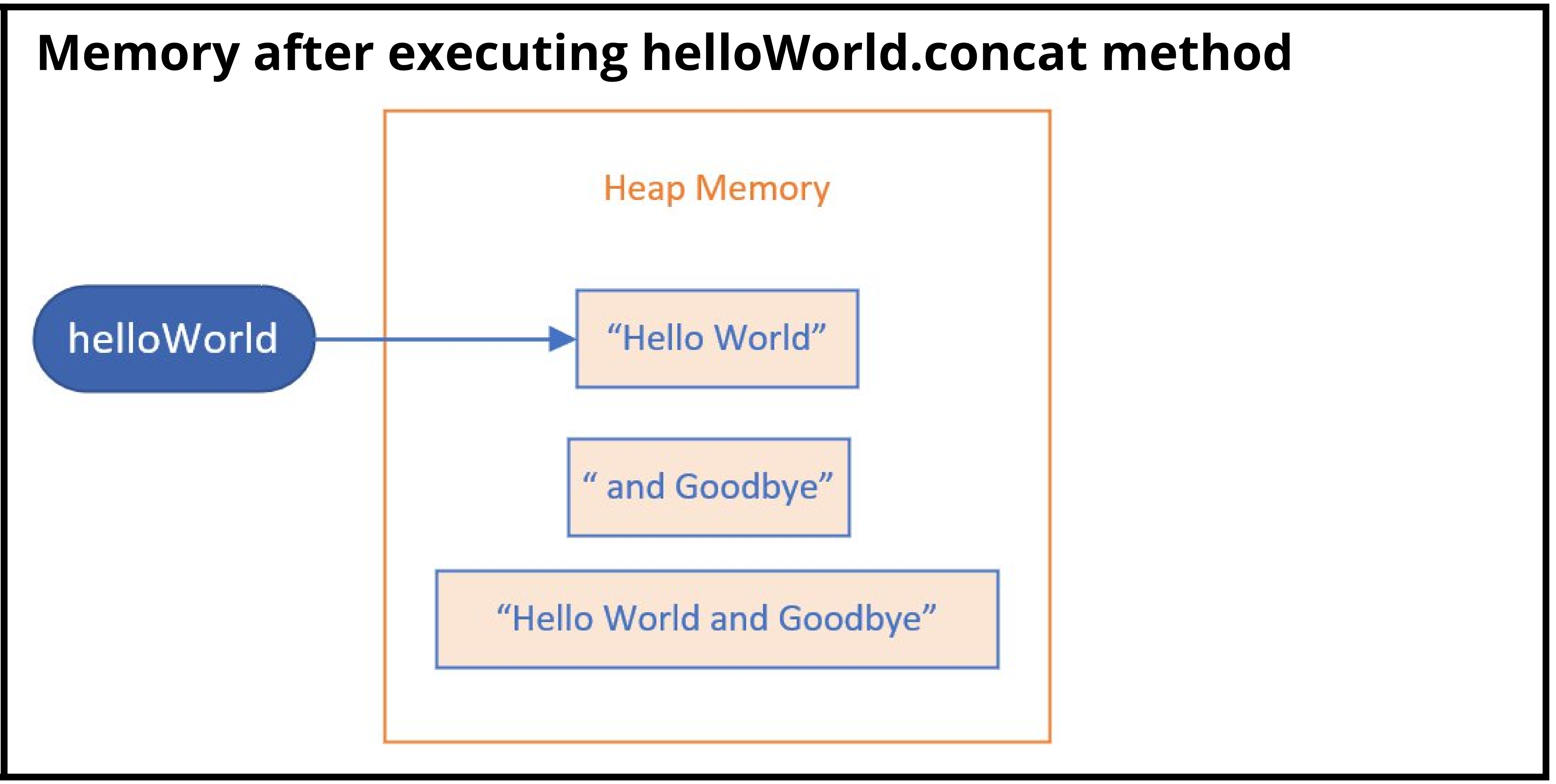
```
StringBuilder helloBuilder = new StringBuilder("Hello");
StringBuilder emptyBuilder = new StringBuilder();
StringBuilder emptyBuilder5 = new StringBuilder(5);
StringBuilder stringBuilder = new StringBuilder(helloBuilder);
```

There are four ways to create a new StringBuilder object using the new keyword:

- Pass a String.
- Pass no arguments at all.
- Pass an integer value.
- Pass some other type of character sequence (like StringBuilder).

String



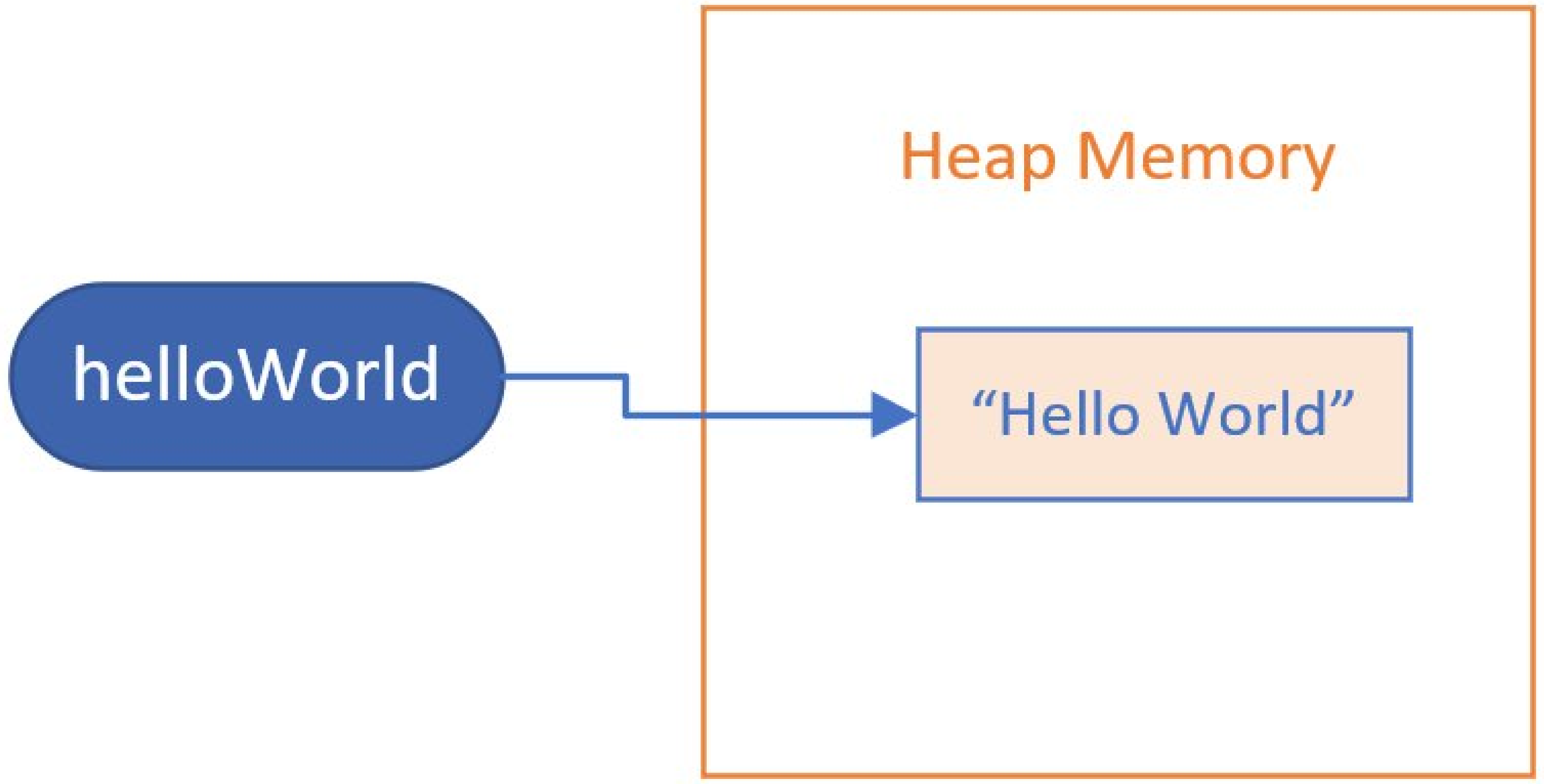


When I passed the String literal, "and Goodbye", to the concat method, this created an Object in memory for that literal, "and Goodbye".

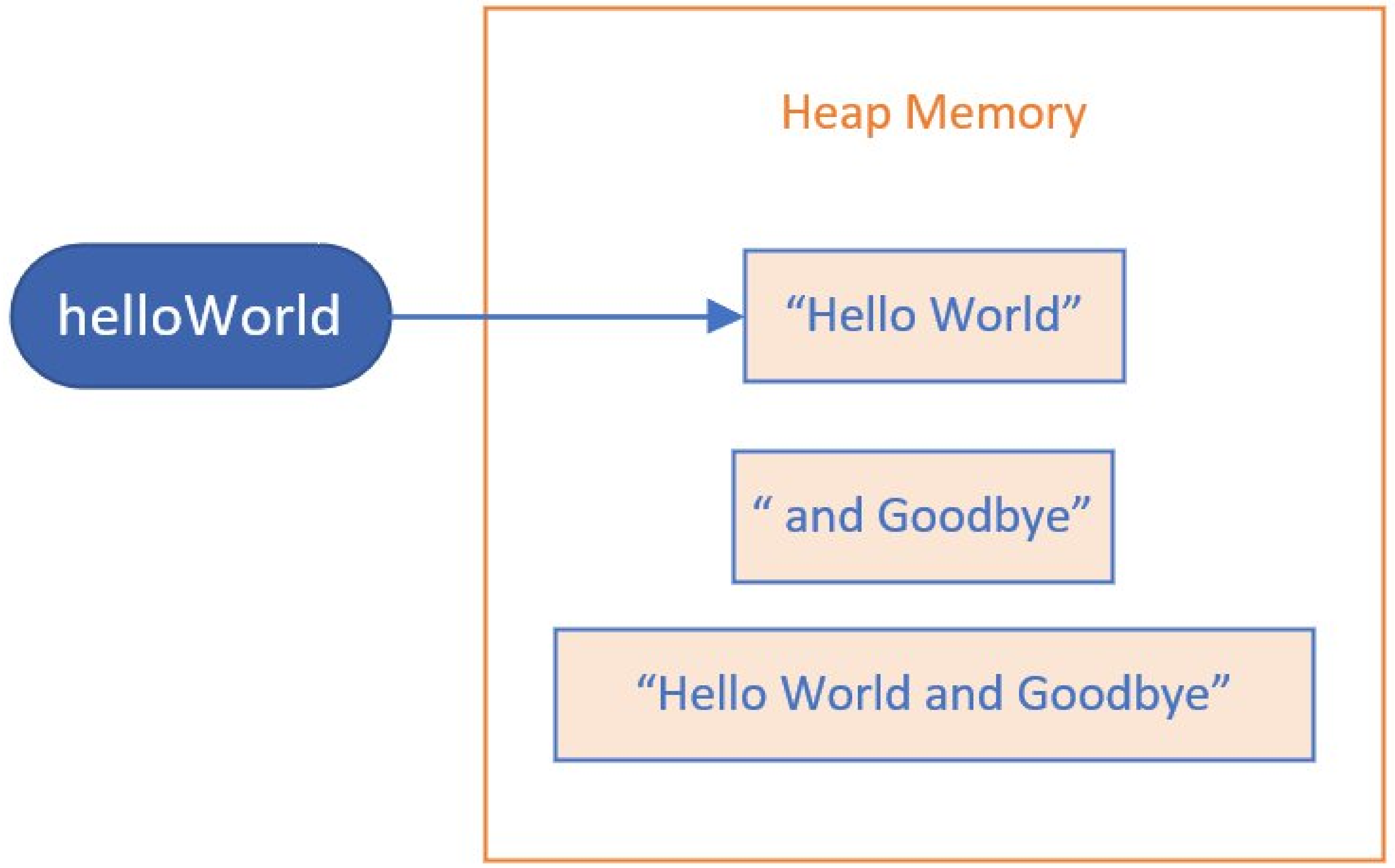
It also created the result of the concat method, the object, the String, that has the value, "Hello World and Goodbye".

String

Memory before executing helloWorld.concat method



Memory after executing helloWorld.concat method



These methods don't change the internals of the existing String object.

The String referenced by the helloWorld variable never changed, instead a new String was created by the method call.

StringBuilder

On this slide Strings and StringBuilders in different colors, with the StringBuilder object in green.

After the call to the append method, I still only have one StringBuilder object.

StringBuilder

The variable helloWorldBuilder is still referencing the same object, but the value of that object changed.

This is important because it means the character sequence in the StringBuilder changed.

String methods vs. StringBuilder methods

String methods create a new object in memory and return a reference to this new object.

StringBuilder methods return a StringBuilder reference, but it's really a self-reference.

Some methods unique to the StringBuilder class

A StringBuilder class has many similar methods to Strings.

But it also has methods to remove and insert characters or Strings. In addition, you can truncate the string builder's size.

method	description
delete deleteCharAt	You can delete a substring using indices to specify a range, or delete a single character at an index.
insert	You can insert text at a specified position.
reverse	You can reverse the order of the characters in the sequence.
setLength	setLength can be used to truncate the sequence, or include null sequences to 'fill out' the sequence to that length.