

**FIGURE 2.5 Some Methods in the Class String**

Method	Return Type	Example for String s = "Java";	Description
<code>charAt(index)</code>	char	<code>c = s.charAt(2); // c = 'v'</code>	Returns the character at <i>index</i> in the string. Index numbers begin at 0.
<code>compareTo(a_string)</code>	int	<code>i = s.compareTo("C++"); // i is positive</code>	Compares this string with <i>a_string</i> to see which comes first in lexicographic (alphabetic, with upper- before lowercase) ordering. Returns a negative integer if this string is first, zero if the two strings are equal, and a positive integer if <i>a_string</i> is first.
<code>concat(a_string)</code>	String	<code>s2 = s.concat("rocks"); // s2 = "Javarocks"</code>	Returns a new string with this string concatenated with <i>a_string</i> . You can use the + operator instead.
<code>equals(a_string)</code>	boolean	<code>b = s.equals("Java"); // b = true</code>	Returns true if this string and <i>a_string</i> are equal. Otherwise returns false.
<code>equalsIgnoreCase(a_string)</code>	boolean	<code>b = s.equals("java"); // b = true</code>	Returns true if this string and <i>a_string</i> are equal, considering upper- and lowercase versions of a letter to be the same. Otherwise returns false.
<code>indexOf(a_string)</code>	int	<code>i = s.indexOf("va"); // i = 2</code>	Returns the index of the first occurrence of the substring <i>a_string</i> within this string or -1 if <i>a_string</i> is not found. Index numbers begin at 0.
<code>lastIndexOf(a_string)</code>	int	<code>i = s.lastIndexOf("a"); // i = 3</code>	Returns the index of the last occurrence of the substring <i>a_string</i> within this string or -1 if <i>a_string</i> is not found. Index numbers begin at 0.
<code>length()</code>	int	<code>i = s.length(); // i = 4</code>	Returns the length of this string.
<code>toLowerCase()</code>	String	<code>s2 = s.toLowerCase(); // s2 = "java"</code>	Returns a new string having the same characters as this string, but with any uppercase letters converted to lowercase. This string is unchanged.
<code>toUpperCase()</code>	String	<code>s2 = s.toUpperCase(); // s2 = "JAVA"</code>	Returns a new string having the same characters as this string, but with any lowercase letters converted to uppercase. This string is unchanged.
<code>replace(oldchar, newchar)</code>	String	<code>s2 = s.replace('a', 'o'); // s2 = "Jovo";</code>	Returns a new string having the same characters as this string, but with each occurrence of <i>oldchar</i> replaced by <i>newchar</i> .
<code>substring(start)</code>	String	<code>s2 = s.substring(2); // s2 = "va";</code>	Returns a new string having the same characters as the substring that begins at index <i>start</i> through to the end of the string. Index numbers begin at 0.
<code>substring(start,end)</code>	String	<code>s2 = s.substring(1,3); // s2 = "av";</code>	Returns a new string having the same characters as the substring that begins at index <i>start</i> through to but not including the character at index <i>end</i> . Index numbers begin at 0.
<code>trim()</code>	String	<code>s = " Java ";  s2 = s.trim(); // s2 = "Java"</code>	Returns a new string having the same characters as this string, but with leading and trailing whitespace removed.