

## the Math class

Some methods can be invoked through their class name, without having to instantiate an object of the class first. These are called *class methods* or *static methods*. Let's look at an example.

The Math class lets us do a large number of basic mathematical functions. The Math class is part of the Java standard class library and is defined in the `java.lang` package. Figure 2.12 lists several of its methods.

AP*→	<code>static int abs (int num)</code>
AP*→	<code>static double abs (double num)</code> Returns the absolute value of num.
	<code>static double acos (double num)</code>
	<code>static double asin (double num)</code>
	<code>static double atan (double num)</code> Returns the arc cosine, arc sine, or arc tangent of num.
	<code>static double cos (double angle)</code>
	<code>static double sin (double angle)</code>
	<code>static double tan (double angle)</code> Returns the angle cosine, sine, or tangent of angle, which is measured in radians.
	<code>static double ceil (double num)</code> Returns the ceiling of num, which is the smallest whole number greater than or equal to num.
	<code>static double exp (double power)</code> Returns the value e raised to the specified power.
	<code>static double floor (double num)</code> Returns the floor of num, which is the largest whole number less than or equal to num.
AP*→	<code>static double pow (double num, double power)</code> Returns the value num raised to the specified power.
AP*→	<code>static double random ()</code> Returns a random number between 0.0 (inclusive) and 1.0 (exclusive).
AP*→	<code>static double sqrt (double num)</code> Returns the square root of num, which must be positive.

figure 2.12 Some methods of the Math class

	<b>String (String str)</b> Constructor: creates a new string object with the same characters as str.
	<b>char charAt (int index)</b> Returns the character at the specified index.
AP*→	<b>int compareTo (String str)</b> Returns a number indicating whether this string comes before (a negative return value), is equal to (a zero return value), or comes after (a positive return value), the string str.
	<b>String concat (String str)</b> Returns a new string made up of this string added to (concatenated with) str.
AP*→	<b>boolean equals (String str)</b> Returns true if this string contains the same characters as str (including upper or lowercase) and false if it does not.
	<b>boolean equalsIgnoreCase (String str)</b> Returns true if this string contains the same characters as str (ignoring upper and lowercase) and false if it does not.
AP*→	<b>int indexOf (String str)</b> Returns the position of the first character in the first occurrence of str in this string.
AP*→	<b>int length ()</b> Returns the number of characters in this string.
	<b>String replace (char oldChar, char newChar)</b> Returns a new string that is identical with this string except that every oldChar is replaced by newChar.
AP*→	<b>String substring (int offset, int endIndex)</b> Returns a new string that is a subset of this string starting at index offset and ending with the character at position endIndex-1.
AP*→	<b>String substring (int offset)</b> Returns a new string that starts at index offset and extends to the end of the string.
	<b>String toLowerCase ()</b> Returns a new string that is the same as this string except all uppercase letters are changed to lowercase.
	<b>String toUpperCase ()</b> Returns a new string that is the same as this string except all lowercase letters are changed to uppercase.

figure 2.5 Some methods of the String class

Notice that some of the String methods refer to the *index* of a particular character. The index is a character's position in the string. The index of the first character in a string is zero, the index of the next character is one, and so on. Therefore in the string "Hello", the index of the character 'H' is zero, 'e' is one, and so on.