Clase Math en Java

¿Qué es la clase Math de Java?

En ocasiones nos vemos en la necesidad de incluir cálculos, operaciones matemáticas, estadísticas, etc. en nuestros programas Java.

Es cierto que muchos cálculos se pueden hacer simplemente utilizando los operadores aritméticos que Java pone a nuestra disposición, pero existe una opción mucho más sencilla de utilizar, sobre todo para cálculos complicados. Esta opción es la clase Math del paquete java.lang.

La clase Math nos ofrece numerosos y valiosos métodos y constantes static, que podemos utilizar tan sólo anteponiendo el nombre de la clase, (Math.metodo() ó Math.CONSTANTE). Además podemos omitir el nombre de la clase si la importamos:

```
import static java.lang.Math. *;
```

Métodos más utilizados de la clase Math (I)

Método	Descripción	Ejemplo de uso	Resultado
abs	Valor absoluto	int x = Math.abs(2.3);	x = 2;
exp	Exponenciación neperiana	double x = Math.exp(1);	x = 2.71828182845904;
log	Logaritmo neperiano	double x = Math.log(2.7172);	x = 0.99960193833500;
pow	Potencia	double x = Math.pow(2,3);	x = 8.0;
floor	Redondeo al entero menor	double x = Math.floor(2.5);	x = 2.0;
sen	Seno	double x = Math.sin(0.5);	x = 0.4794255386042;
cos	Coseno	double x = Math.cos(0.5);	x = 0.87758256189037;
tan	Tangente	double x = Math.tan(0.5);	x = 0.54630248984379;

La clase Math no se puede instanciar y no se puede heredar de ella (final) y nos ofrece infinidad de métodos static

Métodos más utilizados de la clase Math (II)

Método	Descripción	Ejemplo de uso	Resultado
max	Valor máximo	double x = Math.*max*(234,34);	x = 234.0;
min	Valor mínimo	double x = Math.min((45.6, 34.2));	x = 34.2;
sqrt	Raiz cuadrada	double x = Math.sqrt(100);	x = 10.0;
toDegrees	Convierte radianes a grados	double x = Math.toDegrees(1.57);	x = 90.0 ;
toRadians	Convierte grados a radianes	double x = Math.toRadians(90);	x = 1.5707963267948966 ;
ceil	Redondeo al entero mayor	double x = Math.ceil(2.5);	x = 3.0;
random	Número aleatorio	double x = Math.ramdom();	x = 0.20614522323378;
round	Redondeo	double x = Math.round(2.5);	x = 3;

Constantes de la clase Math

Constante	Descripción	Valor
E	Base de los logaritmos naturales	double *E* = 2.7182818284590452354;
PI	Número PI	double *PI* = 3.14159265358979323846;
DEGREES_TO_RADIANS	Ratio para pasar grados a radianes	double *DEGREES_TO_RADIANS* = 0.0174532925199433295;
RADIANS_TO_DEGREES	Ratio para pasar radianes a grados	double *RADIANS_TO_DEGREES* = 57.29577951308232;

Diferencia entre los métodos round, ceil y floor

Los métodos round, ceil y floor se usan para obtener un entero próximo a un número decimal y tienen similitudes, de hecho en algunos casos devuelven el mismo resultado.

Sin embargo también tienen diferencias que es interesante conocer. Estas tres funciones se aplican sobre valores numéricos decimales y retornan un valor numérico que en el caso de round es un entero long, mientras que en el caso de floor y ceil retornan un valor de tipo double coincidente o equivalente con un entero.

El método round redondea siempre al entero más próximo, por ejemplo 2.6 redondea a 3 mientras que -2.6 redondea a -3. Si el decimal está exactamente entre dos valores se redondea al entero superior más próximo (por ejemplo 2.5 redondea a 3 y -2.5 redondea a -2).

El método floor devuelve el entero menor, por ejemplo 2.9 quedaría en 2.0 y -2.9 quedaría en -3.0. También 2.1 quedaría en 2.0 y -2.1 quedaría en -3.0.

El método ceil devuelve el entero mayor, por ejemplo 2.9 quedaría en 3.0 y -2.9 quedaría en -2.0. También 2.1 quedaría a 3.0 y -2.1 quedaría en -2.0.

Tabla ejemplo métodos round, ceil y floor

Valor inicial	2.6	-2.6	2.4	-2.4	2.5	-2.5
round	3	-3	2	-2	3	-2
ceil	3.0	-2.0	3.0	-2.0	3.0	-2.0
floor	2.0	-3.0	2.0	-3.0	2.0	-3.0

Métodos relacionados con ángulos

Los métodos relacionados con ángulos (atan, cos, sin, tan, etc.) trabajan en radianes.

Por tanto, para operar con grados, tendremos que realizar la conversión oportuna.

La propia clase Math facilita los métodos toRadians para transformar grados sexagesimales en radianes y toDegrees para transformar radianes en grados sexagesimales, aunque las conversiones pueden no ser totalmente precisas.

El método random

El método random,permite generar números aleatorios en el rango]0,1[.

Por tanto el 0 y el 1 están excluidos.

Por ejemplo, si queremos elegir un número aleatorio entre los valores 18 y 65, ambos inclusive, podríamos solucionarlo de esta forma:

int n = (int) (Math.random() * (<número_máximo + 1> - <número_mínimo>)) + <numero_mínimo>;

int n = (int) (Math.random() * (66 - 18)) + 18;

La clase Random (off topic)

Otra forma de obtener un número.random consiste en utilizar la clase Random del paquete java.utils.

Por ejemplo para obtener un número aleatorio entre 0 (incluído) y un número determinado (excluido) [0,numero elegido[utilizando la clase Random podríamos utilizar el método nextInt():

Por ejemplo, si queremos elegir un número aleatorio entre los valores 0 (incluído) y 65 (excluido), podríamos solucionarlo de esta forma:

```
Random r = new Random();
int n = r.nextInt(65);
```

Dada la especificación del método nextInt si se desea un número aleatorio entre un rango distinto que no empiece en el 0 habría que realizar una pequeña operación matemática:

```
int n = r.nextInt(max - min + 1) + min;
```

Parte entera de un número real (off topic)

No hay un método directo en la clase Math para obtener la parte entera de un número real, pero para estos casos, se puede obtener de la siguiente manera:

```
int x = (int)(8.7);   // x = 8;

int x = (int)(-8.7);  // x = -8;
```

Tenemos que aclarar que obtener la parte entera no es lo mismo que redondear.