### LA CLASE Math

La clase Math representa la librería matemática de Java. Las funciones que contiene son las de todos los lenguajes, parece que se han metido en una clase solamente a propósito de agrupación, por eso se encapsulan en Math, y lo mismo sucede con las demás clases que corresponden a objetos que tienen un tipo equivalente (Character, Float, etc.). El constructor de la clase es privado, por los que no se pueden crear instancias de la clase. Sin embargo, Math es *public* para que se pueda llamar desde cualquier sitio y *static* para que no haya que inicializarla.

No es necesario importar la clase Math para utilizar sus métodos, puesto que es parte del paquete java.lang, que se importa automáticamente por el compilador.

#### **Funciones matemáticas**

```
para int, long, float y double
Math.abs(x)
Math.sin( double )
Math.cos( double )
Math.tan( double )
Math.asin( double )
Math.acos( double )
Math.atan( double )
Math.atan2( double, double )
Math.exp( double )
Math.log( double )
Math.sqrt( double )
Math.ceil( double )
Math.floor( double )
Math.rint( double )
Math.pow(a,b)
Math.round(x)
                  para double y float
                  devuelve un double
Math.random()
                  para int, long, float y double
Math.max(a,b)
                  para int, long, float y double
Math.min(a,b)
               para la base exponencial
Math.E
               para PI
Math.PI
```

He aquí un ejemplo de uso de algunas funciones de la clase Math:

```
class Mates {
   public static void main( String args[] ) {
      int x;
      double rand, y, z;
      float max;
      rand = Math.random();
      x = Math.abs( -123 );
      y = Math.round( 123.567 );
      z = Math.pow( 2, 4 );
      max = Math.max( (float)1e10, (float)3e9 );

      System.out.println( rand );
      System.out.println( x );
      System.out.println( y );
      System.out.println( z );
      System.out.println( z );
      System.out.println( max );
    }
}
```

#### Miembros dato constantes

La clase *Math* define dos constantes muy útiles, el número pi y el número e.

```
public final class Math {
   public static final double E = 2.7182818284590452354;
   public static final double PI = 3.14159265358979323846;
   //...
}
```

El modificador **final** indica que los valores que guardan no se pueden cambiar, son valores constantes.

Se accede a estas constantes desde la clase Math, de la siguiente forma

```
System.out.println("Pi es " + Math.PI);
System.out.println("e es " + Math.E);
```

#### **Funciones miembro**

La clase Math define muchas funciones y versiones distintas de cada función.

Por ejemplo, para hallar el valor absoluto de un número define las siguientes funciones. Se llama a una u otra dependiendo del tipo de dato que se le pasa en su único argumento.

```
public final class Math {
    public static int abs(int a) {
        return (a < 0) ? -a : a;
    }
    public static long abs(long a) {
        return (a < 0) ? -a : a;
    }
    public static float abs(float a) {
        return (a < 0) ? -a : a;
    }
    public static double abs(double a) {
        return (a < 0) ? -a : a;
    }
}//...
}</pre>
```

Por ejemplo, hallar el valor absoluto de los siguientes números:

```
int i = -9;
double x = 0.3498;

System.out.println("|" + i + "| es " + Math.abs(i));
System.out.println("|" + x + "| es " + Math.abs(x));
```

Math.abs(i), llama a la primera versión, y Math.abs(x) llama a la última versión.

# **Funciones trigonométricas**

En las funciones trigonométricas los argumentos se expresan en radianes. Por ejemplo, el ángulo 45º se convierte en radianes y luego se halla el seno, el coseno y la tangente

```
double angulo = 45.0 * Math.PI/180.0;
System.out.println("cos(" + angulo + ") es " + Math.cos(angulo));
System.out.println("sin(" + angulo + ") es " + Math.sin(angulo));
System.out.println("tan(" + angulo + ") es " + Math.tan(angulo));
```

Para pasar de coordenadas rectangulares a polares es útil la función *atan2*, que admite dos argumentos, la ordenada y la abscisa del punto. Devuelve el ángulo en radianes.

```
double y=-6.2; //ordenada
double x=1.2; //abscisa
System.out.println("atan2(" + y+" , "+x + ") es " + Math.atan2(y, x));
```

## Funciones exponencial y logarítmica

La función exponencial exp devuelve el número e elevado a una potencia

```
System.out.println("exp(1.0) es " + Math.exp(1.0));
System.out.println("exp(10.0) es " + Math.exp(10.0));
System.out.println("exp(0.0) es " + Math.exp(0.0));
```

La función *log* calcula el logaritmo natural (de base e) de un número

```
System.out.println("log(1.0) es " + Math.log(1.0));
System.out.println("log(10.0) es " + Math.log(10.0));
System.out.println("log(Math.E) es " + Math.log(Math.E));
```

# Función potencia y raíz cuadrada

Para elevar un número x a la potencia y, se emplea pow(x, y)

```
System.out.println("pow(10.0, 3.5) es " + Math.pow(10.0,3.5));
```

Para hallar la raíz cuadrada de un número, se emplea la función sgrt

```
System.out.println("La raíz cuadrada de " + x + " es " + Math.sqrt(x));
```

# Aproximación de un número decimal

Para expresar un número real con un número especificado de números decimales empleamos la función *round*. Por ejemplo, para expresar los números *x* e *y* con dos cifras decimales escribimos

```
double x = 72.3543;
double y = 0.3498;
System.out.println(x + " es aprox. " + (double)Math.round(x*100)/100);
System.out.println(y + " es aprox. " + (double)Math.round(y*100)/100);
```

Se obtiene 72.35 y 0.35 como cabría esperar. Fijarse que *round* devuelve un número entero **int** que es necesario promocionar a **double** para efectuar la división entre 100.

Si empleamos la función *floor* en vez de *round* obtendríamos

```
System.out.println(x + " es aprox. " + Math.floor(x*100)/100);
System.out.println(y + " es aprox. " + Math.floor(y*100)/100);
```

Se obtiene 72.35 y 0.34. La aproximación del primero es correcta ya que la tercera cifra decimal es 4 inferior a 5. La aproximación del segundo es incorrecta ya que la tercera cifra decimal es 9 mayor que 5. En la mayor parte de los cálculos se cometen errores, por lo que la diferencia entre *floor* y *round* no es significativa.

## El mayor y el menor de dos números

Para hallar el mayor y el menor de dos números se emplean las funciones *min* y *max* que comparan números del mismo tipo.

```
int i = 7;
int j = -9;
double x = 72.3543;
double y = 0.3498;

// para hallar el menor de dos número
    System.out.println("min(" + i + "," + j + ") es " + Math.min(i,j));
    System.out.println("min(" + x + "," + y + ") es " + Math.min(x,y));

// Para hallar el mayor de dos números
    System.out.println("max(" + i + "," + j + ") es " + Math.max(i,j));
    System.out.println("max(" + x + "," + y + ") es " + Math.max(x,y));
```

### **Números aleatorios**

La clase *Math* define una función denominada *random* que devuelve un número pseudoaleatorio comprendido en el intervalo [0.0, 1.0). Existe otra alternativa, se pueden generar números pseudoaleatorios a partir de un objeto de la clase Random, que llame a la función miembro *nextDouble*.

```
System.out.println("Número aleatorio: " + Math.random());
System.out.println("Otro número aleatorio: " + Math.random());
```