MÉTODOS ÚTILES

Hasta el momento conoce los operados aritméticos y cómo usarlos. Pero, no son suficientes para resolver todos los problemas, por ejemplo si necesita encontrar las raíces de una ecuación cuadrática, debe calcular una raíz cuadra y elevar un número al cuadrado. Alguien podría decir que puede usar el método de Newton y la multiplicación para la primera y segunda operación respectivamente. Sin duda sería un buen ejercicio para mejorar en programación, pero existen un grupo de métodos pre-construidos que vienen en los lenguajes de programación y que es necesario aprender a utilizar.

MÉTODOS MATEMÁTICOS

En matemáticas todos hemos visto funciones tales como seno, coseno u otras como los logaritmos naturales. Aprendimos a evaluar expresiones tales como seno(180º) y log(1/x), en las cuales lo primero que se hace es evaluar la expresión que se encuentra entre paréntesis, la cual se denomina argumento de la función. Lo segundo es evaluar la función por en sí misma, generalmente utilizando una calculadora.

Dentro del lenguaje de programación Java existe una clase que contiene muchos de estos métodos matemáticos. Esa clase se denomina *Math*. Dicha clase no sólo posee métodos sino también cuenta con los valores de las constantes *PI* y *E*.

En la documentación de Java se dice que la clase *Math* contiene métodos para realizar operaciones numéricas básicas, tales como las funciones exponenciales, logarítmicas, raíces cuadradas y trigonométricas.

Son muchos los métodos que posee la clase como para nombrarlos a todos. Es por ello que se mostrarán únicamente aquellos que considero como los mas utilizados. El detalle de todos los métodos los pueden encontrar en la documentación del lenguaje. Así:

- Ir a https://docs.oracle.com/en/java/
- Seleccionar el link Java SE documentation
- API Documentation (en la parte izquierda)
- En la columna Module, seleccionar *java.base*
- En la la columna Package, seleccionar *java.lang*

- Y buscar la clase *Math*, en la tabla Class Summary.

La siguiente tabla muestra algunas de las funciones de la clase *Math*.

Método	**Descripción**
abs(x)	Devuelve el valor absoluto del número x
cos(x)	Devuelve el coseno del ángulo x cuya unidad son los radianes
sin(x)	Devuelve el seno del ángulo x cuya unidad son los radianes
tan(x)	Devuelve la tangente del ángulo x cuya unidad son los radianes
log(x)	Para obtener el logaritmo de base *E* del argumento x
log10(x)	Obtiene el logaritmo de base 10 de x
max(x, y)	Devuelve el número mayor entre x e y
min(x, y)	Devuelve el número menor entre x e y
pow(x, y)	Eleva la base x a la potencia y
sqrt(x)	Calcula la raíz cuadrada del número x
toDegrees(x)	Convierte al ángulo x de radianes a grados
toRadians(x)	Convierte el ángulo x de grados a radianes

¿Qué hacer si necesito calcular esto: $\sqrt[3]{9}$? En el listado de la tabla anterior no se muestra un método que permita calcular la raíz cúbica de un número. Generalizando no existe un método que permita calcular la raíz *y* de un número *x* $\sqrt[y]{x}$.

Para ello es necesario recurrir a la siguiente equivalencia matemáticas: $\sqrt[y]{x} = x^{\frac{1}{y}}$ En términos de métodos sería: $\sqrt[y]{x} = pow(x,1.0/y)$ Observe como ubico 1.0 para señalar que el resultado sea real.

Dentro de los métodos en general, en donde se incluye a los matemáticos, existen algunos conceptos que se deben aclarar, estos se estudiarán a detalle más adelante (existe un apartado en donde se estudia a detalle los métodos). Ahora mismo es necesario que comprender que los métodos matemáticos poseen argumentos y devuelven valores.

Los argumentos son los valores que suministramos al método. En la tabla anterior *x* e *y* son los **argumentos** que se envían a los métodos. En la misma tabla la gran **mayoría de métodos devuelven valores de tipo real**. Dos **excepciones** son los métodos *max* y *min*.

Esos métodos devuelven uno de los argumentos *x* o *y*, por lo que el tipo de dato que devuelve depende del tipo de dato de esos argumentos. Es decir si *x* e *y* son enteros, los métodos *max* o *min*, devolverán un entero, en cambio si son reales, esos mismos métodos, devolverán un real. ¿Qué devolverían si *x* es entero e *y* es real? Más adelante conocerá cómo invocar a esos métodos y podrá responder a esa pregunta.

INVOCACIÓN DE MÉTODOS

Dentro de los lenguajes que siguen el paradigma de la orientación a objetos, el concepto de clase es fundamental. Si bien este documento no tiene como objetivo explicar programación orientada a objetos, es necesario mencionar algo sobre las clases.

Una clase es una abstracción del mundo real que incluye propiedades y métodos. Es de nuestro interés, por ahora, el tema de métodos y en particular aquellos que ya incorpora el lenguaje de programación.

Como se mencionó anteriormente, una de las clases que posee muchos métodos útiles es la clase *Math*. Para invocar a un método dentro de cualquier clase se debe seguir la siguiente notación:

```
<Nombre de la clase>.<método>(<argumentos>)

Math.sin(1)
```

El estándar del lenguaje de programación Java, permite diferenciar cada uno de sus elementos. Así, una clase inicia con un letra mayúscula, mientras que un método con una minúscula. El nombre de una clase es un sustantivo (en la medida de lo posible), el de un método es un verbo. Si usted recuerda esto podrá diferenciar entre variables, clases y métodos.

Las siguientes porciones de código muestran cómo invocar a varios métodos de la clase *Math*, desde *Jshell*.

```
jshell> Math.max(4, 6)
$7 ==> 6
```

```
jshell> Math.min(4, 6)
$8 ==> 4
```

```
jshell> Math.pow(2, 3)
$9 ==> 8.0
```

```
jshell> Math.pow(9, 1.0/3)
$12 ==> 2.080083823051904
```

```
jshell> Math.max(1, 1.0)
$11 ==> 1.0
```

En las expresiones anteriores puede encontrar la respuesta a una pregunta que se planteó anteriormente sobre el tipo de dato que se devuelve cuándo los parámetros son de diferente tipo de dato (ver último código).

El invocar a un método de la clase *Math* devuelve un valor, ese valor se puede asignar a una variable o utilizarlo en alguna operación o presentarlo o etc. Imagínelos como variables en donde puede ir una variable, puede ir un método de la clase *Math*. La porción de código de la

```
jshell> var valueMax = Math.max(10, 12)
valueMax ==> 12

jshell> var root = Math.pow(1, 1.0/valueMax)
root ==> 1.0

jshell> var root = Math.pow(8, 1.0/valueMax)
root ==> 1.189207115002721

jshell>
System.out.println(Math.toDegrees(root))
68.13654865658464
```

arriba muestra parte de los diferentes usos de los métodos. Mientras que la de abajo muestra un error al tratar de asignar un valor real a un variable entera

```
jshell> int value = Math.abs(-2.1)
|  Error:
|  incompatible types: possible lossy conversion from double to int
|  int value = Math.abs(-2.1);
|              ^------------^
```

Es momento de ver otras clases y sus métodos para realizar tareas como la conversión de tipos de datos.

CONVERSIÓN DE TIPOS

Dentro de los lenguajes de programación es posible transformar entre diferentes tipos de datos, es decir a un entero lo puedo transformar en real o a una cadena de texto en número o viceversa. No es recomendable transformar un real a un entero, ya que se pierde precisión, aconsejo utilizar el método *round* de la clase *Math* que ayuda en estos casos.

(<Tipo de dato destino>)<valor>

(double)100

El primer mecanismo de transformación entre valores numéricos fue el casting que se resume así:

En el caso anterior, el resultado sería 100.0, es decir que cambió de entero a real. Para transformar de real a entero vea los casos que se muestran en la siguiente porción de código:

```
jshell> Math.round(100.0)
$18 ==> 100

jshell> Math.round(301.5)
$19 ==> 302

jshell> Math.round(301.3)
$20 ==> 301

jshell> (double)89
$25 ==> 89.0
```

```
jshell> var value = 1265.987
value ==> 1265.987

jshell> String.valueOf(value)
$22 ==> "1265.987"

jshell> String.valueOf(123)
$23 ==> "123"

jshell> String.valueOf(87.87612)
$24 ==> "87.87612"
```

Convertir un número a cadena de texto se puede hacer con el método *valueOf* de la clase *String*, como se muestra en la siguiente porción de código de la izquierda.

La conversión de cadenas de texto (String) a datos numéricos (entero o real) se cubrirá más adelante en el apartado de cadenas de texto.