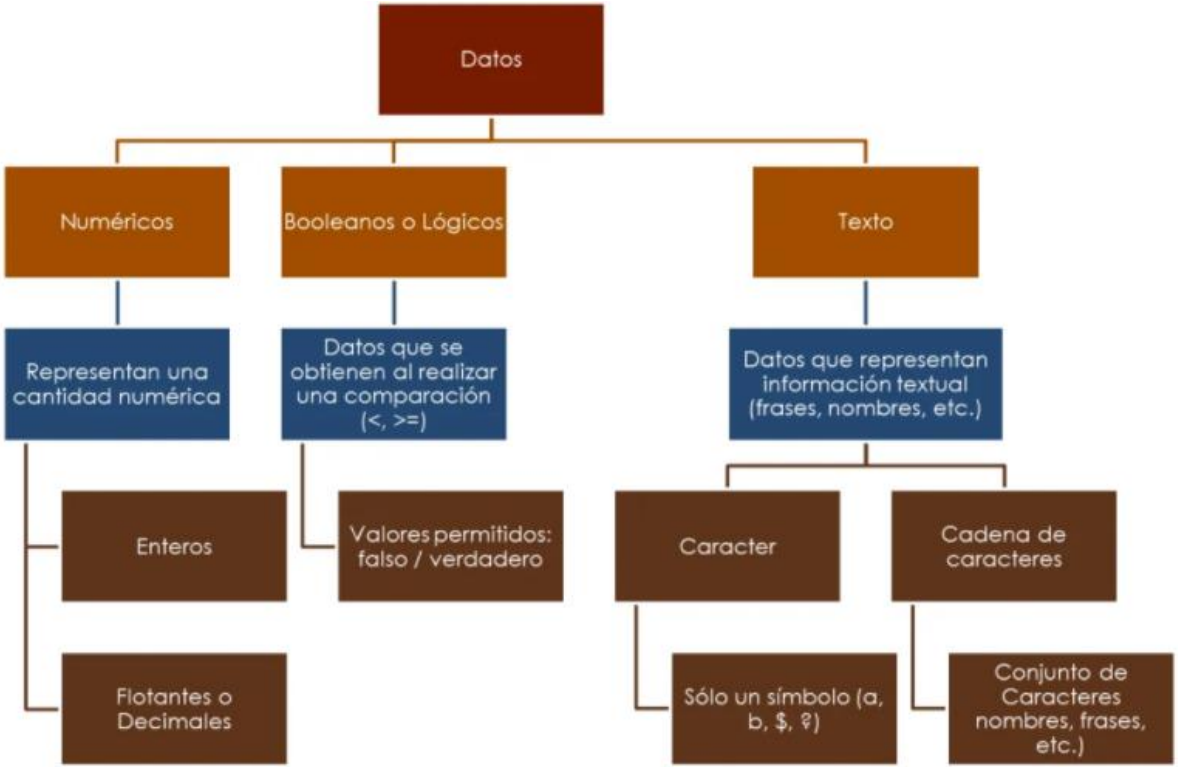
PROGRAMACION MODULAR

TIPOS DE DATOS USADOS EN PROGRAMACIÓN



TIPOS DE DATOS USADOS EN PROGRAMACIÓN

Tipos de datos primitivos de Java.

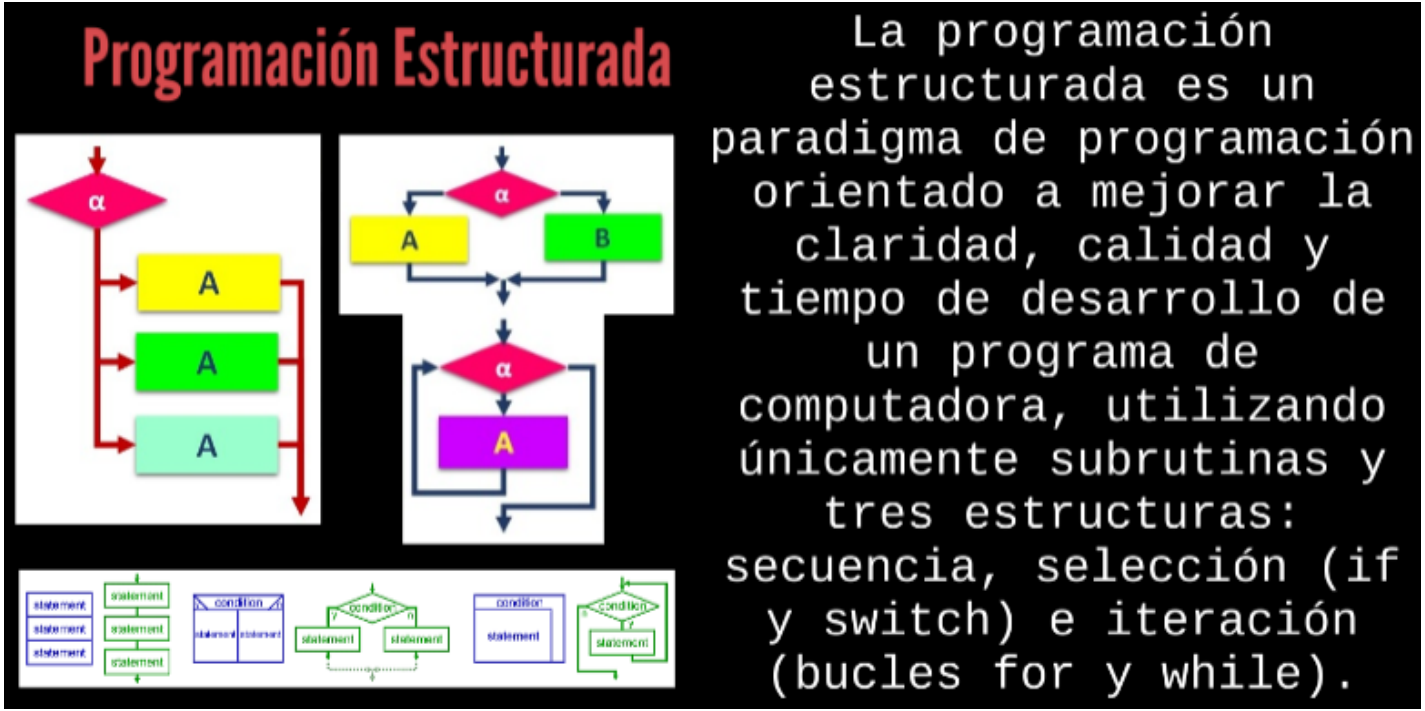
El lenguaje de programación Java es un lenguaje fuertemente ***tipado.*** Esto significa que a todos los datos se les debe definir un tipo antes utilizarlos. Dado que Java es un lenguaje de programación orientado a objetos, utiliza el término "Tipo de datos" para referirse tanto a variables, constantes u objetos, en general a cualquier "cosa" que utilice un espacio en memoria. Además, debido a que Java permite que el programador diseñe sus propias clases, éstas también pueden convertirse en tipos de datos. Es por ello que Java distingue a los tipos de datos "originales", predefinidos por el lenguaje y los denomina ***tipos de datos primitivos***.

Se puede decir que estos tipos de datos son la base para definir a tipos de datos compuestos, por eso se les da el nombre de primitivos, porque de ellos parten los demás.

TIPOS DE DATOS USADOS EN PROGRAMACIÓN

Tipo	Palabra reservada	Tamaño en bits	Rango de valores
Booleano o lógico	boolean	---	true/false
Caracter	char	16	0 a 65535 (o '\u0000' a '\Uffff')
Entero de 8 bits	byte	8	-128 a +127 (o de -2^7 a $2^7 - 1$)
Entero corto	short	16	-32,768 a +32,767 (-2^{15} a $2^{15} - 1$)
Entero	int	32	-2,147,483,648 a +2,147,483,647 (-231 a 231 - 1)
Entero largo	long	64	-9,223,372,036,854,775,808 a +9,223,372,036,854,775,807 (-263 a 263 - 1)
Punto flotante	float	32	Rango negativo: -3.4028234663852886E+38 a -1.4012984643248170 7e-45 Rango positivo: 1.4012984643248170 7e-45 a 3.4028234663852886E+38
Punto flotante de doble precisión	double	64	Rango negativo: -1.7976931348623157E+308 a -4.94065645841246544e-324 Rango positivo: 4.94065645841246544e-324 a 1.7976931348623157E+308

PROGRAMACIÓN ESTRUCTURADA



La programación estructurada es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora, utilizando únicamente subrutinas y tres estructuras: secuencia, selección (if y switch) e iteración (bucles for y while).

PROGRAMACIÓN MODULAR

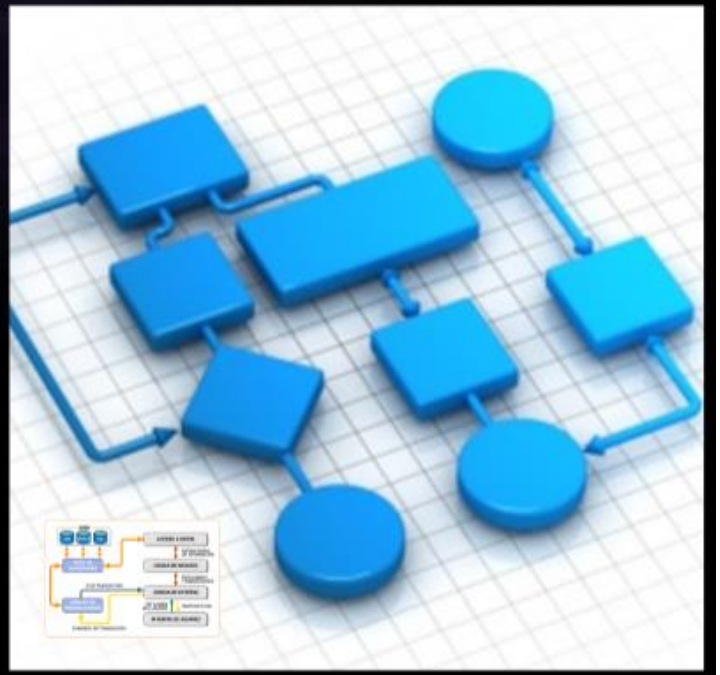
2. Programación Modular

La programación modular es un paradigma de programación que consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más legible y manejable.

Al aplicar la programación modular, un problema complejo debe ser dividido en varios subproblemas más simples y estos a su vez en otros subproblemas aún más simples. Esto debe hacerse hasta obtener subproblemas lo suficientemente simples como para poder ser resueltos fácilmente con algún lenguaje de programación. Esta técnica se llama "refinamiento sucesivo".

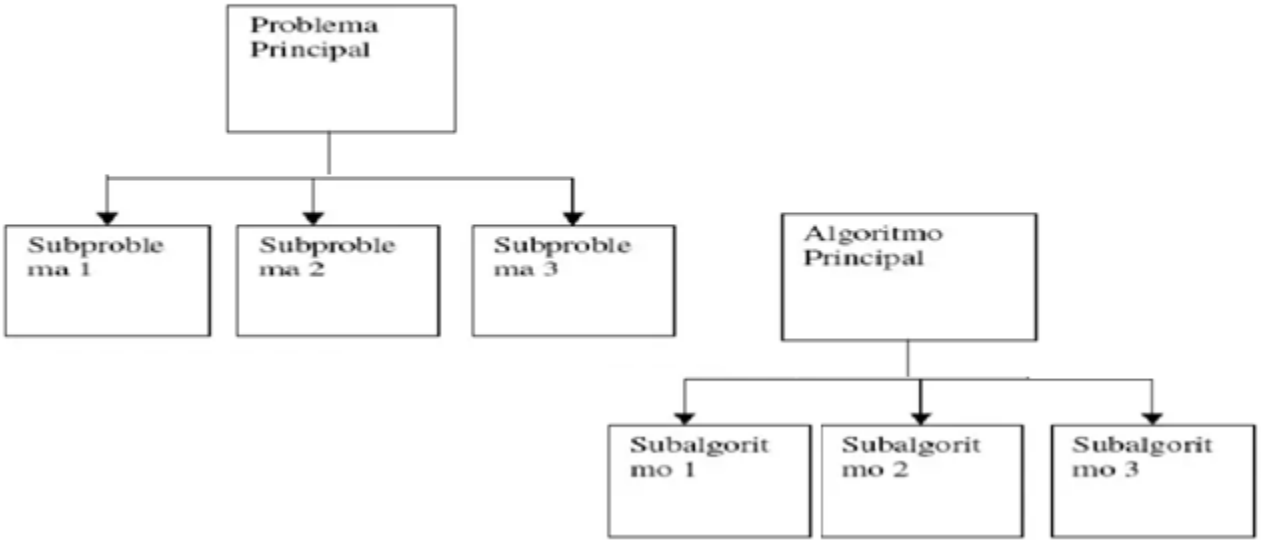
PROGRAMACIÓN MODULAR

Al aplicar la programación modular, un problema complejo debe ser dividido en varios subproblemas más simples y estos a su vez en otros subproblemas aún más simples. Esto debe hacerse hasta obtener subproblemas lo suficientemente simples como para poder ser resueltos fácilmente con algún lenguaje de programación. Ésta técnica se llama "refinamiento sucesivo".



PROGRAMACIÓN MODULAR

Esta técnica se llama “divide y vencerás” .



PROGRAMACIÓN MODULAR

Ventajas de la programación modular.

Las ventajas de la programación modular son:

- Reducir la complejidad del problema
- Reducir el tamaño del problema
- Favorecer el entendimiento del problema
- Facilitar la cooperación entre programadores
- Reutilizar código.
- Facilitan la lectura del código.
- Ayuda a ser más clara la lógica del programa
- Permite plantear una solución completa del problema, para luego profundizar en los detalles.
- La depuración es más fácil de realizar ya que primero se corrigen errores en los módulos de nivel inferior.

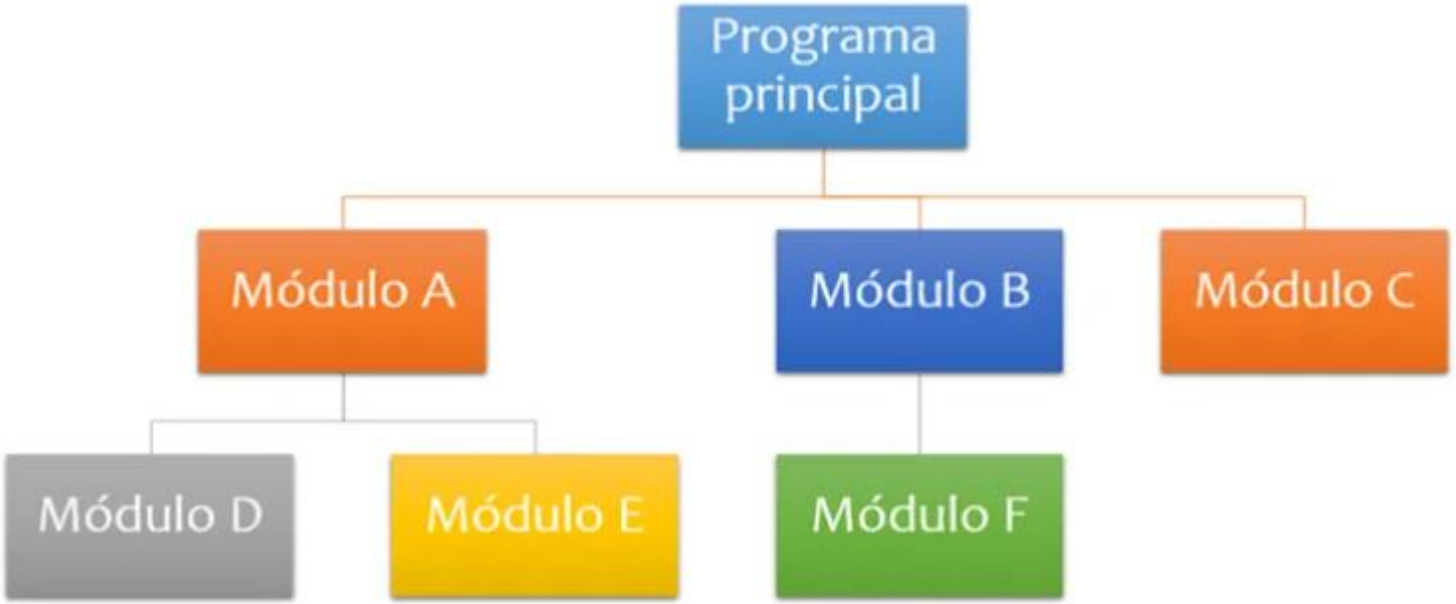
PROGRAMACIÓN MODULAR

¿Cuáles son los componentes de un programa modular?

Un programa que ha sido diseñado bajo este paradigma contiene:

- Un módulo principal, encargado de coordinar la ejecución de los demás módulos.
- Una serie de módulos que resolverán cada de una las tareas concretas del problema

PROGRAMACIÓN MODULAR



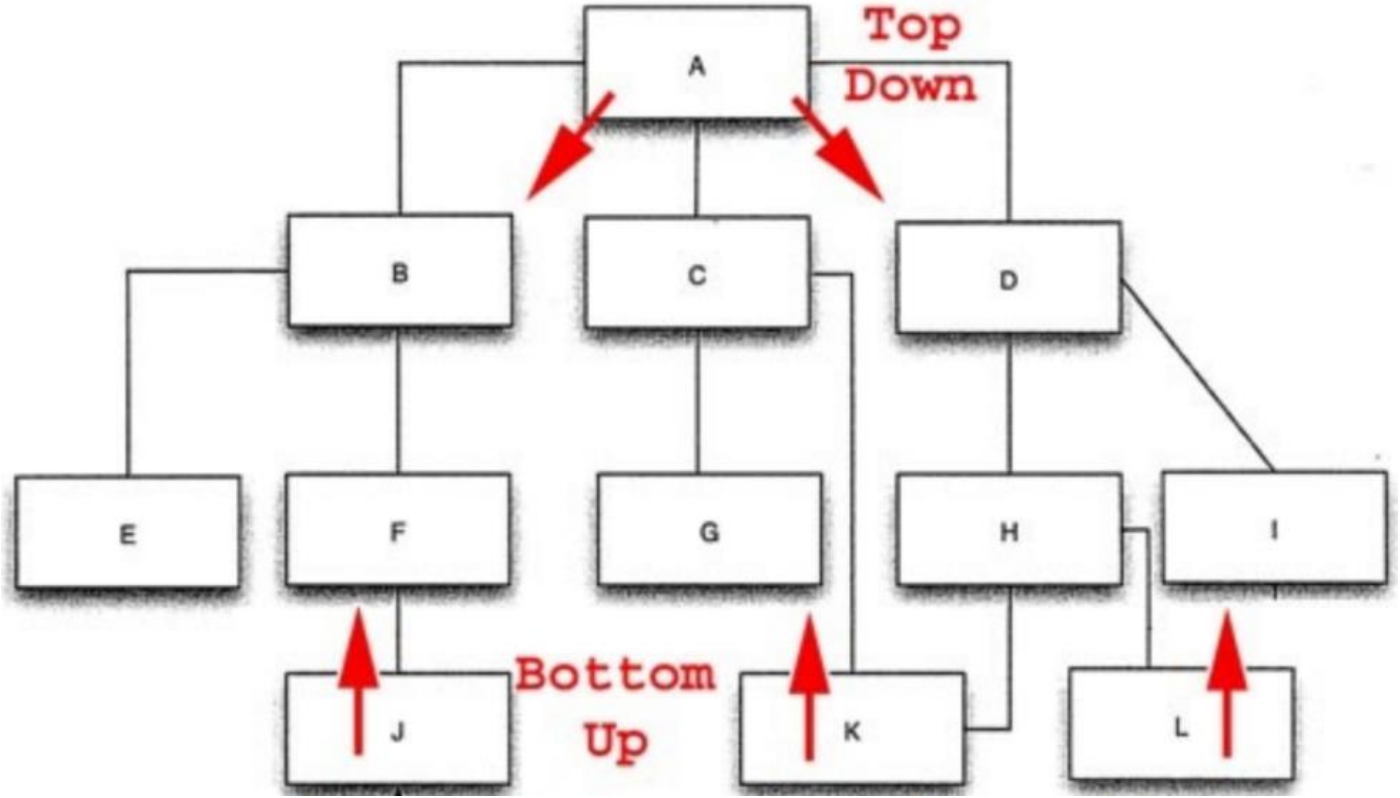
PROGRAMACIÓN MODULAR

Paradigmas de la programación modular.

En ese punto surgen dos estrategias derivadas y aplicables a la programación modular:

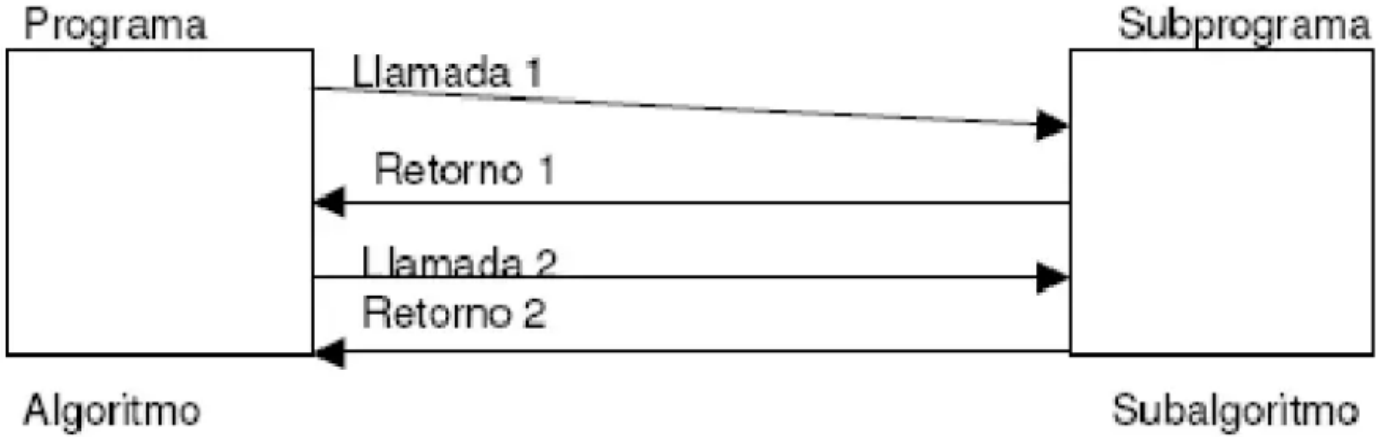
- Top-down
  - El enfoque top-down o “de arriba hacia abajo” es la descomposición del programa en sub-programas, especificando sin entrar en detalles, los módulos inferiores. También se le llama programación descendente, porque los sub-programas o módulos van surgiendo “hacia abajo”, es decir, conforme bajamos de nivel se ve la necesidad de generar más módulos.
- Bottom-up:
  - La estrategia bottom-up o “de abajo hacia arriba” considera que primero se deben programar los módulos del nivel más bajo. Y después van surgiendo los módulos de niveles superiores. Implicar enfocarse en los detalles primero, y después en lo general.

PROGRAMACIÓN MODULAR



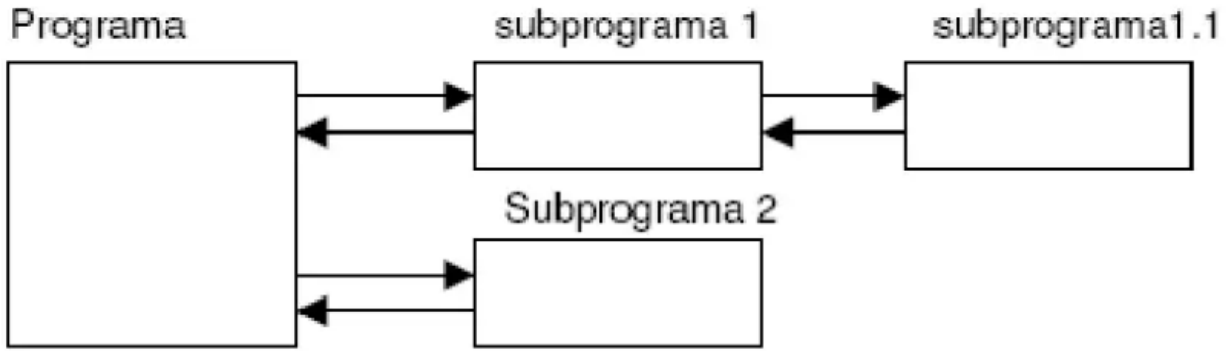
PROGRAMACIÓN MODULAR

- Se dice que el programa principal invoca al subprograma, el subprograma ejecuta la tarea y luego devuelve el control al programa.



PROGRAMACIÓN MODULAR

Un programa puede llamar a su ves a sus propios subprogramas



MODULOS PREDEFINIDOS DE JAVA

Métodos de la clase Math de Java

Esta clase ya viene incluida en nuevas versiones de Java, por lo que no habrá que importar ningún paquete para ello.

Para utilizar esta clase, debemos escribir **Math.método(parámetros);** donde método sera uno de los siguientes y parámetros aquellos que tengamos que usar. Un método puede estar sobrescrito para distintos tipos de datos.

Recuerda que si almacenas el resultado de la función, debe coincidir con el tipo de la variable.

CONSTANTE	DESCRIPCIÓN
PI	Devuelve el valor de PI. Es un double.
E	Devuelve el valor de E. Es un double.

MÉTODO	DESCRIPCIÓN	PARÁMETROS	TIPO DE DATO DEVUELTO
abs	Devuelve el valor absoluto de un numero.	Un parametro que puede ser un int, double, float o long	El mismo que introduces.
arcos	Devuelve el arco coseno de un angulo en radianes.	Double	Double
asin	Devuelve el arco seno de un ángulo en radianes.	Double	Double
atan	Devuelve el arco tangente entre -PI/2 y PI/2.	Double	Double
atan2	Devuelve el arco tangente entre -PI y PI.	Double	Double
ceil	Devuelve el entero más cercano por arriba.	Double	Double
floor	Devuelve el entero más cercano por debajo.	Double	Double
round	Devuelve el entero más cercano.	Double o float	long (si introduces un double) o int (si introduces un float)
cos	Devuelve el coseno de un ángulo.	Double	Double
sin	Devuelve el seno de un ángulo.	Double	Double
tan	Devuelve la tangente de un ángulo.	Double	Double
exp	Devuelve el exponencial de un número.	Double	Double
log	Devuelve el logaritmo natural en base e de un número.	Double	Double
max	Devuelve el mayor de dos entre dos valores.	Dos parametros que pueden ser dos int, double, float o long	El mismo tipo que introduces.
min	Devuelve el menor de dos entre dos valores.	Dos parametros que pueden ser dos int, double, float o long	El mismo tipo que introduces.
random	Devuelve un número aleatorio entre 0 y 1. Se pueden cambiar el rango de generación.	Ninguno	Double
sqlrt	Devuelve la raíz cuadrada de un número.	Double	Double
pow	Devuelve un número elevado a un exponente.	Dos parámetros double (base y exponente)	Double

```java
public class PruebaApp {
    public static void main(String[] args) {

        double operador1=25.5;
        double operador2=15.21;

        System.out.println(Math.ceil(operador1)); // Devuelve 26.0
        System.out.println(Math.floor(operador2)); //Devuelve 15.0
        System.out.println(Math.pow(operador1, operador2)); // Devuelve 2.4744355379753 61E21
        System.out.println(Math.max(operador1, operador2)); //Devuelve 25.5
        System.out.println(Math.sqrt(operador1)); ////Devuelve 5.049752469181039
    }
}
```