Recibe una cálida:

Bienvenida!

Te estábamos esperando 😁







El Entorno Java para la programación

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0





HOJA DE RUTA

¿Cuáles skill conforman el programa?









REPASO CLASE ANTERIOR



En la clase anterior trabajamos 📚:

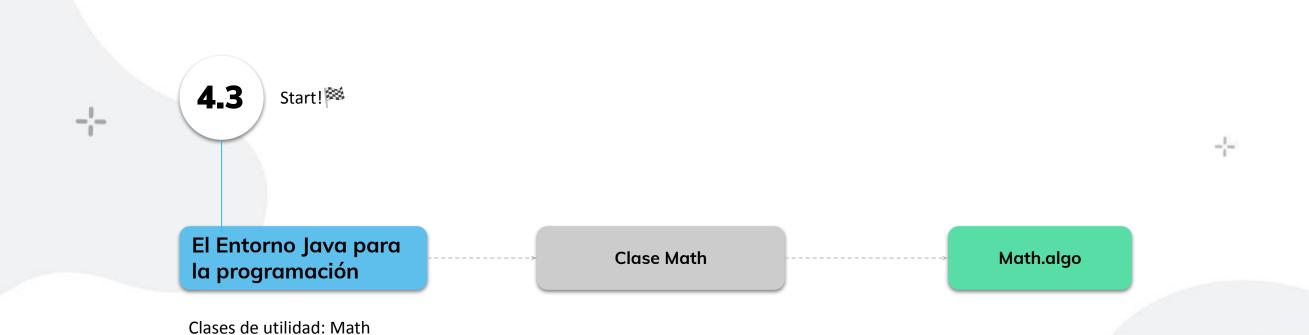
 Comprender la importancia e implementación de la clase String







LEARNING PATHWAY









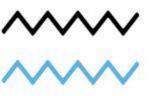
OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?





Conocer y manipular las funciones asociadas a la clase Math







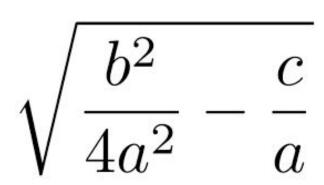
Rompehielo

Calculando: 🙌

Las operaciones matemáticas son moneda corriente en el mundo de la programación. En esta clase veremos diferentes maneras de interactuar con ellas utilizando una clase de Java exclusiva para esta tarea. Veamos la imagen de la derecha:



- 1- ¿Cuántas operaciones vemos en la imagen?
- 2- ¿Cuáles reconociste?
- 3- En programación, ¿hay alguna forma de representar estas operaciones de manera simple?







> Clase de utilidad: Math





Clase Math

¿Qué es la clase Math?

Con la clase Math puedes realizar **operaciones aritméticas** básicas, como sumar, restar, multiplicar y dividir números. También puedes utilizarla para realizar cálculos más complejos, como redondear números, calcular raíces cuadradas, generar números aleatorios y mucho más.

Además de las operaciones matemáticas, la clase Math también contiene **constantes** matemáticas útiles, como **PI** (π) y E (número de Euler).

Para invocar los siguientes métodos, simplemente debes colocar la palabra **Math.** y a continuación el nombre del método que quieres utilizar. Veamos algunos de los métodos más utilizados.





Math: redondeo y truncamiento

Métodos ceil(), floor() y round():

La Clase Math en Java proporciona varios métodos para redondear y truncar números decimales. Estos métodos son útiles cuando necesitamos ajustar los valores numéricos a ciertos límites o para realizar cálculos que requieren valores enteros.

- ---
- Math.ceil(): Redondea hacia arriba, es decir, devuelve el entero más pequeño que es mayor o igual al número dado.
- Math.floor(): Redondea hacia abajo, es decir, devuelve el entero más grande que es menor o igual al número dado.
- Math.round(): Redondea el valor de un número real al entero más cercano.
- --- Veamos el comportamiento en código:





×

Math.ceil()

```
double numero = 5.3;

double resultado = Math.ceil(numero);

System.out.println("Redondeando hacia arriba el número "

+ numero + ": " + resultado);

Console ×

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32

Redondeando hacia arriba el número 5.3: 6.0
```







>

Math.floor()







>

Math.round()

```
double numero = 5.5;

double resultado = Math.round(numero);

System.out.println("Redondeando al entero más cercano el número "

+ numero + ": " + resultado);

Console ×

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v.

Redondeando al entero más cercano el número 5.5: 6.0
```







Math: potenciación y radicación

Métodos pow(), sqrt() y cbrt():

La Clase Math también proporciona funciones para realizar operaciones de potenciación y radicación. Estos métodos son útiles cuando necesitamos elevar números a potencias específicas o calcular raíces de diferentes órdenes.



- Math.pow(): Eleva un número a una potencia determinada.
- Math.sqrt(): Calcula la raíz cuadrada de un número.
- Math.cbrt(): Calcula la raíz cúbica de un número.

Veamos el comportamiento en código:







>

Math.pow()

```
double base = 5;

double exponente = 3;

double resultado = Math.pow(base, exponente);

System.out.println("El número " + base +

" elevado a la potencia " + exponente +

" es: " + resultado);

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

El número 5.0 elevado a la potencia 3.0 es: 125.0
```







__ Math.sqrt()

```
×
```







Math.cbrt()

```
×
```

0

```
double numero = 27;
double resultado = Math.cbrt(numero);

System.out.println("La raíz cúbica del número " + numero + " es: " + resultado);

Console ×

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe

La raíz cúbica del número 27.0 es: 3.0
```







Math: generación de números aleatorios

Método random():

La Clase Math en Java incluye métodos para generar números aleatorios. Estos números pueden ser útiles en una variedad de aplicaciones, como juegos, simulaciones y algoritmos de optimización.

• **Math.random():** Genera un número aleatorio de tipo double entre 0 (inclusive) y 1 (exclusivo).

Veamos el comportamiento en código:







3

Math.random()

```
double aleatorio = Math.random();

System.out.println("Número aleatorio entre 0 y 1: " + aleatorio);

Console ×

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.

Número aleatorio entre 0 y 1: 0.99407537320504
```







Math.random()

Para generar un número entero entre 0 y 9, Hay que escribir la siguiente sentencia:

int aleatorio = (int) (Math.random () * 10);

De esta manera al multiplicar el valor por 10, el rango de valores posibles se convierte en: 0.0 <= numero < 10.0

```
int aleatorio = (int) (Math.random () * 10);

System.out.println("Número aleatorio entre 0 y 9: " + aleatorio);

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-Número aleatorio entre 0 y 9: 6
```





Math: comparación y selección

Métodoa min(), max() y abs():

La Clase Math en Java también proporciona funciones para comparar y seleccionar valores numéricos. Estos métodos son útiles para determinar el mínimo, el máximo o el valor absoluto de los números, así como para realizar otras operaciones de comparación y selección.

- Math.min(): Retorna el menor de dos valores.
- Math.max(): Retorna el mayor de dos valores.
- Math.abs(): Retorna el valor absoluto de un número.

Veamos su aplicación en código:





>

Math.min()







>

Math.max()







><

Math.abs()

```
int num = -10;
int absoluto = Math.abs(num);

System.out.println("El valor absoluto de " + num + " es: " + absoluto);

Console X

Console X

</pre
```







LIVE CODING

Ejemplo en vivo

Math.¿qué?:

Vamos a poner en práctica lo aprendido resolviendo en vivo una operación aritmética bastante frecuente:

1. Desarrollar un programa que tome como dato de entrada un número que corresponde a la longitud del radio una esfera y que calcule y escriba el volumen de la esfera que se corresponde con dicho radio.

La fórmula para calcular el volumen de la esfera es $\mathbf{v} = (4/3) * PI * r^3$

Tiempo: 25 minutos







Ejercicio N° 1 Math.algo







Contexto: 🙌

Como hemos visto, existen muchas formas de manipular números y operaciones aritméticas a través de la clase Math de Java. Vamos a practicar algunos de los métodos aprendidos:

Consigna: 🚣

Escribir un programa que lea un número entero ingresado por teclado y luego muestre por pantalla:

- el doble
- el triple
- la raíz cuadrada
- la potencia elevada a la 7 de ese número.

Tiempo : 25 minutos





¿Alguna consulta?



RESUMEN

¿Qué logramos en esta clase?



Comprender la importancia e implementación de la clase de utilidad Math y sus métodos asociados.







#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 👇 👇

-!-

- 1. Repasar nuevamente la grabación de esta clase
- 2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. Material 1 (Foro)
 - b. Lectura Módulo 4, Lección 3: páginas 15 17
- 3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.





-1-



Muchas Gracias!

Nos vemos en la próxima clase 🤎



M alkemy

>:

Momento:

Time-out!

⊘5 min.



