

Ejercicio 1

Realizar un programa que simule una calculadora.

La estructura del programa debe contener dos clases:

- una clase que tenga el método especial **main** para poder ejecutar lo que se pide en los requerimientos,
- y otra clase llamada **Calculadora**

Requerimientos:

Se necesita realizar un programa que pida por consola dos valores enteros, la operación matemática a realizar y muestre por pantalla el resultado. Por ejemplo:

→ Primer valor: 10

→ Segundo valor: 30

→ Operación: +

→ Resultado: 40

Nota: para obtener el resultado es necesario usar la clase Calculadora

Si la operación ingresada no es válida (es decir **no** es +, -, *, /) se debe mostrar: “Operación no válida”

Una vez obtenido el resultado, se debe evaluar lo siguiente:

- Si el resultado de la operación es un número negativo, mostrar por pantalla el mensaje “Tené cuidado, el resultado es negativo!”.
- De lo contrario si el resultado es mayor o igual a 0 y menor a 10000 , mostrar por pantalla “Resultado dentro de los límites”.
- Y si no mostrar un mensaje que diga “Error, resultado muy grande”

No importa si la operación es válida o no, el programa **finaliza después de ejecutarse 5 veces**

Calculadora:

La clase Calculadora debe tener los métodos básicos para operar: **suma**, **resta**, **multiplicación** y **división**. Estos métodos deben **recibir** como parámetros los valores que se ingresaron por consola y **retornar** el resultado. Ejemplo:

suma(10, 30) -> retorna 40

No es necesario que Calculadora tenga atributos, pero si lo considera necesario puede agregarlos.

Nota: Recordar que en el caso de la división no está definida la división por cero. Manejar este caso de manera adecuada según su criterio.

Ejercicio 2

Se necesita modelar los personajes de un juego y para ello se dispone de la siguiente información:

- **Personaje:** tiene el atributo **salud** (tipo entero) y un método abstracto **entrenar()**
- **Espadachín:** este personaje extiende de Personaje y además tiene el atributo **ataque** (tipo entero). Cuando el espadachín entrena, incrementa su ataque en 10.
- **Torre:** extiende de Personaje y tiene el atributo **nivel** (tipo entero). Cuando entrena, aumenta su nivel en 1
- **Mago:** extiende de Personaje y tiene el atributo **magia** (tipo String). Los magos son vagos, no les gusta entrenar. Por lo tanto cuando entrenan, no sucede nada.

Todas las clases deberán contar con sus constructores correspondientes y los métodos getters & setters.

Nota: no es necesario crear ninguna clase que contenga main para instanciar estas clases pero es recomendable hacerlo para probar que funciona correctamente lo que desarrollaron.

Ejercicio 3

Subir todo lo desarrollado en un repositorio privado en Github, agregar al docente como colaborador del proyecto (Usuario en github: ProfeMMQA) y enviar el link del repo por privado en Slack.