

RELACION DE EJERCICIOS 6. PROGRAMACIÓN ORIENTADA A OBJETOS

1. Cree una clase `Rectángulo` con los atributos `longitud` y `ancho`, cada uno con un valor predeterminado igual a 1. Proporcione las funciones `set` y `get` para los atributos `longitud` y `ancho` respectivamente. Las funciones `set` deben verificar que `longitud` y `ancho` contengan números reales mayores que cero y menores que 20. Además, proporcione métodos que calculen el perímetro y el área del rectángulo.

Escriba un método `main` que solicite los datos de un rectángulo y muestre cual es su área y su perímetro. Probar a introducir un dato incorrecto (mayor o igual que 20).

2. Realizar un programa que permita gestionar el saldo de una cuenta corriente. Una vez introducido el saldo inicial, se mostrará un menú que permitirá efectuar las siguientes operaciones:
 1. Hacer un reintegro, se pedirá la cantidad a retirar
 2. Hacer un ingreso, se pedirá la cantidad a ingresar.
 3. Consultar el saldo y el número de reintegros e ingresos realizados.
 4. Finalizar las operaciones. Debe confirmar si realmente desea salir e informar del saldo al final de todas las operaciones.

Debe realizarse una clase `Cuenta` y la clase `Principal`

3. Realizar una programa en Java que sirva para simular una máquina de café.

La máquina tiene varios depósitos:

- Un depósito café con capacidad para 50 dosis.
- Un depósito de leche con capacidad para 50 dosis.
- Un depósito de vasos con capacidad para 80 vasos.

También dispone de monedero con una cantidad de euros. Cuando se cree una nueva máquina se hará con un monedero inicial y con los depósitos llenos. El café cuesta 1 euro, la leche 0.8 euros y el café con leche 1.5 euros. Al servir cada bebida deben descontarse una dosis de los depósito/s correspondientes, un vaso y añadir al monedero lo recaudado.

Deben implementarse un menú con 5 opciones:

1. Servir café solo (1 euro).
2. Servir leche (0.8 euros)
3. Servir café con leche (1,5 euros)
4. Consultar estado máquina. Aparecen los datos de los depósitos y del monedero.
5. Apagar máquina y salir

Cuando se sirva una bebida habrá que solicitar el dinero que va a introducir el usuario. Si introduce una cantidad inferior al precio o no hay existencias debe informarse con el error correspondiente. También debe producirse un error si la máquina no tiene cambio de esa cantidad. Si puede dar cambio informará con el mensaje "Recoge tu cambio de XX euros"

La clase máquina debe disponer al menos de los siguientes métodos

- Método para llenar depósitos. No pueden llenarse los depósitos por separado.
- Método para vaciar monedero.
- Método para consultar el estado de la máquina
- Método/s para servir café sólo, leche, o café con leche. Este método mostrará los mensajes de error correspondientes o el mensaje "Producto servido".

4. Implemente una clase llamada `Complejo` para realizar aritmética con números complejos sabiendo que los números complejos tienen la forma:
 $\text{parteReal} + \text{parteImaginaria} * i$

La clase deberá proporcionar un constructor que permita inicializar un objeto cuando se declare. También deberá proporcionar funciones miembro de tipo *public* para cada una de las siguientes operaciones:

- Suma de dos números Complejos: las partes reales se suman y las partes imaginarias se suman.
- Resta de dos números Complejos: la parte real de operando derecho se resta a la parte real del operando izquierdo, y la parte imaginaria del operando derecho se resta a la parte imaginaria del operando izquierdo.
- Impresión de números Complejos de la forma (a, b), en donde a es la parte real y b es la parte imaginaria.

NOTA: Utilice variables *double* para representar datos miembros de la clase.

Escriba un método `main` que cree dos números complejos (solicitando parte real e imaginaria) y muestre cual es el número complejo resultado de la suma y de la resta

5. Utilizando la clase `Complejo` creada en el ejercicio 1 para realizar un programa que muestre un menú de este tipo:
1. Sumar complejos: Debe solicitar dos números complejos y mostrar el resultado de la suma
 2. Restar complejos: Debe solicitar dos número complejos y mostrar el resultado de la resta
 3. Salir
6. Crea una clase `Racional` que permita trabajar con números racionales (fracciones). Incluye los siguientes métodos: `suma`, `resta`, `multiplicación`, `división`, `copia()`, `simplificar()` y la redefinición de `equals` y `toString`.