

Tarea 1 – Programacion1

1. Gestion de Cuentas Bancarias

Preguntas:

- **¿Qué atributos debería tener la clase?**
 - numeroCuenta: Identifica la cuenta
 - saldo: Almacena el dinero disponible
- **¿Cómo se asegura que el saldo no se vuelva negativo?**
 - En el constructor : si el saldo inicial es negativo, se asigna 0
 - En retirar(): solo permite retiros si hay suficientes saldo

2. Control de Temperatura

Preguntas:

- **¿Cómo se aplica la encapsulación en este problema?**
 - temperaturaCelsius solo es accesible mediante métodos.
 - Métodos get y set: Controlan el acceso y modificación de la temperatura.
- **¿Cómo se realiza la conversión de temperatura en los métodos?**
 - **Fahrenheit:** $(\text{temperaturaCelsius} * 9/5) + 32$
 - **Kelvin:** $\text{temperaturaCelsius} + 273.15$

3. Registro de Productos en un Inventario

Preguntas:

- **¿Cómo se encapsula la información del producto?**
 - Los atributos código, nombre y precio son privados.
 - Se acceden y modifican mediante getters y setters, asegurando control sobre los datos.
- **¿Por qué es importante validar los valores ingresados?**
 - Evita precios negativos con setPrecio(), garantizando datos coherentes.
 - Protege la integridad del producto y evita errores en cálculos.

4. Temporizador con Alarma

Preguntas:

- **¿Cómo interactúan las clases Temporizador y Alarma?**
 - Temporizador tiene un atributo alarma, configurado con setAlarma().
 - iniciar(int duracion) cuenta los segundos.
 - Compara tiempoActual con tiempoObjetivo y llama activar() si coinciden.
- **¿Cómo se asegura que los atributos sean accesibles solo mediante métodos específicos?**
 - tiempoObjetivo, tiempoActual y alarma son privados.
 - getTiempoObjetivo() consulta, setAlarma() asigna sin exponer datos.
 - Solo iniciar() modifica el tiempo, y activar() se ejecuta cuando corresponde.

5. Control de Notas de un Curso

Preguntas:

- **¿Cómo se encapsulan los datos de los estudiantes dentro del curso?**
 - Atributos privados: nombre, carnet, y notaFinal en Estudiante, y estudiantes en curso.
 - Métodos de acceso: getNombre(), getCarnet(), getNotaFinal() para consultar datos.
 - Control de lista: agregarEstudiante() evita modificaciones externas.
 - calcularPromedio() y mostrarEstudiantesAprobados() manejan la información sin exponer la lista.
- **¿Cómo se garantiza que las notas sean válidas?**
 - Se logra mediante el método setNotaFinal() dentro de la clase Estudiante.
 - Si la nota esta entre 0 y 100.
 - Si esta fuera de rango se muestra un error.