

## Unidad 3 Programación Orientada a Objetos 1

### Boletín de ejercicios 1

**NOTA:** Todas las operaciones se realizarán mediante métodos y todos los ejercicios tendrán una clase principal donde se comprueben todas las acciones que haga el programa. Nunca leeremos datos por teclado dentro de los métodos, si hace falta se les pasarán como parámetro.

Además, estos ejercicios se harán sin usar arrays de objetos que se verán más adelante en otro boletín.

### CON UNA SOLA CLASE (MÁS PRINCIPAL Y/O LEER)

1. Implementar una clase que modele la cabecera de los exámenes de un centro escolar. La información sobre cada examen será el nombre de la asignatura, aula donde se realiza y un String con la fecha. Solo tendrá un método para imprimir adecuadamente dicha cabecera.
2. Hacer un programa que calcule el área de un círculo en  $\text{cm}^2$ . Debemos tener la clase Círculo, con un solo atributo llamado radio. En el main, debemos crear un objeto tipo Círculo y llamar a dicho método. Mejorar el programa añadiendo a la clase Círculo otro método que calcule el área del círculo en  $\text{m}^2$ .
3. Implementar un programa que contenga dos métodos en una clase “operaciones”: un método al que se le pase un número entero y diga si este es positivo o negativo y otro que diga si un número es par o impar. La clase no debe tener ningún atributo. Se deben probar los métodos en el main.
4. Realizar un programa que calcule el volumen de un cilindro. Vosotros debéis decidir qué método usar y qué valores pasarle. Se debe tener la clase Cilindro.
5. Cuando era pequeño me mandaban a copiar determinadas frases cuando hacía algo mal para no olvidarlo. Realizar un programa que lea un texto del teclado y un número entero y muestre en pantalla dicho texto una cantidad de veces igual al número introducido. La clase se puede llamar "Copiado".
6. Crear una clase llamada Generadora sin atributos, pero con varios métodos que generen aleatoriamente distintos números o resultados según las características del sorteo o para qué se vaya a usar, por ejemplo, generar 1, x, 2 para una quiniela, generar par o impar para jugar a pares o nones, generar 1, 2 o 3 para jugar a los "chinos", del 1 al 49 para el sorteo de primitiva... Crear una clase principal donde se prueben todos los métodos pidiendo los datos necesarios para cada generación.
7. Crear un programa que simule el juego de cara o cruz. Para ello, se usarán los siguientes métodos:
  - Método que simule el lanzamiento de la moneda y resultado.
  - Método para comprobar ganador, se le pasará el número elegido por el jugador y dentro se generará el aleatorio llamando al método anterior. Devuelve true si son iguales o false si no.
  - Un método para mostrar en bonito el ganador. Si se le pasa true mostrar un mensaje de ganador y si no, debe mostrar "Siga jugando".
  - En el main, se debe crear lo necesario para simular un juego de cara o cruz.
8. Se desea llevar un control del estado de una cuenta corriente. Una cuenta corriente está caracterizada por su saldo y el nombre del titular y sobre ella se pueden realizar tres tipos de operaciones:
  - Ingresar (cantidad): ingresa en la cuenta una cantidad de dinero dada por el usuario. No imprime de nuevo el saldo, pero sí lo actualiza.

- Retirar (cantidad): Debe comprobar si queda suficiente saldo y si es así, disminuye la cuenta una determinada cantidad de dinero especificada por el usuario. No imprime de nuevo el saldo, pero sí lo actualiza.
- Calcular el saldo en dólares americanos. Se le debe pasar el valor de cambio y debe devolver el valor del saldo en dólares.
- Consultar el saldo. Solo imprime el saldo.

Supón que la cuenta inicialmente tiene un saldo de cero. Escribe una clase CuentaCorriente que implemente las funcionalidades descritas y un main con un menú para probarlas.

9. Crear un programa que gestione un sorteo de lotería. La lotería de Navidad se realiza sobre números que van desde el 00001 al 99999 (no tendremos en cuenta series). Para facilitar, tendremos solo la clase Sorteo y la principal para probar todo. El programa debe hacer lo siguiente:

- Comprar un décimo (eligiendo el número por teclado).
- Comprar un décimo generado por la máquina (aleatorio).
- Hacer sorteo (generará aleatoriamente el número premiado).
- Comprobar si nuestro décimo está premiado devolviendo un boolean.
- Mostrar si eres ganador o no.

**Nota: si alguien quiere "entrenar" más, se pueden realizar todos los ejercicios hechos hasta ahora en los temas 1 y 2 mediante programación orientada a objetos.**

## CON MÁS DE UNA CLASE (MÁS PRINCIPAL Y/O LEER)

10. Este programa tendría otra serie de clases, pero por simplificar lo haremos solo con 3 clases. Crear un programa que simule una máquina vendedora de tickets de metro. Tendremos que crear la clase Ticket (Clase POJO), la clase Maquina (donde estarán todos los métodos con funcionalidad) y la clase principal de prueba. Tendrá un menú donde se pueda:

- Comprar uno o varios billetes.
- Calcular el cambio a devolver.
- Imprimir por pantalla el billete. Solo uno. Si se han comprado "x" debemos mostrar en el ticket el mensaje "válido para x personas".
- Opciones de operario: el operario que lleva el mantenimiento de la máquina, debe tener disponible un método que le ofrezca el saldo total de la recaudación de la máquina en ese día (no es necesario usar fechas) y otro método que pueda poner a cero el contador de saldo total.
- Dentro de la opción de operario, también debe contar con la posibilidad de cambiar el precio de los billetes ya que estos suelen subir todos los años.
- Las operaciones para el operario se harán solo si se introduce la contraseña adecuada antes de cualquier otra cosa, por lo que se debe tener el método comprobar contraseña.

11. Crear un programa para poner las notas de un alumno (clase Alumno con atributos). Los atributos de un alumno son su nombre, curso, un array de notas, número de suspensos y nota media.

En una clase GestionNotas, el programa debe poder poner notas a un solo alumno, mostrar todas sus notas por pantalla, modificar una nota, calcular la media y dar su número de suspensos.

Probar todo en la clase Principal.

### DE AMPLIACIÓN (no se entregan ni se corregirán)

Realizar el ejercicio de la máquina expendedora con el siguiente diagrama de clases, añadiendo aquellos métodos que consideremos adecuados para agregar funcionalidades:

