

## EJERCICIOS II: DESARROLLO DE CLASES

Desarrolla las siguientes clases usando el lenguaje de programación Orientado a Objetos Java:

- Modificar **Motocicleta**, **Alumno** y **Carta**
  - Crear **Mano**, **Partida**, **Cliente**, **Cuenta** y **Cajero**
- 

1. Modifica las clases **Motocicleta**, **Alumno** y **Carta** añadiéndoles los modificadores de protección y encapsulamiento de datos. Añade la clase alumno al **paquete personas**, la clase motocicleta al **paquete vehículos** y la clase carta al **paquete juegos**.
2. Dentro del paquete juegos implementa una nueva clase **Mano**. Esta clase representa a una mano de 3 cartas para simular un juego de cartas. Por tanto, como atributos tendrá un entero tamaño que indica el **número de cartas** y 3 **cartas**.

Además de definir los atributos indicados, habrá que implementar:

- Un constructor por defecto que inicialice el tamaño a 3 y cree las cartas por defecto.
  - Un método **getTamaño** que devuelva el tamaño de la mano.
  - Un método **mostrarCarta(int)** que reciba como argumento cuál de las tres cartas se quiere mostrar y que muestre la carta indicada por pantalla.
  - Un método **setCarta(int)** que reciba como argumento cuál de las tres cartas se quiere modificar y que pida el nuevo valor de la carta por teclado.
  - El método **toString()**
  - Un método **pedirMano()** que pida los valores necesarios para todas las cartas de la mano.
  - Un método **mostrarMano()** que muestre por pantalla todas las cartas de la mano.
  - Un método **manoGanadora()** que compare la mano que llama al método con la recibida como parámetro. Este método indicará por pantalla quien ha ganado si la primera o la segunda además de mostrar sus cartas. Para esto sumara los números de las cartas de cada mano ganando el mayor. En caso de empate se mirará quien tiene más cartas rojas (corazones y rombos) en caso de seguir empatando se dirá que se ha producido un empate y se mostraran ambas manos.
  - Crear una clase **Partida** dentro del paquete juego que cree dos manos por defecto y pida las cartas para jugar. Después comprobará que mano es la ganadora.
3. Se quiere crear una pequeña aplicación para gestionar un banco. Para esto se creará un paquete llamado **banco** y en este se crearán las clases **Cliente**, **Cuenta** y **Cajero**.
    - a) La clase Cuenta que tiene como atributos el **número de cuenta**, el **saldo** y la **contraseña**. Además de los siguientes métodos:
      - Un constructor por defecto y un constructor por parámetros que recibe el valor de todos los atributos como parámetro.
      - Los métodos getters que devuelven el valor de los atributos saldo y número de cuenta, que podrán ser accesibles sólo por las clases amigas.
      - El método **toString()**
      - Un método **modificar\_saldo(double)** que recibe el ~~saldo~~ la cantidad a ingresar o retirar como parámetro. Este método deberá estar protegido.
      - Un método **generar\_contraseña()** que devuelva un número aleatorio entre 1000 y 9999. Este método sólo es accesible por la clase Cuenta.

- Un método **modificar\_contraseña()** que pida la contraseña actual. Si es errónea mostrará un mensaje de error. Si es correcta generará una nueva contraseña mostrándola por pantalla y almacenándola en el atributo contraseña.
  - Un método **validar\_contraseña(int)** que devuelva un booleano indicando si la contraseña indicada como parámetro es la correcta o no.
  - Un método **mostrar\_datos(int)** que muestre el valor de los atributos y que reciba como argumento la contraseña. Sólo si la contraseña es correcta mostrará el valor de todos los atributos.
- b) La clase **Ciente** esta clase tiene como atributos el **dni** del cliente y su **suelo**, además contendrá un atributo de la clase **Cuenta**. Será necesario crear los siguientes métodos:
- Un constructor por defecto y un constructor por parámetros que reciba el dni y el saldo del cliente y **la cuenta**. En el constructor por parámetros la contraseña se inicializará al valor del dni.
  - Los métodos **get y set** de los atributos dni y sueldo. En los métodos set se comprobarán que los datos sean correctos
  - El método **toString()**
  - Un método **ingresar\_nómina()** que ingrese el sueldo del cliente en la cuenta.
  - Un método **sacar\_dinero(String, int)** que recibe como parámetro el dni y la contraseña de la cuenta del cliente, si esta es válida preguntará la cantidad de dinero a retirar y actualizarla el saldo. Si se produce algún error mostrará un mensaje de error.
- c) La clase **Cajero** sólo contendrá un método **main** para simular el funcionamiento del cajero. En primer lugar creará un cliente con los siguientes datos:

DNI: 987654

Suelo: 1200

Num Cuenta: 123456789

Saldo: 15000

Contraseña 987654

Al ejecutarlo mostrará un menú que se repita constantemente hasta pulsar salir, como el siguiente:

CAJERO AUTOMÁTICO

-----

1. Consultar/Modificar Datos personales
2. Ingresar Nomina
3. Consultar Datos Bancarios
4. Sacar Dinero
5. Modificar Contraseña
6. Salir

Para cada una de las opciones deberá realizar:

1. Mostrará el DNI y el sueldo del cliente y preguntará si quiere cambiarlos. Si se introduce s (si) pedirá por teclado los nuevos valores.
2. Ingresará el valor del sueldo en la cuenta (saldo).
3. Solicitando la contraseña mostrará por pantalla los datos bancarios (num\_cuenta, y saldo)
4. Solicitará la cantidad de dinero a retirar y la actualizará el saldo. Si la cantidad a retirar es superior al saldo mostrará un error
5. Permitirá modificar la contraseña generando una nueva y mostrándola.
6. Cierra la aplicación