

Ejercicios para entregar

En github crear un repositorio privado con la estructura accesoDatos/tema1/entregaminombre en el que subir las distintas clases solicitadas

Ejercicio 1

Construir una clase **Complejo** con dos atributos:

dPReal: parte real de tipo double

dPImag: parte imaginaria también de tipo double

y los constructores necesarios y tres métodos:

- *public void asignar(double x, double y)*
- *public Complejo sumar(Complejo c1, Complejo c2)*
- *public void imprimir*

Construir una clase **ComplejoTest** que contenga un método principal que trabaje con instancias de la clase Complejo y realice la prueba de los distintos métodos.

Incluye una imagen llamada **ComplejoTestPrueba.jpg** que sea un pantallazo de la ejecución.

Será necesario entregar el enlace a github

Ejercicio 2

En una tienda se venden 2 tipos de ordenadores: portátiles y de sobremesa. Ambos tipos de ordenadores se caracterizan por su código, RAM, CPU y por su precio. Además, los ordenadores portátiles tienen un atributo peso, y los de sobremesa la descripción del tipo de torre. Se pide:

- Definir e implementar los atributos y métodos de cada una de las clases, así como la relación entre ellos.
- Incluir un método getCaracterísticas() que devuelva las propiedades de cada uno de ellos y el método toString() para ver los valores
- Crear un programa en Java (**OrdenadorTestMiNombre**) que cree un par de objetos de tipo Ordenador y Portátil y realice operaciones con ellos.
- Incluye una imagen llamada **OrdenadorTestPruebaMiNombre.jpg** que sea un pantallazo de la ejecución.

Ejercicio 3

Queremos crear la estructura de un parking con programación orientada a objetos.

Crea el paquete parking para las clases de este ejercicio.

Escribe una **interfaz Parking** que incluya los métodos

- *hayPlaza()* que indique si hay sitio para aparcar
 - *aparcaCoche()* que estaciona un vehículo en el parking. Se pasa por parámetro el coche.
 - *sacaCoche()* que saca un vehículo del parking. Se pasa por parámetro el coche.
 - *getFactura()* que devuelve el importe a pagar por cada vehículo. Se pasa por parámetro el coche.
-

Escribe una clase **ParkingVehiculo** que implemente la interfaz Parking y tenga como contante el precio por minuto de 0,0425 euros y el máximo de plazas a 100 y como atributos dirección del parking, plazas actuales ocupadas.

Tendrá un constructor con la dirección del parking e implementará los métodos del interfaz.

Escribe una clase **Vehiculo** que tenga como atributos la matrícula y fecha inicio y fecha fin del estacionamiento.

Constructor con la matrícula y los getters y setters de los 3 atributos y toString()

Escribe una clase **ParkingTestMiNombre** que cree 5 vehículos y un parking y los aparque, saque del parking y calcule la factura.

Incluye una imagen llamada **ParkingTestPruebaMiNombre.jpg** que sea un pantallazo de la ejecución.

Ejemplo de clase ParkingTest y ejecución.

```

1 package main.clase.parking;
2 public class ParkingTest {
3     public static void main(String[] args) throws Exception {
4         System.out.println(x: "Inicio ParkingTest");
5         try {
6             Vehiculo coche1 = new Vehiculo(paramMatricula: "1234LZP");
7             ParkingVehiculo parkingCortes = new ParkingVehiculo(paramDir: "cortes 1 Madrid");
8             parkingCortes.aparcarCoche(coche1);
9             //Espero 2 minutos
10            Thread.sleep(120*1000);
11            parkingCortes.sacaCoche(coche1);
12            double dFactura= parkingCortes.getFactura(coche1);
13            System.out.println(coche1+"; Factura="+dFactura);
14        } catch (Exception e) {
15            e.printStackTrace();
16        }
17    }
18 }

```

PROBLEMAS 5 SALIDA CONSOLA DE DEPURACIÓN TERMINAL

```

C:\Program Files\Eclipse Adoptium\jdk-17.0.4-hotspot\bin\java.exe -XX:+ShowCodeDetailsInExceptionMessages -cp 'I:\AppData\Roaming\Code\User\workspaceStorage\8c6ee167b1286fc04180c0ff52a1fa3\redhat.java\jdt_ws\javaAccesoDatos_c5ac547d\bin' main.clase
ParkingTest
Vehiculo [InicioParking=2022-09-24T18:15:11.648966100, InicioParking=2022-09-24T18:13:11.611479900, sMatricula=1234LZP]; Factura=0.085
C:\Program Files\Eclipse Adoptium\jdk-17.0.4-hotspot\bin\java.exe

```

Ejercicio 4

Define la clase Punto que tiene como atributos las coordenadas x e y, de tipo entero, que lo sitúan en el plano.

Además del constructor, implementa el método

- *double distancia(Punto otroPunto)* que devuelve la distancia a otro punto que se le pasa como parámetro.

Crea la subclase Punto3D de Punto que representa un punto en tres dimensiones y necesita una coordenada adicional z. Reimplementa el método *distancia()* para puntos 3D

Implementa el método *equals()* para Punto y Punto3D, teniendo en cuenta que dos puntos son iguales si todas sus coordenadas son iguales.

Crea la subclase Suceso de Punto3D que incluye el atributo tiempo y descripción del suceso.

Implementa el método *equals()* para sucesos.