1- Diseñar la clase CuentaCorriente, que almacena los datos: DNI y nombre del titular, así como el saldo. Las operaciones típicas con una cuenta corriente son: • Crear una cuenta: se necesita el DNI y nombre del titular. El saldo inicial será 0. • Sacar dinero: el método debe indicar si ha sido posible llevar a cabo la operación, si existe saldo • suficiente. • Ingresar dinero: se incrementa el saldo. • Mostrar información: muestra la información disponible de la cuenta corriente.

**package** main;

**import** java.util.Scanner;

**public** **class** CuentaCorriente {

**private** String titular;

**private** String dni;

**private** **int** saldo;

**public** CuentaCorriente(String titular, String dni, **int** saldo) {

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = 0;

}

**public** String getTitular() {

**return** titular;

}

**public** **void** setTitular(String titular) {

**this**.titular = titular;

}

**public** String getDni() {

**return** dni;

}

**public** **void** setDni(String dni) {

**this**.dni = dni;

}

**public** **int** getSaldo() {

**return** saldo;

}

**public** **void** setSaldo(**int** saldo) {

**this**.saldo = saldo;

}

**public** **void** ingresarDinero(**float** ingreso) {

**if** (ingreso > 0) {

**this**.saldo += ingreso;

System.***out***.println("El saldo es: " + saldo);

}

}

**public** **void** retirarDinero() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Ingrese la cantidad de dinero a retirar");

**int** dineroRetirar = sc.nextInt();

**int** saldoRestante = saldo;

**if** (saldo > 0 && dineroRetirar < saldoRestante) {

saldoRestante = saldo - dineroRetirar;

System.***out***.println("Le queda " + saldoRestante);

} **else** {

System.***out***.println("No hay saldo suficiente");

}

}

}

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

CuentaCorriente cuenta = **new** CuentaCorriente("Garazi", "1234A", 0);

System.***out***.println("Para crear una cuenta se necesita el nombre del titular y el DNI");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println("Ingresando dinero...");

cuenta.ingresarDinero();

System.***out***.println("Retirando dinero...");

cuenta.retirarDinero();

System.***out***.println("Mostrando información: ");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println(cuenta.getSaldo());

}

}

2- En la clase CuentaCorriente sobrecargar los constructores para poder crear objetos. • Con el DNI del titular de la cuenta y un saldo inicial. • Con el DNI, nombre y el saldo inicial. Escribir un programa que compruebe el funcionamiento de los métodos

**public** CuentaCorriente(String dni, **int** saldo) {

**this**.dni = dni;

**this**.saldo = saldo;

}

3- Todas las cuentas corrientes con las que se va a trabajar pertenecen al mismo banco. Añadir un atributo que almacene el nombre del banco (que es único) en la clase CuentaCorriente. Diseñar un método que permita recuperar y modificar el nombre del banco (al que pertenecen todas las cuentas corrientes).

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

CuentaCorriente cuenta = **new** CuentaCorriente("Garazi", "1234A", 0);

System.***out***.println("Para crear una cuenta se necesita el nombre del titular y el DNI");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println("Ingresando dinero...");

cuenta.ingresarDinero();

System.***out***.println("Retirando dinero...");

cuenta.retirarDinero();

System.***out***.println("Mostrando información: ");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println(cuenta.getSaldo());

CuentaCorriente cuentaB = **new** CuentaCorriente("Garazi", "1234A", 0);

cuentaB.operacionBanco();

}

**package** main;

**import** java.util.Scanner;

**public** **class** CuentaCorriente {

**private** String titular;

**private** String dni;

**private** **int** saldo;

**private** **static** String *banco*;

**public** String getBanco() {

**return** *banco*;

}

**public** **void** setBanco(String banco) {

**this**.*banco* = banco;

}

**public** CuentaCorriente(String titular, String dni, **int** saldo) {

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = saldo;

}

**public** CuentaCorriente(String dni, **int** saldo) {

**this**.dni = dni;

**this**.saldo = saldo;

}

**public** CuentaCorriente(String titular, String dni, **int** saldo, String banco) {

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = saldo;

**this**.*banco* = banco;

}

**public** String getTitular() {

**return** titular;

}

**public** **void** setTitular(String titular) {

**this**.titular = titular;

}

**public** String getDni() {

**return** dni;

}

**public** **void** setDni(String dni) {

**this**.dni = dni;

}

**public** **int** getSaldo() {

**return** saldo;

}

**public** **void** setSaldo(**int** saldo) {

**this**.saldo = saldo;

}

**public** **void** operacionBanco() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Introduzca el nuevo nombre del banco");

*banco* = sc.nextLine();

System.***out***.println("El nuevo nombre del banco es " + *banco*);

}

**public** **void** ingresarDinero() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Ingrese dinero");

saldo = sc.nextInt();

System.***out***.println("El saldo es: " + saldo);

}

**public** **void** retirarDinero() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Ingrese la cantidad de dinero a retirar");

**int** dineroRetirar = sc.nextInt();

**int** saldoRestante = saldo;

**if** (saldo > 0 && dineroRetirar < saldoRestante) {

saldoRestante = saldo - dineroRetirar;

System.***out***.println("Le queda " + saldoRestante);

} **else** {

System.***out***.println("No hay saldo suficiente");

}

}

}

4- Existen gestores que administran las cuentas bancarias y atienden a sus propietarios. Cada cuenta, en caso de tenerlo, cuenta con un único gestor. Diseñar la clase Gestor de la que interesa guardar su nombre, teléfono y el importe máximo autorizado con el que pue­ den operar. Con respecto a los gestores, existen las siguientes restricciones: • Un gestor tendrá siempre un nombre y un teléfono. • Si no se asigna, el importe máximo autorizado por operación será de 10 000 euros. • Un gestor, una vez asignado, no podrá cambiar su número de teléfono. Y todo el mundo podrá consultarlo. El nombre será público y el importe máximo solo será visible por clases vecinas. Modificar la clase CuentaCorriente para que pueda disponer de un objeto Gestor. Escribir los métodos necesarios.

**package** main;

**import** java.util.Scanner;

**public** **class** CuentaCorriente {

**private** String titular;

**private** String dni;

**private** **int** saldo;

**private** String banco;

**public** Gestor gestor;

**public** String getBanco() {

**return** banco;

}

**public** **void** setBanco(String banco) {

**this**.banco = banco;

}

**public** CuentaCorriente(String titular, String dni, **int** saldo) {

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = saldo;

}

**public** CuentaCorriente(String dni, **int** saldo) {

**this**.dni = dni;

**this**.saldo = saldo;

}

**public** CuentaCorriente(String titular, String dni, **int** saldo, String banco) {

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = saldo;

**this**.banco = banco;

}

**public** CuentaCorriente(String titular, String dni, **int** saldo, Gestor gestor) { // Gestor gestor pasa por parámetro un objeto de tipo gestor con tdodos sus parámetros

**this**.titular = titular;

**this**.dni = dni;

**this**.saldo = saldo;

**this**.gestor = gestor;

}

**public** String getTitular() {

**return** titular;

}

**public** **void** setTitular(String titular) {

**this**.titular = titular;

}

**public** String getDni() {

**return** dni;

}

**public** **void** setDni(String dni) {

**this**.dni = dni;

}

**public** **int** getSaldo() {

**return** saldo;

}

**public** **void** setSaldo(**int** saldo) {

**this**.saldo = saldo;

}

**public** **void** operacionBanco() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Introduzca el nuevo nombre del banco");

banco = sc.nextLine();

System.***out***.println("El nuevo nombre del banco es " + banco);

}

**public** **void** ingresarDinero() {

**if** (**this**.gestor == **null**) {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Ingrese dinero");

saldo = sc.nextInt();

System.***out***.println("El saldo es: " + saldo);

} **else** **if** (**this**.gestor == gestor) {

**if** () {

}

}

}

**public** **void** retirarDinero() {

Scanner sc = **new** Scanner(System.***in***);

System.***out***.println("Ingrese la cantidad de dinero a retirar");

**int** dineroRetirar = sc.nextInt();

**int** saldoRestante = saldo;

**if** (saldo > 0 && dineroRetirar < saldoRestante) {

saldoRestante = saldo - dineroRetirar;

System.***out***.println("Le queda " + saldoRestante);

} **else** {

System.***out***.println("No hay saldo suficiente");

}

}

}

**package** main;

**public** **class** Gestor {

**public** String nombre;

**private** **int** telefono;

**int** importe;

**public** Gestor(String nombre, **int** telefono) {

**this**.nombre = nombre;

**this**.telefono = telefono;

}

**public** Gestor (String nombre, **int** telefono, **int** importe) {

**this**.nombre = nombre;

**this**.telefono = telefono;

**this**.importe = importe;

}

**public** String getNombre() {

**return** nombre;

}

**public** **void** setNombre(String nombre) {

**this**.nombre = nombre;

}

**public** **int** getTelefono() {

**return** telefono;

}

**public** **void** setTelefono(**int** telefono) {

**this**.telefono = 123456;

}

**public** **int** getImporte() {

**return** importe;

}

**public** **void** setImporte(**int** importe) {

**this**.importe = 10000;

}

}

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

CuentaCorriente cuenta = **new** CuentaCorriente("Garazi", "1234A", 0);

System.***out***.println("Para crear una cuenta se necesita el nombre del titular y el DNI");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println("Ingresando dinero...");

cuenta.ingresarDinero();

System.***out***.println("Retirando dinero...");

cuenta.retirarDinero();

System.***out***.println("Mostrando información: ");

System.***out***.println(cuenta.getTitular());

System.***out***.println(cuenta.getDni());

System.***out***.println(cuenta.getSaldo());

CuentaCorriente cuentaB = **new** CuentaCorriente("Garazi", "1234A", 0, "Dinerito");

cuentaB.operacionBanco();

}

}

5- Definir una clase que permita controlar un sintonizador digital de emisoras FM; concretamente, se desea dotar al controlador de una interfaz que permita subir (up) o bajar (down) la frecuencia (en saltos de 0,5 MHz) y mostrar la frecuencia sintonizada en un momento dado (display). Supondremos que el rango de frecuencias para manejar oscila entre los 80 MHz y los 108 MHz y que, al inicio, el controlador sintonice la frecuencia indicada en el constructor o 80 MHz por defecto. Si durante una operación de subida o bajada se sobrepasa uno de los dos límites, la frecuencia sintonizada debe pasar a ser la del extremo contrario. Escribir un pequeño programa principal para probar su funcionamiento.

package main;

public class Main {

public static void main(String[] args) {

Emisora em = new Emisora(79);

em.bajarFre(79);

em.subirFre(79);

em.mostrarFre(em.getFrecuencia());

}

}

**package** main;

**public** **class** Emisora {

**int** min;

**int** max;

**float** frecuencia;

**int** frecuenciaDefecto;

**public** **int** getMin() {

**return** min;

}

**public** **void** setMin(**int** min) {

**this**.min = 80;

}

**public** **int** getMax() {

**return** max;

}

**public** **void** setMax(**int** max) {

**this**.max = 180;

}

**public** **float** getFrecuencia() {

**return** frecuencia;

}

**public** **void** setFrecuencia(**float** frecuencia) {

**this**.frecuencia = frecuencia;

}

**public** **int** getFrecuenciaDefecto() {

**return** frecuenciaDefecto;

}

**public** **void** setFrecuenciaDefecto(**int** frecuenciaDefecto) {

**this**.frecuenciaDefecto = min;

}

**public** Emisora (**float** frecuencia) {

**this**.frecuencia = (**float**) frecuencia;

}

**public** **void** subirFre (**float** frecuencia) {

frecuencia += (**float**) 0.5f;

System.***out***.println("La nueva frecuencia es " + frecuencia);

**if** (frecuencia > 180) {

frecuencia = 80;

System.***out***.println("La frecuencia no puede pasar de 180. Bajando a 80");

}

}

**public** **void** bajarFre (**float** frecuencia) {

frecuencia -= (**float**) 0.5f;

System.***out***.println("La nueva frecuencia es " + frecuencia);

**if** (frecuencia < 80) {

frecuencia = 180;

System.***out***.println("La frecuencia no puede bajar de 80. Subiendo a 180");

}

}

**public** **void** mostrarFre (**float** frecuencia) {

System.***out***.println("Estás sintonizando la emisora " + frecuencia);

}

}

6- . Modelar una casa con muchas bombillas, de forma que cada bombilla se pueda encender o apagar individualmente. Para ello, hacer una clase Bombilla con una variable privada que indique si está encendida o apagada, así como un método que nos diga el estado de una bombilla concreta. Además, queremos poner un interruptor general, de forma que si este se apaga, todas las bombillas quedan apagadas. Cuando el interruptor general se activa, las bombillas vuelven a estar encendidas o apagadas, según estuvieran antes. Cada bombilla se enciende y se apaga individualmente, pero solo responde que está encendida si su interruptor particular está activado y además hay luz general.

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Bombilla bom = **new** Bombilla (**true**, **false**);

bom.estadoBombilla();

bom.interpGen();

}

}

**package** main;

**public** **class** Bombilla {

**private** **boolean** encendida;

**public** **static** **boolean** *interruptorGen*;

**private** **boolean** interruptorInd;

**public** Bombilla(**boolean** encendida, **boolean** interruptorInd) {

**this**.encendida = encendida;

**this**.interruptorInd = interruptorInd;

}

**public** **static** **boolean** isInterruptorGen() {

**return** *interruptorGen*;

}

**public** **static** **void** setInterruptorGen(**boolean** interruptorGen) {

Bombilla.*interruptorGen* = **true**;

}

**public** **boolean** isInterruptorInd() {

**return** interruptorInd;

}

**public** **void** setInterruptorInd(**boolean** interruptorInd) {

**this**.interruptorInd = interruptorInd;

}

**public** **boolean** isEncendida() {

**return** encendida;

}

**public** **void** setEncendida(**boolean** encendida) {

**this**.encendida = encendida ;

}

**public** **void** estadoBombilla() {

**if** (**this**.encendida == **true** && **this**.interruptorInd == **true**) {

System.***out***.println("La bombilla está encendida");

} **else** {

System.***out***.println("La bombilla está apagada");

}

}

**public** **void** interpGen() {

**if** (**this**.*interruptorGen* == **true** && **this**.encendida == **true**) {

**this**.encendida = **false**;

System.***out***.println("La bombilla está ahora apagada");

} **else** **if** ((**this**.*interruptorGen* == **true** && **this**.encendida == **false**)) {

**this**.encendida = **true**;

System.***out***.println("La bombilla está ahora encendida");

}

}

}

7- . Hemos recibido el encargo de un cliente para definir los paquetes y las clases necesarias (solo implementar los atributos y los constructores) para gestionar una empresa ferroviaria, en la que se distinguen dos grandes grupos: el personal y la maquinaria. En el primero se ubican todos los empleados de la empresa, que se clasifican en tres grupos: los maquinistas, los mecánicos y los jefes de estación. De cada uno de ellos es necesario guardar: • Maquinistas: su nombre, DNI, sueldo y el rango que tienen adquirido. • Mecánicos: su nombre, teléfono (para contactar en caso de urgencia) y en qué especialidad desarrollan su trabajo (esta puede ser: frenos, hidráulica, electricidad o motor). • Jefes de estación: su nombre, DNI y la fecha en la que fue nombrado jefe de estación. En la parte de maquinaria podemos encontrar trenes, locomotoras y vagones. De cada uno de ellos hay que considerar: • Vagones: tienen un número que los identifica, una carga máxima (en kilos), la carga actual y el tipo de mercancía con el que están cargados. • Locomotoras: disponen de una matrícula (que las identifica), la potencia de sus motores y una antigüedad (año de fabricación). Además, cada locomotora tiene asignado un mecánico que se encarga de su mantenimiento.

Trenes: están formados por una locomotora y un máximo de 5 vagones. Cada tren tiene asignado un maquinista que es responsable de él. Todas las clases correspondientes al personal (Maquinista, Mecanico y JefeEstacion) serán de uso público. Entre las clases relativas a la maquinaria solo será posible construir, desde clases externas, objetos de tipo Tren y de tipo Locomotora. La clase Vagon será solo visible por sus clases vecinas.

**package** maquinaria;

**import** personal.Mecanico;

**public** **class** Locomotora {

**public** **int** matricula;

**public** **int** potencia;

**public** **int** antiguedad;

**public** Mecanico mecanico;

**public** Locomotora (**int** matricula, **int** potencia, **int** antiguedad, Mecanico persona) {

**this**.matricula = matricula;

**this**.potencia = potencia;

**this**.antiguedad = antiguedad;

**this**.mecanico = mecanico;

}

}

**package** maquinaria;

**import** personal.Maquinista;

**public** **class** Tren {

**private** Locomotora locomotora;

**private** Vagon vagon;

**public** Maquinista maquinista;

**public** Tren(Locomotora locomotora, Vagon vagon, Maquinista maquinista) {

**this**.locomotora = locomotora;

**this**.vagon = vagon;

**this**.maquinista = maquinista;

}

}

**package** maquinaria;

**class** Vagon {

**int** numero;

**int** carga;

String tipoMercancia;

Vagon(**int** numero, **int** carga, String tipoMercancia) {

**this**.numero = numero;

**this**.carga = carga;

**this**.tipoMercancia = tipoMercancia;

}

}

**package** personal;

**public** **class** JefeEstacion {

**private** String nombre;

**private** String DNI;

**private** String fechaNombramiento;

**public** JefeEstacion(String nombre, String DNI, String fechaNombramiento) {

**this**.nombre = nombre;

**this**.DNI = DNI;

**this**.fechaNombramiento = fechaNombramiento;

}

}

**package** personal;

**public** **class** Maquinista {

String nombre;

String DNI;

**int** sueldo;

String rango;

**public** Maquinista(String nombre, String DNI, **int** sueldo, String rango) {

**this**.nombre = nombre;

**this**.DNI = DNI;

**this**.sueldo = sueldo;

**this**.rango = rango;

}

}

**package** personal;

**public** **class** Mecanico {

**private** String nombre;

**private** **int** telefono;

**private** Especialidad especialidad;

**enum** Especialidad {

***FRENOS***, ***HIDRÁULICA***, ***ELECTRICIDAD***, ***MOTOR***};

**public** **void** Mecanico(String nombre, **int** telefono, String especialidad) {

**this**.nombre = nombre;

**this**.telefono = telefono;

**this**.especialidad = Especialidad.*valueOf*(especialidad);

}

}

8- Crea una clase llamada Libro que guarde la información de cada uno de los libros de una biblioteca. La clase debe guardar el título del libro, autor, número de ejemplares del libro y número de ejemplares prestados. La clase contendrá los siguientes métodos: • Constructor por defecto. • Constructor con parámetros. • Métodos Setters/getters • Método préstamo que incremente el atributo correspondiente cada vez que se realice un préstamo del libro. No se podrán prestar libros de los que no queden ejemplares disponibles para prestar. Devuelve true si se ha podido realizar la operación y false en caso contrario. • Método devolución que decremente el atributo correspondiente cuando se produzca la devolución de un libro. No se podrán devolver libros que no se hayan prestado. Devuelve true si se ha podido realizar la operación y false en caso contrario. • Método toString para mostrar los datos de los libros. Este método se heredada de Object y lo debemos modificar (override) para adaptarlo a la clase Libro. Escribe un programa para probar el funcionamiento de la clase Libro.

**package** main;

**public** **class** Main {

**private** **static** Object *toString*;

**public** **static** **void** main(String[] args) {

Libro lib = **new** Libro ("RPG", "Yo Misma", 10, 0);

lib.prestamo ("RPG", "Yo Misma", 20, 21);

lib.devolucion("RPG", "Yo Misma", 11, 20);

lib.toString();

}

}

**package** main;

**public** **class** Libro {

**private** String titulo;

**private** String autor;

**private** **int** ejemplares;

**private** **int** ejemplaresPrestados;

**public** Object toString;

**public** String getTitulo() {

**return** titulo;

}

**public** **void** setTitulo(String titulo) {

**this**.titulo = titulo;

}

**public** String getAutor() {

**return** autor;

}

**public** **void** setAutor(String autor) {

**this**.autor = autor;

}

**public** **int** getEjemplares() {

**return** ejemplares;

}

**public** **void** setEjemplares(**int** ejemplares) {

**this**.ejemplares = ejemplares;

}

**public** **int** getEjemplaresPrestados() {

**return** ejemplaresPrestados;

}

**public** **void** setEjemplaresPrestados(**int** ejemplaresPrestados) {

**this**.ejemplaresPrestados = ejemplaresPrestados;

}

**public** Libro (String titulo, String autor, **int** ejemplares, **int** ejemplaresPrestados) {

**this**.titulo = titulo;

**this**.autor = autor;

**this**.ejemplares = ejemplares;

**this**.ejemplaresPrestados = ejemplaresPrestados;

}

**public** Libro() {

**this**.titulo = titulo;

**this**.autor = autor;

**this**.ejemplares = ejemplares;

**this**.ejemplaresPrestados = ejemplaresPrestados;

}

**public** **void** prestamo(String titulo, String autor, **int** ejemplares, **int** ejemplaresPrestados){

**boolean** sePuedePrestar = **false**;

**if** (ejemplares > ejemplaresPrestados) {

ejemplaresPrestados++;

System.***out***.println("Hay ejemplares suficientes para hacer un préstamo");

sePuedePrestar = **true**;

}

}

**public** **void** devolucion(String titulo, String autor, **int** ejemplares, **int** ejemplaresPrestados){

**boolean** sePuedePrestar = **false**;

**if** (ejemplares < ejemplaresPrestados) {

ejemplaresPrestados--;

System.***out***.println("No hay ejemplares suficientes para hacer un préstamo");

sePuedePrestar = **true**;

}

}

**public** String toString() {

System.***out***.println(**this**.autor);

System.***out***.println(**this**.titulo);

System.***out***.println(**this**.ejemplares);

System.***out***.println(**this**.ejemplaresPrestados);

**return** (String) toString;

}

}

9- Se requiere un programa que modele el concepto de un planeta del sistema solar. Un planeta tiene los siguientes atributos: • Un nombre de tipo Sring. • Cantidad de satélites. • Masa en kilogramos. • Volumen en kilómetros cúbicos. • Diámetro en kilómetros. • Distancia media al Sol en millones de kilómetros. • Tipo de planeta de acuerdo con su tamaño, de tipo enumerado con los siguientes valores posibles: GASEOSO, TERRESTRE y ENANO. Observable a simple vista, de tipo booleano con valor inicial false. La clase debe incluir los siguientes métodos: • La clase debe tener un constructor que inicialice los valores de sus respectivos atributos. Definir un método que imprima en pantalla los valores de los atributos de un planeta. • Calcular la densidad un planeta, como el cociente entre su masa y • su volumen. • Determinar si un planeta del sistema solar se considera exterior. • Un planeta exterior está situado más allá del cinturón de asteroides. El cinturón de asteroides se encuentra entre 2.1 y 3.4 UA. Una unidad astronómica (UA) es la distancia entre la Tierra y el Sol= 149597870 Km. • En un método main se deben crear dos planetas y mostrar los valores de sus atributos en pantalla. Además, se debe imprimir la densidad de cada planeta y si el planeta es un planeta exterior del sistema solar.

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

Planeta p = **new** Planeta ();

p.mostrar();

p.densidad(1000000, 999999999);

p.esExterior(10000000);

Planeta p2 = **new** Planeta ();

p2.mostrar();

p2.densidad(5972 \* Math.*pow*(10, 24), 108321 \* 1012);

p2.esExterior(150000000);

}}

**package** main;

**import** main.TipoPlaneta;

**public** **class** Planeta {

**private** String nombre;

**private** **int** satelites;

**private** **double** masaKG;

**private** **double** volKM3;

**private** **double** diametro;

**private** **double** KMSol;

**boolean** observable = **false**;

**private** TipoPlaneta TipoPlaneta;

**public** Planeta() {

**this**.nombre = "Nirn";

**this**.satelites = 2;

**this**.masaKG = 999999;

**this**.volKM3 = 100000;

**this**.diametro = 1000000;

**this**.KMSol = 50;

**this**.observable = **false**;

**this**.TipoPlaneta = TipoPlaneta.***TERRESTRE***;

}

**public** **void** PlanetaDos() {

**this**.nombre = "Tierra";

**this**.satelites = 1;

**this**.masaKG = 5972 \* Math.*pow*(10, 24);

**this**.volKM3 = 108321 \* 1012;

**this**.diametro = 12756;

**this**.KMSol = 150000000;

**this**.observable = **true**;

**this**.TipoPlaneta = TipoPlaneta.***TERRESTRE***;

}

**public** String getNombre() {

**return** nombre;

}

**public** **void** setNombre(String nombre) {

**this**.nombre = nombre;

}

**public** **int** getSatelites() {

**return** satelites;

}

**public** **void** setSatelites(**int** satelites) {

**this**.satelites = satelites;

}

**public** **double** getMasaKG() {

**return** masaKG;

}

**public** **void** setMasaKG(**double** masaKG) {

**this**.masaKG = masaKG;

}

**public** **double** getVolKM3() {

**return** volKM3;

}

**public** **void** setVolKM3(**double** volKM3) {

**this**.volKM3 = volKM3;

}

**public** **double** getDiametro() {

**return** diametro;

}

**public** **void** setDiametro(**double** diametro) {

**this**.diametro = diametro;

}

**public** **double** getKMSol() {

**return** KMSol;

}

**public** **void** setKMSol(**double** kMSol) {

KMSol = kMSol;

}

**public** **boolean** isObservable() {

**return** observable;

}

**public** **void** setObservable(**boolean** observable) {

**this**.observable = observable;

}

**public** **void** mostrar() {

System.***out***.println("El planeta se llama " + **this**.nombre);

System.***out***.println("Diámetro: " + **this**.diametro);

System.***out***.println("Satélites " + **this**.satelites);

System.***out***.println("Masa en kg " + **this**.masaKG);

System.***out***.println("Volumen " + **this**.volKM3);

System.***out***.println("Km al sol " + **this**.KMSol);

System.***out***.println("Observable:" + **this**.observable);

System.***out***.println("Tipo de planeta: " + **this**.TipoPlaneta);

}

**public** **void** densidad(**double** masaKG, **double** volKM3) {

**double** den = masaKG / volKM3;

System.***out***.println("La densidad del planeta es " + den);

}

**public** **void** esExterior(**double** KMSol) {

**final** **double** UA = 149597870;

**boolean** esExt = **false**;

**if** (KMSol > (UA \* (**float**) 3.4)) {

**boolean** exExt = **true**;

System.***out***.println("El planeta es exterior");

} **else** {

esExt = **false**;

System.***out***.println("El planeta no es exterior");

}

}

}

**package** main;

**enum** TipoPlaneta {

***GASEOSO***, ***TERRESTRE***, ***ENANO***

}

10- Realizar un programa en Java que permita modelar un artículo científico. Los artículos científicos contienen los siguientes metadatos: nombre del artículo, autor, palabras claves, nombre de la publicación, año y resumen. Se deben definir tres constructores sobrecargados: • El primero inicializa un artículo científico con solo su título y autor. • El segundo constructor, un artículo científico con su nombre, autor, palabras claves, nombre de la publicación y año. Debe invocar al primer constructor. • El tercer constructor, un artículo científico con su nombre, autor, palabras claves, nombre de la publicación, año y resumen. Debe invocar al segundo constructor. Se requiere un método que imprima los atributos de un artículo en pantalla. Realizar un método main que utilice el tercer constructor para instanciar un artículo científico e imprima los valores de sus atributos en pantalla.

**package** main;

**public** **class** Main {

**public** **static** **void** main(String[] args) {

ArticuloCientifico ac = **new** ArticuloCientifico();

ac.ArticuloCientifico("Historia de la magia en la cultura popular", "Merlín", "magia, historia, cultura pop", "La runa dorada", 2024, "Lorem ipsum dolor sit amet");

ac.mostrarInfo();

}

}

**package** main;

**public** **class** ArticuloCientifico {

**private** String titulo;

**private** String autor;

**private** String palabrasClave;

**private** String nombrePublicacion;

**private** **int** anyo;

**private** String resumen;

**public** **void** ArticuloCientifico(String titulo, String autor) {

**this**.titulo = titulo;

**this**.autor = autor;

}

**public** **void** ArticuloCientifico(String titulo, String autor, String palabrasClave, String nombrePublicacion, **int** anyo) {

**this**.titulo = titulo;

**this**.autor = autor;

**this**.palabrasClave = palabrasClave;

**this**.nombrePublicacion = nombrePublicacion;

**this**.anyo = anyo;

}

**public** **void** ArticuloCientifico(String titulo, String autor, String palabrasClave, String nombrePublicacion, **int** anyo, String resumen) {

**this**.titulo = titulo;

**this**.autor = autor;

**this**.palabrasClave = palabrasClave;

**this**.nombrePublicacion = nombrePublicacion;

**this**.anyo = anyo;

**this**.resumen = resumen;

}

**public** String getNombre() {

**return** titulo;

}

**public** **void** setNombre(String nombre) {

**this**.titulo = nombre;

}

**public** String getAutor() {

**return** autor;

}

**public** **void** setAutor(String autor) {

**this**.autor = autor;

}

**public** String getPalabrasClave() {

**return** palabrasClave;

}

**public** **void** setPalabrasClave(String palabrasClave) {

**this**.palabrasClave = palabrasClave;

}

**public** String getNombrePublicacion() {

**return** nombrePublicacion;

}

**public** **void** setNombrePublicacion(String nombrePublicacion) {

**this**.nombrePublicacion = nombrePublicacion;

}

**public** **int** getAnyo() {

**return** anyo;

}

**public** **void** setAnyo(**int** anyo) {

**this**.anyo = anyo;

}

**public** String getResumen() {

**return** resumen;

}

**public** **void** setResumen(String resumen) {

**this**.resumen = resumen;

}

**public** **void** mostrarInfo() {

System.***out***.println("El título del artículo es: " + **this**.titulo + ". Su autor es " + **this**.autor + ", con las palabras clave de " + **this**.palabrasClave + ", publicado en el año " + **this**.anyo + " en la publicación " + **this**.nombrePublicacion + ", y el resumen de su contenido es " + **this**.resumen);

}

}