Modelo del PRIMER PARCIAL – PROGRAMACIÓN II – 2024

De acuerdo a las descripciones de las siguientes clases, se pide:
Modelar en UML, crear el código correspondiente en JAVA (reutilizando código).

Generar un proyecto en JAVA nombrado como: ***Entidades.Apellido.Nombre***, que sea de tipo *Biblioteca de Clases,* el cual tendrá las siguientes clases (todas en el paquete ***apellido.nombre***):

Pasajero con los atributos privados: apellido (String), nombre (String) y rango (**Rangos:** *AZAFATA, COCINERO, MAQUINISTA y CLIENTE*. Agregar el archivo correspondiente).

Agregar los métodos ***getters*** a cada uno de los atributos.

Diseñar el constructor y sus sobrecargas para poder inicializar los objetos de tipo Pasajero de la siguiente forma:

- El constructor con cero parámetros inicializará a sus atributos con "SIN APELLIDO", "SIN NOMBRE" y Rangos.Cliente, según corresponda.
- El constructor con un parámetro inicializará a sus atributos con el apellido recibido, "SIN NOMBRE" y Rangos.Cliente, según corresponda.
- El constructor con dos parámetros inicializará a sus atributos con el apellido recibido, el nombre recibido y Rangos.Cliente, según corresponda.
- El constructor con tres parámetros inicializará a sus atributos con el apellido recibido, el nombre recibido y el rango recibido, según corresponda.

Agregar el método público y de clase ***sonIguales(Pasajero, Pasajero)***, retorna ***true***, si los pasajeros tienen el mismo nombre, apellido y rango, ***false***, caso contrario.

Aplicar polimorfismo en el método **toString().** Utilizar, preferentemente, el objeto **StringBuilder.**

Aplicar polimorfismo en el método **equals()**, para que retorne ***true***, si el pasajero tiene el mismo nombre y apellido, ***false***, caso contrario**.**

Aplicar polimorfismo en el método **hashCode()**, para que retorne el valor del ***hashCode*** del toString del pasajero (Objects.hashCode())**.**

Crear la clase "**Tren**" que posea los atributos que **solo** deberán ser visibles desde la propia clase y las clases que deriven de ella.
Los atributos serán: cantidadMaximaPasajeros (entero), destino (cadena de caracteres) y motorEncendido (booleano).

Agregar los getters y setters para los atributos ***cantidadMaximaPasajeros*** y ***motorEncendido***, con las siguientes características: el atributo de tipo entero, será de solo lectura, mientras que el atributo booleano, será de lectura y escritura.

Crear el método getter para el atributo ***destino***, que retorne el valor de dicho atributo en mayúscula y que pueda indicar el tipo de tren (en las clases derivadas).

Realizar un método de instancia y público llamado ***getPasajeros():ArrayList<Pasajero>***, que se deba implementar en las clases derivadas y que se relaciona con el atributo ***pasajeros*** de dichas clases.

Diseñar su constructor para que reciba un entero y una cadena para inicializar los atributos correspondientes.

Realizar el método estático ***sonIguales(Pasajero, Tren)*****,** que retornará ***true***, si el pasajero (primer parámetro) se encuentra en el tren (segundo parámetro), ***false***, caso contrario.

Realizar el método de instancia ***agregar(Pasajero)***, que no retorna nada. Recibe, cómo parámetro un objeto de tipo **Pasajero**, el cuál se agrega al tren, siempre y cuando la cantidad máxima de pasajeros no sea superada (informar si se ha superado) y el pasajero no esté ya en el tren (informar si ya existe el pasajero).

Aplicar polimorfismo en:

- ***toString***, para que retorne la información completa del tren en formato de cadena de caracteres.

Crear una clase llamada "TrenElectrico" que herede de "Tren" y que posea el atributo protegido y no escalar, ***pasajeros*** (lista de *tipo Pasajero)*. Asociar este atributo con el método *getPasajeros*.

Aplicar polimorfismo en el método ***toString()***, para que retorne la información completa del tren en formato de cadena de caracteres.

Crear una clase llamada "TrenBala" que herede de "Tren" y que posea los atributos protegidos **cantidadElectroimanes** (de tipo entero) y **pasajeros** (lista de tipo *Pasajero*). Asociar este atributo con el método *getPasajeros*.

A esta clase se le indicará el valor de la cantidad de electroimanes a través de su único constructor.

Aplicar polimorfismo en el método ***toString()***, para que retorne la información completa del tren en formato de cadena de caracteres.

Generar el ***.jar*** del proyecto de tipo Biblioteca de Clases. Crear un nuevo proyecto de tipo Aplicación de Consola (Test.Apellido.Nombre), agregar el ***.jar*** y copiar las siguientes líneas de código en el método ***main***, sin modificar nada.

MAIN:

```
Pasajero p0 = new Pasajero();
Pasajero p1 = new Pasajero("Pratto");
Pasajero p2 = new Pasajero("Quintero", "Juanfer");
Pasajero p3 = new Pasajero("Pratto", "Oso", Rangos.MAQUINISTA);
Pasajero p4 = new Pasajero();
Pasajero p5 = new Pasajero("Martinez", "Pity", Rangos.COCINERO);
Pasajero p6 = new Pasajero("Martinez", "Pity", Rangos.CLIENTE);
String p7 = "pasajero7";

boolean sonIguales = Pasajero.sonIguales(p0, p4);

if(sonIguales)
{
    System.out.println("Los pasajeros 0 y 4, son iguales");
    System.out.println(p0.toString());
}

sonIguales = p4.equals(p0);
if(sonIguales)
{
    System.out.println("Los pasajeros 0 y 4, son iguales");
    System.out.println(p4.toString());
}

if(p1.equals(p3))
{
    System.out.println("Los pasajeros 1 y 3, son iguales");
    System.out.println(p1.toString());
    System.out.println();
}

if( ! Pasajero.sonIguales(p5, p6))
{
    System.out.println("Los pasajeros 5 y 6, NO son iguales");
}

if( ! p2.equals(p7))
{
    System.out.println("Los pasajeros 2 y 7, NO son iguales");
}

System.out.println("Codigo hash pasajero 5: " + p5.hashCode());
System.out.println("Codigo hash pasajero 6: " + p6.hashCode());
System.out.println();

TrenElectrico te = new TrenElectrico(4, "Constitucion");
TrenBala tb = new TrenBala(5, "Rosario", 350);
```

```
te.agregar(p0);
te.agregar(p1);
// REPETIDO
te.agregar(p4);
te.agregar(p2);
te.agregar(p3);
// NO HAY LUGAR
te.agregar(p5);

System.out.println();
// SE MUESTRA TREN ELECTRICO
System.out.println(te);

tb.getCantidadMaximaPasajeros();
tb.getDestino();

tb.agregar(p6);
tb.agregar(p5);

if(Tren.sonIguales(p6, tb))
{
    System.out.println("El pasajero ya esta en el tren:");
    System.out.println(p6);
}

// SE MUESTRA TREN BALA
System.out.println(tb);
```

```
Los pasajeros 0 y 4, son iguales
APELLIDO: SIN APELLIDO
NOMBRE: SIN NOMBRE
RANGO: CLIENTE

Los pasajeros 0 y 4, son iguales
APELLIDO: SIN APELLIDO
NOMBRE: SIN NOMBRE
RANGO: CLIENTE

Los pasajeros 5 y 6, NO son iguales
Los pasajeros 2 y 7, NO son iguales
Codigo hash pasajero 5: 1254258750
Codigo hash pasajero 6: -1761219658

El pasajero ya esta en el tren.
Tren completo!!!

CANTIDAD MAXIMA DE PASAJEROS: 4
MOTOR ENCENDIDO?: false
DESTINO: El tren electrico tiene destino: CONSTITUCION
PASAJEROS:
APELLIDO: SIN APELLIDO
NOMBRE: SIN NOMBRE
RANGO: CLIENTE
APELLIDO: Pratto
NOMBRE: SIN NOMBRE
RANGO: CLIENTE
APELLIDO: Quintero
NOMBRE: Juanfer
RANGO: CLIENTE
APELLIDO: Pratto
NOMBRE: Oso
RANGO: MAQUINISTA

El pasajero ya esta en el tren:
APELLIDO: Martinez
NOMBRE: Pity
RANGO: CLIENTE

CANTIDAD MAXIMA DE PASAJEROS: 5
MOTOR ENCENDIDO?: false
DESTINO: El tren bala tiene destino: ROSARIO
CANTIDAD ELECTROIMANES: 350
PASAJEROS:
APELLIDO: Martinez
NOMBRE: Pity
RANGO: CLIENTE
APELLIDO: Martinez
NOMBRE: Pity
RANGO: COCINERO
```

SALIDA DE CONSOLA ESPERADA.