



Examen 2do Parcial

Valor: 70 Puntos

Fecha: 12/11/2016

Una compañía que fabrica AirBags (Bolsas de aire de seguridad) para diferentes modelos de vehículos, en conjunto con el departamento de Matemática y Física de la Universidad Nacional Experimental del Táchira, desean realizar una serie de experimentos para determinar algunos parámetros, necesarios para la fabricación óptima de sus dispositivos de seguridad.

Pero antes de gastar unos cuantos millones de Bs en experimentos, requieren hacer unos cálculos teóricos a través de un programa en Java realizado por usted que procese diferentes combinaciones de experimentos en segundos.

En teoría de choques, la cantidad de movimiento de un cuerpo se define como Masa (m_x) por Velocidad (V_x), y la cantidad de movimiento final de un cuerpo (resultado de un choque inelástico) es la suma de las cantidades de movimiento de los cuerpos involucrados en el choque.

$$p = m_x V_x \quad (\text{Cantidad de movimiento})$$

$$m_1 V_1 + m_2 V_2 = (m_1 + m_2) V_f \quad (\text{Cantidad de movimiento final})$$

Y la energía mecánica perdida (en Joule) en ese choque (inelástico) se define con:

$$Em_o = \frac{1}{2} m_1 V_1^2 + \frac{1}{2} m_2 V_2^2 \quad (\text{Inicial})$$

$$Em_f = \frac{1}{2} (m_1 + m_2) V_f^2 \quad (\text{Final})$$

$$\Delta Em = Em_f - Em_o \quad (\text{Energía mecánica perdida})$$


En cuanto a conversiones se refiere tenemos:

$$1 \text{ Tonelada} = 1000 \text{ Kg}$$

$$1 \text{ Km/h} = 1000 \text{mts}/3600 \text{seg}$$

$$1 \text{ J} = \text{kg} \cdot \text{m}^2/\text{s}^2$$

Se tienen diferentes tipos de vehículos (Cuerpos) cada uno con una masa específica:

Vehículo			
	Automóvil (1)	Camioneta (2)	Autobús (3)
Masa	Masa neta + 5 puestos + 0 cargas	Masa neta + 2 puestos + 20 cargas	Masa neta + 32 puestos + 16 cargas

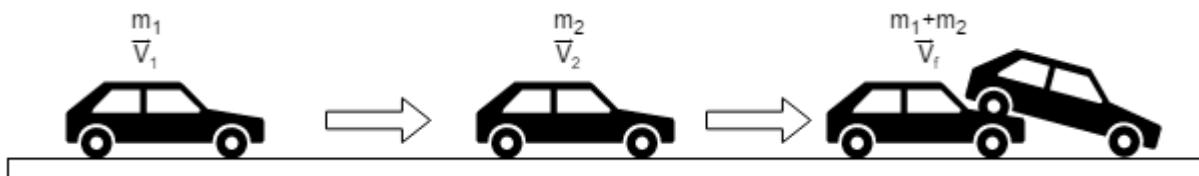
Nota: Un **Puesto** es una persona que se encuentra en el vehículo con una masa promedio de 50Kg y las **Cargas** tienen una masa promedio de 20Kg c/u.

Se requiere un programa escrito en Java que contemple los siguientes requerimientos:

- ❖ Una clase **abstracta** denominada **Vehículo** en el paquete **Insumos** con las siguientes características:
 - Atributos de texto para **marca** y **modelo** del vehículo.

- Atributo **ID** del vehículo.
 - Un único constructor paramétrico que inicializa todos los atributos.
 - Métodos Getter para todos los atributos.
 - No contiene métodos Setter.
 - Método **abstracto masaTotal** en toneladas, que devuelve el cálculo de la masa para cada tipo de vehículo sumando la masa neta del vehículo + total masa puestos + total masa de cargas.
- ❖ Una interfaz **Valores** en el paquete **Insumos** con los siguientes atributos;
- Constante **masaPuestos** con valor 50,0 Kgs.
 - Constante **masaCargas** con valor 20,0 Kgs.
- ❖ Una clase para cada tipo de vehículo (**Automóvil, Camioneta, Autobús**) en el paquete **Insumos** y que hereda de la clase **Vehículo** e implemente la interfaz **Valores**, con las siguientes características:
- Atributo **masaNeta** del vehículo en Kilogramos.
 - Un constructor paramétrico que reciba la marca, modelo, ID y masa neta del vehículo. Información extraída del archivo: **vehiculos.txt**
 - Métodos getter de **masaNeta**.
 - Implementación del Método **masaTotal**, el cual devuelve la suma de las masas neta + puestos + cargas de acuerdo a la tabla de vehículos y usando las constantes de la interfaz Valores.
- Ejemplo:**
Un Vehículo Automóvil con peso neto 1.61 Toneladas devolvería: 1610 (masa neta)+ 5 * 50 (puestos)+ 0 * 20 (carga) como masa total en Kgs.
- ❖ Una clase **Cuerpo** en el paquete **Laboratorio** que contenga lo siguiente:
- Un atributo **velocidad** del tipo double expresada en m/s. Información extraída del archivo: **experimentos.txt**
 - Un atributo **vehículo** del tipo Clase Vehículo. Información extraída del archivo: **experimentos.txt**
 - Un constructor paramétrico para velocidad y vehículo.
 - Un método getter de velocidad.
 - Sobrescribir método **ToString()** que devuelve una cadena con información del cuerpo en el siguiente formato:
 - {<Tipo Vehículo>: {<marca>,<modelo>},Velocidad:<velocidad> m/s,Masa:<masa> Kgs}}
 - **Ejemplo:**
 - {Camioneta:{Ford,F150},Velocidad: 0 m/s, Masa: 2250 Kgs}
 - Método **calcularCantMov** (Calcular cantidad de movimiento) que devuelve un valor double de acuerdo a la fórmula antes descrita en Kg*m/s.
 - Método **calcularEnergiaMec** (Calcular Energía Mecánica) que devuelve un valor double de acuerdo a la fórmula antes descrita en Joule (J).
 - Método **getMasa()** que retorna la masaTotal del vehículo.
- ❖ Una clase **Experimento** en el paquete **Laboratorio** que conste de lo siguiente:

- Dos atributos **cuerpo1** y **cuerpo2** del tipo Clase Cuerpo que serán los elementos involucrados en el choque.
- Constructor paramétrico, que permite inicializar los atributos de la clase.
- Un método **getVelocidadFinal()** que devuelve la suma de la velocidad final de ambos cuerpos incrustados justo después del choque, de acuerdo a las fórmulas facilitadas.
- Un método **getEnergiaMecFinal()** que devuelve la suma de la Energía Mecánica final de ambos cuerpos incrustados justo después del choque, de acuerdo a las fórmulas facilitadas.
- Un método **getEnergiaMecPerdida()** que devuelve la Diferencia entre la suma de las Energía Mecánicas de los cuerpos antes del choque, contra la Energía mecánica final de ambos cuerpos.
- Un método **imprimirExperimento()** que devuelve toda la información relacionado al experimento: Información de los cuerpos (la que retorna el método toString), velocidad final y energía mecánica perdida. De acuerdo al siguiente formato:
 - Experimento:
 - Cuerpo 1: <Información cuerpo 1>
 - Cuerpo 2: <Información cuerpo 2>
 - Velocidad Final: <Velocidad final>
 - Energía Mecánica Perdida: <Energía mecánica perdida>



- ❖ Una clase **Principal** en el paquete **Laboratorio** que conste de lo siguiente:
 - Arreglo **vehículos** de la clase **Vehículo**.
 - Arreglo **experimentos** de la clase **Experimento**.
 - Método **cargarDatosVehiculos()** que se encargará de leer el archivo **vehiculos.txt** para cargar la información correspondiente de la Clase **Vehículos** y sus **subclases** en el arreglo **vehículos**
 - Método **cargarDatosExperimentos()** que se encargará de leer el archivo **experimentos.txt** para cargar la información correspondiente de la Clase **Experimento** y **Cuerpo** en el arreglo **experimentos** junto a la información cargada en el vector **vehículos**.
 - Método **ejecutarExperimentos()** que invoque de todos los experimentos el método **imprimirExperimento()** para ver toda su información. Separe los datos de cada experimento con un mensaje de encabezado "--Experimento #--", donde # es el número del experimento.
 - Método **main** que instancie o invoque cada método de ésta clase.

Contenido de los archivos:

vehiculos.txt

```
1-Ford-Laser-1.61-01
1-Mazda-Mazda6-1.82-02
2-Ford-F150-2.25-03
1-Renault-Logan-1.54-04
3-Iveco-59 12-3.12-05
2-Toyota-Tacoma-2.10-06
```

1-Chery-QQ-0.98-07
 3-Scania-Irizar-4.7-08
 2-Mazda-BT50-1.98-09
 3-Encava-ENT 610-3.51-10
 3-Ford-Turbico-3.76-11

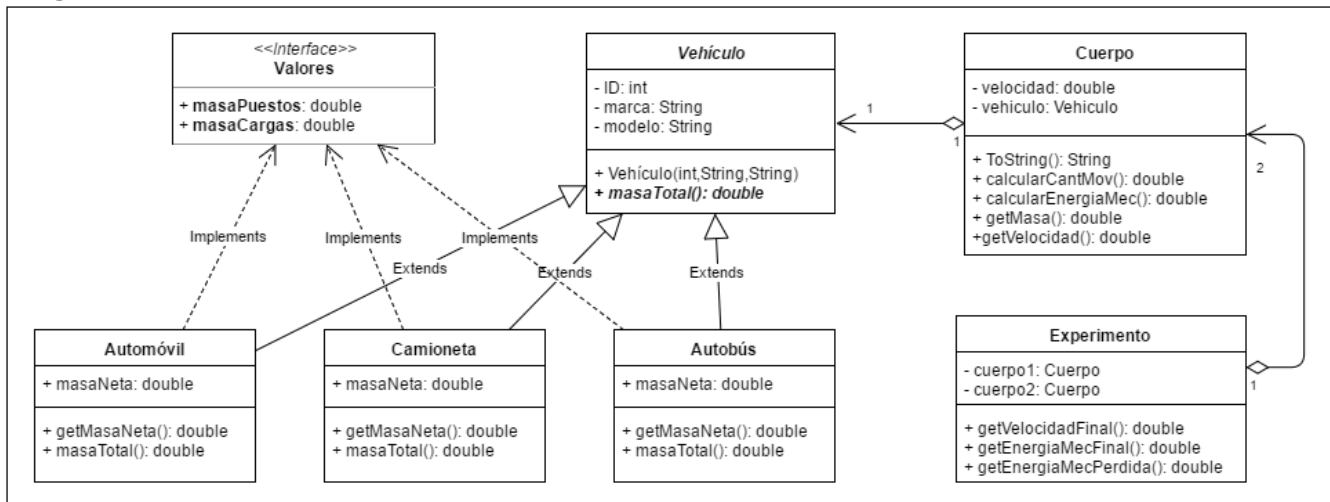
Formato: <Tipo vehículo> - <Marca> - <Modelo> - <Masa en Toneladas> - <ID>
 Tipo vehículo: 1=Automóvil, 2=Camioneta, 3=Autobús.

experimentos.txt

01-50;02-0
 02-80;03-20
 04-120;05-100
 06-90;01-10
 07-50;08-20
 09-60;06-10
 10-130;01-0
 11-100;02-10
 03-50;07-10
 05-75;04-60
 08-48;09-0

Formato: <ID₁> - <Velocidad₁ en Km/h> ; <ID₂> - <Velocidad₂ en Km/h>

Diagrama de Clases:



Baremo de Corrección:

A	Estructura de las clases Base-Hijas-Interface	5	Tiempo de duración: 3 ½ horas. Puede utilizar cualquier material de apoyo, siempre y cuando no sea en formato digital. ES SU RESPONSABILIDAD GUARDAR CORRECTAMENTE EL EXAMEN. EXÁMENES MAL GUARDADOS TIENE 0 PUNTOS
B	Clase Cuerpo: métodos toString, calcularCantMov y calcularEergiaMec	7	
C	Clase Experimento con sus métodos	10	
D	Separar los datos de los vehículos y cargarlos en el vector de objetos polimórfico	12	
E	Separar los datos de los experimentos y cargarlos en el objeto	12	
F	Validar los datos de los vehículos en el experimento	6	
G	Ejecutar experimentos	10	
H	Datos correctos	8	
Total		70	