



UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA
VICERRECTORADO ACADÉMICO
DECANATO DE DOCENCIA
DEPARTAMENTO DE INFORMÁTICA
ASIGNATURA: PROGRAMACIÓN I (0416202) LAPSO 2022-3

PRÁCTICA PRIMER EXAMEN PARCIAL | VALOR: 30 PUNTOS | FECHA: 27/11/2022 4:30 pm

Tiempo de duración: 4 horas.

Para ser desarrollado en pareja ó individual

Fecha de entrega 28/Noviembre/2022 hasta las 10pm

ES SU RESPONSABILIDAD COMPRIMIR EL PROYECTO

AL ARCHIVO COMPRIMIDO IDENTIFICARLO CON SUS NOMBRES, APELLIDOS Y SECCIÓN
Y ALOJAR CORRECTAMENTE EL EXAMEN EN EL AULA VIRTUAL.

Control de Vuelos.

El aeropuerto internacional Simón Bolívar desea llevar un registro y control de las personas que viajan en cada vuelo. Se desea registrar datos personales, categorías y datos del vuelo. Para llevar a cabo esta tarea debe realizar lo siguiente:

Una clase llamada **Vuelos**, que se encuentre en un paquete denominado **Aerolinea**, que contiene lo siguiente:

- Un atributo privado **nombre** de tipo String, que almacena el nombre de la aerolínea
- Un atributo privado **vuelo** de tipo int, que guarda el número del vuelo
- Un atributo privado **destino** de tipo String, que almacena la ciudad destino
- Un constructor paramétrico que recibe los datos para inicializar todos sus atributos
- Un método llamado **mostrar** que no recibe y retorna una cadena que contiene los datos del vuelo de la siguiente forma: “Aerolínea – Número del vuelo – Destino”
- Los métodos getter para sus atributos.
- No contiene métodos setter.

Una clase llamada **Personas**, que se encuentre en el paquete **Aerolinea**, que contenga lo siguiente:

- Un atributo privado **id** de tipo int, que almacena el N° de identificación de la persona
- Un atributo privado **nombre** de tipo String, que almacena el nombre de la persona
- Un atributo privado **edad** de tipo int, que guarda la edad de la persona
- Un atributo privado **vuelo** de tipo int, que guarda el número del vuelo
- Un atributo protegido **categoría** de tipo String, que almacena la categoría de la persona
- Un constructor paramétrico que recibe los datos para inicializar todos sus atributos
- Un constructor por defecto
- Un método llamado **mostrar** que no recibe ni retorna ningún valor
- Un método llamado **calcularCategoría** que no recibe ni retorna valores. Este método será implementado en las clases derivadas de la siguiente manera:
 - Para los pasajeros: si el valor del pasaje es menor a 450 Bs su categoría es “Económica” y si es superior es “1era Clase”. Si es igual a 450Bs su categoría es “Turista”

- Para las azafatas: si su edad <22 y habla máximo 2 idiomas es “Aprendiz”, si su edad es >=22 y habla máximo 2 idiomas es “Auxiliar”, si habla más de 2 idiomas es “Titular”
- Para el piloto: si tiene más de 1500 horas de vuelo es “Capitán”, si tiene 1500 horas de vuelo o menos es “1er Piloto”
- Los métodos getter para sus atributos.
- No contiene métodos setter.

Una clase llamada **Datos** que se encuentra en el paquete “**data**” que contiene lo siguiente:

- Un atributo público llamado **datos_vuelos** que es un vector de tipo String que contiene "Conviasa;763;SC", "Conviasa;125;Porlamar", "Aeropostale;815;Porlamar", "Conviasa;805;VAL", "Aserca;725;MBO". Este vector almacena la información de los vuelos en forma de cadena, y los atributos se encuentran delimitados por un “;”, el primer valor corresponde al nombre de la *aerolínea*, el segundo al *número del vuelo* y el tercero indica la *ciudad destino*.
Para separar la cadena de caracteres puede usar el método *split* o la clase *StringTokenizer*
- Un atributo público llamado **datos_persona** que es un vector de tipo String que contiene {"1;Jose Gonzalez;23;763;1;23A;550", "2;María Ramírez;19;125;2;1.65;3", "3;José Montoya;29;725;3;1350", "4;Gerardo Jaimes;25;805;3;2345", "5;Ana Pérez;36;815;2;1.72;4", "6;Emiro Martinez;33;815;1;5A;700", "7;Anabella Martinez;18;763;1;14C;320", "8;Mercedes Delgado;58;763;1;18D;440", "9;Diego Herrera;25;125;3;2445", "10;Cluadia Fernandez;47;763;1;6E;350"}. Este vector almacena la información de los viajeros en forma de cadena, y los atributos se encuentran delimitados por un “;”, el primer valor corresponde al *id*, el segundo al *nombre*, el tercero indica la *edad*, el cuarto al *número del vuelo*, el quinto es un código que indica el tipo de persona (*1 para Pasajero, 2 para Azafata y 3 para Piloto*):
 - En el caso de los pasajeros: el sexto corresponde a *número de asiento* y el séptimo al *valor del pasaje*
 - En el caso de las azafatas: el sexto corresponde a *estatura* y el séptimo a la *cantidad de idiomas*
 - En el caso de los pilotos: el sexto corresponde a *horas de vuelo*

Una clase llamada **Pasajero** que se encuentra en el paquete **Aerolinea** y debe heredar de Personas, y contiene lo siguiente:

- Un atributo privado **numAsiento** de tipo String, que almacena el N° de identificación del asiento
- Un atributo privado **valorPasaje** de tipo int, que almacena costo en Bolívares del pasaje
- Un constructor paramétrico que recibe los datos para inicializar todos los atributos (propios y heredados)
- Un constructor por defecto
- Los métodos getter para sus atributos.
- No contiene métodos setter.

- Sobreescriba el método **mostrar** para que imprima por consola los datos como se presentan a continuación:
 “Nombre: **nombre** Edad: **edad** Categoría: **categoria**”
 “Asiento: **numAsiento** Valor: **valorPasaje**”
- Sobreescriba el método **calcularCategoría** para que identifique la categoría del Pasajero y la asigne al atributo **categoría**

Una clase llamada **Azafata** que se encuentra en el paquete **Aerolinea** que debe heredar de **Personas**, y contiene lo siguiente:

- Un atributo privado **altura** de tipo double, que almacena la estatura de la azafata
- Un atributo privado **idiomas** de tipo int, que cantidad de idiomas que habla la azafata
- Un constructor paramétrico que recibe los datos para inicializar todos sus atributos (propios y heredados)
- Los métodos getter para sus atributos.
- No contiene métodos setter.
- Sobreescriba el método **mostrar** para que imprima por consola los datos como se presentan a continuación:
 “Nombre: **nombre** Edad: **edad** Categoría: **categoria**”
 “Estatura: **altura** Idiomas: **idiomas**”
- Sobreescriba el método **calcularCategoría** para que identifique la categoría de la Azafata y la asigne al atributo **categoría**

Una clase llamada **Piloto** que se encuentra en el paquete **Aerolinea** que debe heredar de **Personas**, y contiene lo siguiente:

- Un atributo privado **horas** de tipo int, que almacena el N° de horas de vuelo
- Un constructor paramétrico que recibe los datos para inicializar todos sus atributos (propios y heredados)
- Los métodos getter para sus atributos.
- No contiene métodos setter.
- Sobreescriba el método **mostrar** para que imprima por consola los datos como se presentan a continuación:
 “Nombre: **nombre** Edad: **edad** Categoría: **categoria**”
 “Horas de vuelo: **horas**”
- Sobreescriba el método **calcularCategoría** para que identifique la categoría del Piloto y la asigne al atributo **categoría**

Una clase llamada **Principal** en el paquete “**Implementacion**” que contenga lo siguiente:

- Un atributo **datos** de tipo **Datos**, para poder acceder a los **datos_vuelos** y **datos_persona**.
- Un atributo **viajes** de tipo de **Vuelo**, donde se almacenarán los datos de los vuelos.
- Tres vectores, uno para pasajeros, otro para azafatas y otro para pilotos.
- Un método llamado **cargarDatos** que no recibe ni retorna valores, el cual es el encargado de separar los datos de los vuelos y personas de los vectores **datos_vuelos**, **datos_personas** respectivamente y almacenarlos en los vectores correspondientes.
- Un método llamado **mostrarCategoría** que no recibe ni retorna valores. Este método muestra organizadamente la información de los pasajeros, azafatas y pilotos, debe indicar el valor de los atributos. Ejemplo:

Pasajeros

Nombre: Jose Gonzalez Edad: 23 Categoría: 1era Clase

Asiento: 23A Valor: 550

...

Azafatas

Nombre: Edad: Categoría:

Estatura: Idiomas:

...

Pilotos

Nombre: Edad: Categoría:

Horas de vuelo:

...

- Un método llamado **destinoFavorito** que no recibe ni retorna valores. Este método muestra el destino que presente mayor número de pasajeros e indica el monto en Bs recaudado en pasajes.
- Un método llamado **mejorPiloto** que no recibe ni retorna valores. Este método muestra el nombre del Piloto que tiene mayor número de horas de vuelo registradas.
- El método **main** encargado de llamar a los métodos de una instancia de la clase principal.

Nota: Los vectores **datos_vuelos** y **datos_persona** deben ser usados solo para cargar la información en el vector de objetos de viajes y de viajeros y no deben ser usados para realizar ningún otro cálculo.

A	Organización correcta de paquetes	1	Tiempo de duración: 4 horas. ES SU RESPONSABILIDAD GUARDAR CORRECTAMENTE EL EXAMEN. EXAMENES MAL GUARDADOS TIENEN 0 PUNTOS
B	Herencia e implementación correcta	1	
C	Clase Pasajero	2	
D	Clase Azafata	2	
E	Clase Piloto	2	
F	Clase Principal – método cargarDatos	2	
G	Clase Principal – mostrarCategoría	6	
H	Clase Principal – destinoFavorito	6	
I	Clase Principal – mejorPiloto	6	
J	Resultados correctos	2	
		Total	30