

EXAMEN PARCIAL 2

Leé con cuidado el enunciado y por lo menos dos veces para resolver lo pedido. Pensá bien la estrategia de resolución antes de comenzar el desarrollo de lo que te solicitan. El objetivo de este examen es **evaluar la correcta aplicación de los conceptos y técnicas** vistos hasta el momento:

- Correcta implementación de constructores.
- Modularización reutilizable y mantenible con uso de métodos con correcta parametrización y correcto encapsulamiento, publicando *setters* y *getters* sólo cuando corresponda.
- Manejo de clases, enumerados y colecciones.
- Importación y Exportación de proyectos Java desde Eclipse.

Antes de comenzar creá un proyecto JAVA cuyo nombre cumpla con el siguiente formato:

THP-2P-SEDE-CURSO-APELLIDO-Nombre

Ejemplos:

THP-2P-BE-11A-GONZALEZ-Benito

THP-2P-YA-11A-HARA-Jaime

Este proyecto deberá ser exportado al entregar con este nombre. Tené en cuenta que esto también forma parte del examen y se califica.

NOTA IMPORTANTE:

- No te quedes con dudas, preguntá lo que haga falta.
- Al entregar el examen asegurate de finalizar la entrega. y confirmá la recepción con tu profesor.
- **No se aceptarán exámenes por fuera de la página de entrega oficial de la plataforma.**

Enunciado

Nos solicitan desarrollar un programa que permita gestionar las reservas de los asientos en un vuelo. Del **Vuelo** conocemos los siguientes atributos: número de vuelo (String), cantidad total de asientos que posee, la lista de asientos reservados y el monto total recaudado, que en un inicio será 0.

En cuanto al **Asiento**, sabemos que tiene un pasajero (de tipo Pasajero), una clase (Enumerado, de valores PRIMERA, BUSINESS, TURISTA) y un precio. Los datos de un pasajero no pueden estar asignados a más de un asiento.

El **Pasajero** cuenta con los siguientes atributos: DNI y teléfono. Ambos enteros.

Se nos pide crear lo siguiente en donde corresponda:

- Constructor de Vuelo: Recibe el número de vuelo y la cantidad de asientos. El monto recaudado en un inicio debe ser 0.
 - Método público `reservarAsiento()`: Dados DNI, teléfono y clase, reservar asiento agregándolo a la lista y teniendo en cuenta la actualización del monto total recaudado.
- Observación: si se cumple alguno de los siguientes casos, el asiento no se agregará a la lista, retornando el respectivo estado con un enumerado `ResultadoReservaBoletos` que refleje estas posibilidades:

- "Error avión completo".
- "Error ya tiene un pasaje".

En el caso exitoso, es decir, cuando no se cumpla alguna de estas dos condiciones devolver un valor que signifique "Reserva confirmada".

- Método privado `buscarAsientoReservado()`: Recibe un DNI y, en el caso de existir en la lista de asientos reservados, retorna el asiento correspondiente. De lo contrario retorna NULL.
- Método público `listarAsientosReservados()`: Muestra por pantalla la cantidad de asientos reservados y un detalle con los datos de cada uno.

EXAMEN PARCIAL 2

- Método público `anularReserva()`: Dado un DNI recibido por parámetro, quita el asiento de la lista si encuentra al pasajero. Muestra por pantalla "Se ha anulado la reserva con éxito" o "No existe una reserva con ese DNI", según corresponda.
- Método privado `obtenerAcumuladoPorClase()` : Recibe la clase y, en caso de haber, retorna el monto acumulado de los asientos reservados para la misma
- Método público `mostrarMontoRecaudadoDeClase()` : Recibe la clase y muestra los siguientes mensajes según corresponda:
 - ☐ El monto acumulado hasta el momento en la clase _____ es: \$ _____
 - ☐ No hay asientos reservados para clase _____
 - ☐ No hay asientos reservados en ninguna clase
- Método privado `setPrecioSegunClase` en la clase Asiento: setea el precio según la clase (turista, business o primera) teniendo en cuenta que en caso de ser de clase Turista, el precio es de \$50.000, Business \$100.000 y Primera \$150.000.

Implementar los getters y setters necesarios y el método toString de cada una de las clases.

Crear una instancia de la clase Vuelo, con número de vuelo "AA3313" y 180 asientos, en la función main() de la clase Test respetando el siguiente escenario:

1. Reservar asiento con los siguientes datos:
 - DNI: 11111111
 - Teléfono: 1234567890
 - Clase: PRIMERA
2. Reservar asiento con los siguientes datos:
 - DNI: 22222222
 - Teléfono: 1234567891
 - Clase: PRIMERA
3. Reservar asiento con los siguientes datos:
 - DNI: 33333333
 - Teléfono: 1234567892
 - Clase: TURISTA
4. Reservar asiento con los siguientes datos:
 - DNI: 44444444
 - Teléfono: 1234567894
 - Clase: TURISTA
5. Reservar asiento con los siguientes datos:
 - DNI: 55555555
 - Teléfono: 1234567895
 - Clase: BUSINESS
6. Reservar asiento con los siguientes datos:
 - DNI: 11111111
 - Teléfono: 1234567890
 - Clase: PRIMERA
7. Mostrar el monto recaudado de PRIMERA CLASE
8. Mostrar el monto recaudado de BUSINESS
9. Mostrar el monto recaudado de TURISTA
10. Listar los asientos reservados
11. Anular reserva con DNI 55555555
12. Anular reserva con DNI 99999999

EXAMEN PARCIAL 2

13. Mostrar el monto recaudado de BUSINESS
14. Listar los asientos reservados

La ejecución del programa debería mostrar:

```
RESERVA_CONFIRMADA
RESERVA_CONFIRMADA
RESERVA_CONFIRMADA
RESERVA_CONFIRMADA
RESERVA_CONFIRMADA
ERROR_YA_TIENE_PASAJE
El monto acumulado hasta el momento en PRIMERA es: $300000.0
El monto acumulado hasta el momento en BUSINESS es: $100000.0
El monto acumulado hasta el momento en TURISTA es: $100000.0
Cantidad de asientos reservados hasta el momento: 5
Boleto [pasajero=Pasajero [dni=11111111, telefono=1234567890], clase=PRIMERA, precio=150000.0]
Boleto [pasajero=Pasajero [dni=22222222, telefono=1234567891], clase=PRIMERA, precio=150000.0]
Boleto [pasajero=Pasajero [dni=33333333, telefono=1234567893], clase=TURISTA, precio=50000.0]
Boleto [pasajero=Pasajero [dni=44444444, telefono=1234567894], clase=TURISTA, precio=50000.0]
Boleto [pasajero=Pasajero [dni=55555555, telefono=1234567895], clase=BUSINESS, precio=100000.0]
Se ha anulado la reserva con éxito
No existe una reserva con ese dni
No hay asientos reservados para clase BUSINESS
Cantidad de asientos reservados hasta el momento: 4
Boleto [pasajero=Pasajero [dni=11111111, telefono=1234567890], clase=PRIMERA, precio=150000.0]
Boleto [pasajero=Pasajero [dni=22222222, telefono=1234567891], clase=PRIMERA, precio=150000.0]
Boleto [pasajero=Pasajero [dni=33333333, telefono=1234567893], clase=TURISTA, precio=50000.0]
Boleto [pasajero=Pasajero [dni=44444444, telefono=1234567894], clase=TURISTA, precio=50000.0]
```

Al finalizar el examen subí un txt que contenga el código de todos los .java que creaste y exportá el proyecto tal como se pide al principio. Luego subí ambos archivos al Aula Virtual como entrega del examen. Finalizó el examen y antes de cerrar la página confirmá la entrega con tu profesor.