

Materia: TALLER DE HERRAMIENTAS DE PROGRAMACIÓN
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 con tu nombre y apellido

Enunciado

Nos solicitan el desarrollo de 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), un precio, su fila y su número (ubicación del asiento). Respecto de las ubicaciones las filas van de la A hasta la T y en cada fila los asientos se numeran del 1 hasta el 6.

Aclaración: para que sea más práctico, la nomenclatura de asientos de nuestro avión es al revés de los aviones verdaderos: filas con letras y columnas con número.

En este momento especial que atraviesa la sociedad las aerolíneas toman medidas de cuidado. Una de ellas es que entre cada pasajero debe haber un asiento intermedio libre.

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, clase, fila y letra, 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".
 - "Error el asiento de al lado está ocupado" (**para cada asiento debe verificarse que el asiento de al lado de la misma fila no esté ocupado**).

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 una fila y un número de asiento y, en el caso de existir en la lista de asientos reservados, retorna el asiento correspondiente. De lo contrario retorna NULL.
- Método privado **verDistanciamiento()**: Recibe una fila y un número de asiento y debe verificar que los asientos que están a la izquierda y a la derecha del que está en la ubicación recibida estén libres.

EXAMEN PARCIAL 2

- 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
 - Fila A
 - Número 1
2. Reservar asiento con los siguientes datos:
 - DNI: 22222222
 - Teléfono: 1234567891
 - Clase: TURISTA
 - Fila F
 - Número 6
3. Reservar asiento con los siguientes datos:
 - DNI: 33333333
 - Teléfono: 1234567892
 - Clase: PRIMERA
 - Fila A
 - Número 2
4. Reservar asiento con los siguientes datos:
 - DNI: 44444444
 - Teléfono: 1234567894
 - Clase: TURISTA
 - Fila F
 - Número 5
5. Reservar asiento con los siguientes datos:
 - DNI: 55555555
 - Teléfono: 1234567895
 - Clase: PRIMERA
 - Fila A
 - Número 1
6. Mostrar el monto recaudado de PRIMERA CLASE
7. Mostrar el monto recaudado de BUSINESS
8. Mostrar el monto recaudado de TURISTA

La ejecución del programa debería mostrar:

```
RESERVA_CONFIRMADA
RESERVA_CONFIRMADA
ERROR_EL_ASIENTO_DE_AL_LADO_ESTA_OCUPADO
ERROR_EL_ASIENTO_DE_AL_LADO_ESTA_OCUPADO
ERROR_YA_ESTA_VENDIDO
El monto acumulado hasta el momento en PRIMERA es: $150000.0
No hay asientos reservados para clase BUSINESS
El monto acumulado hasta el momento en TURISTA es: $50000.0
```

Al finalizar el examen, subí un txt que contenga el código de todas los .java que creaste, y exportá el proyecto con el nombre: **THP-2P-ApellidoNombre.zip**. Luego subí ambos archivos en el aula virtual.

NOTA IMPORTANTE:

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