**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 **(Pasajero)**, tipos de **Asiento(Primera, Turista y Business)**, un precio, su fila (String) y su número (int)  (que indican la ubicación en el avión). 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, sin que haya un pasillo que los separe.

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 números.

**La aerolínea mantiene una medida pandémica para que los pasajeros viajen más cómodos, entre cada pasajero debe haber un asiento intermedio libre.**

El **Pasajero** cuenta con los siguientes atributos: DNI y teléfono. Ambos enteros. Los datos de un pasajero no pueden estar asignados a más de un asiento.

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 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 no estén reservados. *(Ayudita: hay que chequear en la misma fila, y una columna anterior para ver el de la izquierda, y una columna posterior para ver el asiento que está a la derecha. Tené en cuenta que a la izquierda de la columna 1 no hay asiento, y a la derecha de la columna 6 tampoco)*
* 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 **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 de ser necesarios. Además implementar una interfaz **Calculable** en la clase Vuelo.

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

1. Reservar asiento con los siguientes datos:

* DNI: 22222222
* Teléfono: 1234567891
* Clase: TURISTA
* Fila F
* Número 6

1. Reservar asiento con los siguientes datos:

* DNI: 33333333
* Teléfono: 1234567892
* Clase: PRIMERA
* Fila A
* Número 2

1. Reservar asiento con los siguientes datos:

* DNI: 44444444
* Teléfono: 1234567894
* Clase: TURISTA
* Fila F
* Número 5

1. Reservar asiento con los siguientes datos:

* DNI: 55555555
* Teléfono: 1234567895
* Clase: PRIMERA
* Fila A
* Número 1
* Mostrar el monto recaudado de PRIMERA CLASE
* Mostrar el monto recaudado de BUSINESS
* Mostrar el monto recaudado de TURISTA
* Mostrar el monto total recaudado por todas las clases.

La ejecución del programa debería mostrar:

