

CURSO 2024-25.

Práctica 3. Tema 4.

1. OBJETIVOS:

- Manejo de clases y objetos.

2. Enunciado

Escribir un programa en java que permita gestionar la reserva de billetes de avión de una pequeña aerolínea que opera sus vuelos entre Madrid y el Caribe. El programa permitirá mediante el menú correspondiente las siguientes opciones:

1. Inicializar aviones y vuelos
2. Reservar asiento
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar

El programa descrito deberá tener en cuenta las siguientes características.

- Aplicación *GestionAerolinea*: La aplicación se encuentra en fase de pruebas y solo gestionará 3 aviones que realizarán 3 vuelos en un día concreto.
- *Aviones*. Todos los aviones tienen el mismo número de asientos por fila (4). Cada avión tiene un número de filas diferente para la clase o categoría *business* y un mayor número para la *turista*. Viajando en *business* se obtienen más servicios y comodidades que en *turista* y por tanto es una clase mucho más cara.
- *Vuelos*. Cada vuelo está asociado a un único avión en una única fecha. Todos tienen origen en Madrid pero con destinos diferentes (La Habana, Cancún y Punta Cana).
- Un *Pasajero* solo podrá reservar un *asiento* en un vuelo si queda alguno libre en alguna de las dos categorías. Podrá llevar consigo sin coste adicional una *maleta*.
- Se proporciona una clase de apoyo denominada *Azar* que sirve para generar aleatoriamente una serie de elementos.
- Todas las clases necesarias serán descritas más adelante.

En el proyecto existen las siguientes clases ya creadas:

- El tipo enumerado `enum Clase` puede tomar dos valores para indicar la categoría del asiento: BUSINESS o TURISTA. Está incluido junto a la clase `Avion`.
- `GestionAerolinea`, clase por donde comienza a ejecutarse el programa y que mostrara el menú de la gestión de la aerolínea. Esta clase viene implementada con los siguientes elementos:

```
private final int PRECIO_BILLETE_TURISTA = 350;
private final int PRECIO_BILLETE_BUSINESS = 1500;
private final float DESCUENTO_INFANTIL = 15f;
```



UNIVERSIDAD POLITÉCNICA DE MADRID
ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS
Departamento de Sistemas Informáticos
Asignatura: Fundamentos de Programación
Ingeniería del Software y Sistemas de Información



```
private final int NUM_AVIONES = 3;
private final int NUM_VUELOS = NUM_AVIONES;
private Avion[] aviones, lista de aviones de la aerolínea.
private Vuelo[] vuelos, lista de vuelos de la aerolínea.
```

- Métodos ya implementados por completo que no hay que tocar.

```
public GestionAerolinea(): constructor de la clase.
public static void main(String[] args).
public void iniciarAvionesYVuelos():
método que inicializa las listas de aviones y de vuelos.
public int leerNumero(int minimo, int maximo, String texto):
método que me permite pedir un número y valida que este entre un límite mínimo y uno
máximo pasados por parámetro, también permite poner un texto en pantalla indicando qué
tiene que hacer el usuario.
public Vuelo preguntarVuelo():
método que pregunta el número del destino del vuelo, 0: La Habana, 1: Cancún, 2: Punta Cana.
Devuelve el vuelo seleccionado.
public Clase preguntarClase():
método que pregunta la clase del asiento, 0: Business, 1: Turista. Devuelve Clase.BUSINESS
o Clase.TURISTA, según la elección.
```

- Métodos por completar:

```
public void ejecutarOpcion(int opcion), este método ejecutará la opción del
menú elegida por el usuario en el método menu().
public int menu(), método que muestra el menú y recoge la opción elegida por el usuario.
public void reservarAsiento(Avion avion, Clase clase), este método buscará
un asiento disponible en el avión y en la clase pasados como argumentos. Si hay hueco, le
asignará un pasajero al azar con Azar.generaPasajero() al primer asiento libre que
encuentre y mostrará el mensaje de texto:
Reservado el asiento <asiento> de clase <clase> al pasajero <nombre>
Por ejemplo,
Reservado el asiento 1A de clase turista al pasajero Ana Perez Gomez
Si no lo encuentra, mostrará el mensaje:
No hay asientos libres en clase <clase>
public void mostrarPasajeros(Avion avion), este método mostrará los datos de
los pasajeros en el avión pasado por argumento organizados por clase y por filas. En la primera
línea debe poner "Avión " seguido del modelo al que corresponde. A continuación, muestra
los pasajeros de la clase business con el número de fila y todos los datos del pasajero. Y se
repite para los de la clase turista. Si hay algún asiento libre, sólo mostrará el texto "(libre)".
public void mostrarPasajeros(Avion avion, int edad), este método mostrará
los datos de los pasajeros menores de la edad pasada por argumento en el avión indicado
organizados por clase. En la primera línea debe poner "Avión " seguido del modelo al que
corresponde. A continuación, muestra los pasajeros de la clase business sin el número de fila
y el nombre y edad del pasajero menor de edad. Y se repite para los de la clase turista.
public void mostrarIngresos(Avion avion), este método mostrará los ingresos
que ocasionan los billetes de los pasajeros teniendo en cuenta el precio de cada clase y el
descuento infantil.
```

Por ejemplo,

El avión Airbus A310 proporciona ingresos de 36220.0 €



UNIVERSIDAD POLITÉCNICA DE MADRID
ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS
Departamento de Sistemas Informáticos
Asignatura: Fundamentos de Programación
Ingeniería del Software y Sistemas de Información



- **Vuelo**, clase que contiene los datos de un vuelo. Esta clase está por completar con los siguientes elementos:
 - Propiedades

```
private String paisOrigen
private String paisDestino
private String fecha
private Avion avion
```
 - Métodos a completar

```
public Vuelo(String origen, String destino, String fecha, Avion avion),
es el constructor de la clase.
public String getPaisOrigen()
public String getPaisDestino()
public String getFecha()
public Avion getAvion()
```
- **Avion**, clase ya implementada que proporciona los siguientes elementos:
 - Propiedades

```
private final int ASIENTOS_X_FILA = 4, indica el número de asientos por fila.
private String modelo, es el modelo del avión.
private int numeroAsientosBusiness, es el número de asientos totales en clase
business.
private Asiento[][] asientoBusiness, es una matriz que guarda la información de
cada asiento de la clase business.
private int numeroAsientosTurista, es el número de asientos totales en clase turista.
private Asiento[][] asientoTurista, es una matriz que guarda la información de
cada asiento de la clase turista.
```
 - Métodos

```
public Avion(String modelo, int business, int turista), es el constructor de
la clase.
public String getModelo()
public int getNumeroFilas(Class clase) , devuelve el número de filas de la clase
pasada por argumento.
public int getButacasPorFila(), devuelve el número de butacas por fila.
public Asiento reservarAsiento(int fila, int butaca, Class clase,
Pasajero pasajero), reserva un asiento en la posición (fila,butaca) de la clase indicada
para el pasajero pasado por argumento. Devuelve el asiento creado, si hay hueco; null, si no
lo hay.
public void mostrarMapaDeAsientos(), muestra el estado de reserva de los asientos
del avión completo de la siguiente manera:
```
 - La primera línea mostrará el texto "Avión " seguido del modelo.
 - Cada línea de las siguientes mostrará por columnas cada asiento de la zona business y de la zona turista con los caracteres 'B', 'T' o '.', según corresponda. Los asientos libres se mostrarán con un punto '.' y los ocupados con una 'B' o una 'T' dependiendo de la zona en la que estén.
 - Entre ambas zonas debe haber un espacio de separación.

Por ejemplo, para un avión modelo Airbus A300 con un total de 20 asientos de clase business y 56 de clase turista con muchos asientos libres se mostrará de la siguiente manera:



UNIVERSIDAD POLITÉCNICA DE MADRID
ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS
Departamento de Sistemas Informáticos
Asignatura: Fundamentos de Programación
Ingeniería del Software y Sistemas de Información



```
Avión Airbus A300
BBB... TTTT.....
BBB... TTTT.....
BB...  TTTT.....
BB...  TTTT.....
```

- **Asiento**, clase que representa un asiento del avión.
 - Propiedades:
 - `private int fila`, es el número de fila del asiento.
 - `private int butaca`, es el número de butaca del asiento.
 - `private Pasajero pasajero`, es el pasajero que se ocupa el asiento.
 - Métodos a implementar:
 - `public Asiento(int fila, int butaca, Pasajero pasajero)`, es el constructor de la clase.
 - `public int getFila()`
 - `public int getButaca()`
 - `public Pasajero getPasajero()`
 - `public String toString()`, devuelve un String formado por un número y una letra. El número es el de la fila y la letra el de la butaca haciendo la siguiente correspondencia:
0 A, 1 B, 2 C, 3 D.
Por ejemplo, el asiento de la fila 21, butaca 2 se corresponderá con `21C`.
- **Pasajero**, clase que representa a un pasajero.
 - Propiedades:
 - `private String nombre`, es el nombre del pasajero.
 - `private String pasaporte`, es el número de pasaporte.
 - `private String telefono`, es el número de telefono.
 - `private int edad`, es la edad en años.
 - `private Maleta maleta`, es una referencia a la maleta que lleva el pasajero.
 - Métodos a implementar:
 - `public Pasajero(String nombre, String pasaporte, String teléfono, int edad, Maleta maleta)`, es el constructor de la clase.
 - `public String getNombre()`
 - `public String getPasaporte()`
 - `public String getTelefono()`
 - `public int getEdad()`
 - `public Maleta getMaleta()`
- **Maleta**, clase que guarda las medidas y peso de una maleta.
 - Propiedades:
 - `private final double PESO_MAXIMO`, constante para el peso máximo permitido.
 - `private double peso`, es el peso de la maleta.
 - `private final String MEDIDA_TOTAL_MAXIMA`, constante de la medida total máxima permitida.
 - `private int ancho`, es el ancho de la maleta.
 - `private int alto`, es el alto de la maleta.
 - `private int fondo`, es el fondo de la maleta.
 - Métodos a implementar:

`public` Maleta(`double` peso, `int` ancho, `int` alto, `int` fondo), es el constructor de la clase.

`public double` getPeso(), devuelve el peso.

`public int` getMedidatotal(), devuelve la suma del ancho, alto y fondo.

`public boolean` excedeDePeso(), devuelve `true`, si el peso excede del peso máximo permitido; `false`, en caso contrario.

`public boolean` excedeDeMedidas(), devuelve `true` si las medidas totales exceden del máximo permitido; `false`, en caso contrario.

- **Azar**, clase ya implementada que proporciona los siguientes elementos:

`private static String` generaNombre(), devuelve un string formado por un nombre y dos apellidos generados al azar a partir de varias listas.

`private static String` generaNum(`int` longitud), devuelve un String numérico con la longitud indicada.

`private static String` generaPasaporte(`int` longitud), genera al azar un String con el código del pasaporte con el siguiente formato: 3 letras seguidas de 6 números.

`private static String` generaTelefono(`int` longitud), genera al azar un número de teléfono de 9 cifras que empieza siempre por 6.

`private static int` generaEdad(), devuelve un número aleatorio entre 1 y 99.

`private static Maleta` generaMaleta(), crea una maleta y genera sus datos aleatoriamente.

`public static Pasajero` generaPasajero(), genera un pasajero con datos aleatorios.

El programa tendrá la siguiente funcionalidad.

1. Inicializar aviones y vuelos. Crea los aviones y los vuelos con datos predefinidos. Muestra el mensaje "Aviones y vuelos inicializados".
2. Reservar asiento de un vuelo. Primero pregunta en qué vuelo se reservará y luego pregunta en qué clase (business o turista) se quiere el asiento. A continuación, llama al método estático `reservarAsiento()` para reservar un asiento en la clase y vuelo leídos.
3. Mostrar el mapa de asientos. Pregunta por el vuelo para saber cuál es el avión del que se quiere mostrar su mapa de asientos.
4. Mostrar la lista de pasajeros. Primero pregunta por el vuelo a mostrar y luego llama al método estático `mostrarPasajeros(Avion)`. Debe mostrar la salida tal y como se describe en la clase `GestionAerolinea` y en los ejemplos de ejecución.
5. Mostrar pasajeros menores de 15. Primero pregunta por el vuelo a mostrar y luego llama al método estático `mostrarPasajeros(Avion, int)`. Debe mostrar la salida tal y como se describe en la clase `GestionAerolinea` y en los ejemplos de ejecución.
6. Mostrar ingresos del vuelo. Primero pregunta por el vuelo a mostrar y luego llama al método estático `mostrarIngresos()`. Debe mostrar la salida tal y como se describe en la clase `GestionAerolinea` y en los ejemplos de ejecución.
0. Salir de la aplicación. Muestra el mensaje "Fin de la ejecución."

Las opciones de la 2 a la 6 tienen que tener en cuenta que previamente se hayan inicializado los aviones y los vuelos, si no, mostrarán el siguiente mensaje "Error. Aviones y vuelos no inicializados".



3. Entrega de la práctica.

Se entregará el proyecto *IntelliJ* resultante de hacer la práctica "FP Practica3", conteniendo las clases creadas para solucionar la práctica 3.

Para entregarlo, se comprimirá el proyecto en un archivo ZIP, y se subirá al Moodle de la asignatura. El nombre de dicho archivo ZIP será el mismo, "FP Practica3.zip"

En la primera línea de la clase **GestionAerolinea** se debe indicar el nombre, apellidos y número de matrícula del alumno que realiza la práctica.

4. Ejemplos de Ejecución:

```
1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar
Introduce un número entre 0 y 6: 2
Error. Aviones y vuelos no inicializados.
```

```
1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar
Introduce un número entre 0 y 6: 1
Aviones y vuelos inicializados
```

```
1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar
Introduce un número entre 0 y 6: 2
Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 2
Elija la clase (0: Business, 1: Turista) 1
Reservado el asiento 26A de clase turista al pasajero Violeta Garcia Segovia
```

```
1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar
Introduce un número entre 0 y 6: 2
Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 0
Elija la clase (0: Business, 1: Turista) 0
No hay asientos libres en clase business
```



UNIVERSIDAD POLITÉCNICA DE MADRID
ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS
Departamento de Sistemas Informáticos
Asignatura: Fundamentos de Programación
Ingeniería del Software y Sistemas de Información



1. Inicializar aviones y vuelos
 2. Reservar asiento en un vuelo
 3. Mostrar el mapa de asientos
 4. Mostrar la lista de pasajeros
 5. Mostrar pasajeros menores de 15 años
 6. Mostrar ingresos del vuelo
 0. Finalizar
- Introduce un número entre 0 y 6: 3
Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 2

Avión Airbus A350

```
BBBBBBBBBBBBB TTTTTTTTTTTTTTTTTTTTTT.....  
BBBBBBBBBBBBB TTTTTTTTTTTTTTTTTTTTTT.....  
BBBBBBBBBBBBB TTTTTTTTTTTTTTTTTTTTTT.....  
BBBBBBBBBBBBB TTTTTTTTTTTTTTTTTTTTTT.....
```

1. Inicializar aviones y vuelos
 2. Reservar asiento en un vuelo
 3. Mostrar el mapa de asientos
 4. Mostrar la lista de pasajeros
 5. Mostrar pasajeros menores de 15 años
 6. Mostrar ingresos del vuelo
 0. Finalizar
- Introduce un número entre 0 y 6: 4
Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 2

Avión Airbus A350

Lista de pasajeros de la clase Business:

Fila 1

Soledad Fernandez Delgado	DVK097669	34 años
Maria Gomez Rodriguez	MIW615697	14 años
Esmeralda Cuenca Cuenca	FYW769975	16 años
Pilar Suarez Fernandez	EDJ596665	66 años

Fila 2

Adolfo Segovia Delgado	JVP517535	33 años
Miguel Segovia Suarez	JYD617547	12 años
Javier Lopez Diaz	EVS699479	55 años
Soledad Gomez Lopez	PFL439762	6 años

...

Lista de pasajeros de la clase Turista:

Fila 1

Esmeralda Fernandez Sevilla	EYD349248	41 años
Juan Gomez Castillo	KKM331347	72 años
Adolfo Fernandez Sevilla	RKD071829	44 años
Soledad Lopez Rodriguez	ULH136479	94 años

Fila 2

Violeta Rodriguez Rodriguez	ETQ530983	50 años
Jose Lopez Perez	MS0964048	34 años
Sonia Suarez Lopez	FGN900932	39 años
Susana Delgado Castillo	EEN661874	29 años

...

Fila 45

Javier Fernandez Madrid (libre) (libre) (libre)	ERT947237	53 años
--	-----------	---------



UNIVERSIDAD POLITÉCNICA DE MADRID
ETS INGENIERÍA DE SISTEMAS INFORMÁTICOS
Departamento de Sistemas Informáticos
Asignatura: Fundamentos de Programación
Ingeniería del Software y Sistemas de Información



1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar

Introduce un número entre 0 y 6: 5

Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 1

Avión Airbus A310

Lista de pasajeros menores de 15 años de la clase Business:

Jorge Perez Toledo	3 años
Sonia Delgado Cano	4 años
Sonia Martinez Arroyo	9 años

Lista de pasajeros menores de 15 años de la clase Turista:

Violeta Diaz Suarez	1 años
Soledad Cuenca Toledo	6 años
Susana Martinez Toledo	13 años
Rocio Garcia Delgado	2 años
Maria Rodriguez Gutierrez	14 años
Soledad Gutierrez Hernandez	13 años
Ana Gomez Cano	13 años
Susana Sevilla Martinez	7 años
Susana Fernandez Alameda	12 años
Susana Lugo Gutierrez	7 años

1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar

Introduce un número entre 0 y 6: 6

Elija el vuelo (0: La Habana, 1: Cancún, 2: Punta Cana) 2

El avión Airbus A350 proporciona ingresos de 103810.0 €

1. Inicializar aviones y vuelos
2. Reservar asiento en un vuelo
3. Mostrar el mapa de asientos
4. Mostrar la lista de pasajeros
5. Mostrar pasajeros menores de 15 años
6. Mostrar ingresos del vuelo
0. Finalizar

Introduce un número entre 0 y 6: 0

Fin de la ejecución.