

Nombre: _____

DNI: _____ **Fila:** _____ **Columna:** _____

Instrucciones

- El ejercicio a pie de máquina se entregará al terminar el examen. Se recogerán en pendrive por parte del profesor.
- NO está permitida la conexión a internet en el examen. Tampoco la utilización de tu propio portátil ni la consulta de apuntes o de otros códigos.

DeustoAir es una empresa encargada del embarque de pasajeros a aviones del aeropuerto de Loiu. Te han contratado para crear una aplicación que les ayude a gestionar sus embarques en la que tendrás que programar las siguientes clases y métodos:

Clase Maleta (1 puntos)

Esta clase gestiona las maletas de los pasajeros que embarcarán en cada vuelo y se almacenará en el fichero **maleta.py**.

Sus propiedades son:

- **codigo:** cadena de caracteres (string), por ejemplo "VL1131".
- **altura:** en centímetros, mayor o igual a cero.
- **anchura:** en centímetros, mayor o igual a cero.
- **largura:** en centímetros, mayor o igual a cero.
- **peso:** en kg, mayor o igual a cero.

Sus métodos son:

- Constructor **con argumentos y valores por defecto**.
- **Getters y setters**.
- Método **__str__** que devuelve un string con el código de la maleta, sus medidas y su peso: **"Maleta VL1131, 55x65x35 (12.3 kg)"**.

Tanto el constructor con argumentos como los setters de la clase Maleta deberán comprobar que son valores válidos los que se asignan a los atributos (consulta las indicaciones aportadas en cada uno de ellos previamente). Si no lo son, el constructor pondrá valores por defecto y los setters dejarán los valores que ya estaban.

Clase Pasajero (2 puntos)

Esta clase gestiona los pasajeros en DeustoAir y se almacenará en el fichero **pasajero.py**.

Sus propiedades son:

- **nombre:** cadena de caracteres (string), por ejemplo "Pablo".
- **apellido:** cadena de caracteres (string), por ejemplo "Garaizar".
- **talla:** entero entre 0 y 5 que indica la talla aproximada del pasajero.
- **menu:** cadena de caracteres (string) que puede tomar los siguientes valores "normal", "vegetariano", "celiaco".

- **cabina:** lista de objetos de la clase Maleta que almacena las maletas que lleva el pasajero como equipaje de mano (lista vacía si no lleva ninguna maleta de mano).
- **facturado:** lista de objetos de la clase Maleta que almacena las maletas que ha facturado el pasajero como equipaje en bodega (lista vacía si no ha facturado ninguna maleta).

Sus métodos son:

- Constructor **con argumentos y valores por defecto.**
- **Getters y setters.**
- Método **__str__** que devuelve un string con el nombre y apellido del pasajero y el peso total de su equipaje: **"Pablo Garaizar (43.5 kg)"**
- Método **getPesoTotal** que devuelve un float correspondiente al peso total de todo el equipaje de cabina y facturado del pasajero.

DeustoAir

Programa principal que se almacenará en el fichero **deustoair.py** y contendrá una lista de **pasajeros** vacía además de los siguientes métodos:

- **checkinOnline(pasajeros):** añade a la lista de pasajeros 5 objetos de la clase Pasajero que tengan entre 0 y 2 maletas en cabina o facturadas usando los constructores de las clases apropiadas **(0,5 puntos).**
- **volumenCabina(pasajeros):** devuelve un float con el volumen total en cm³ del equipaje de cabina de todos los pasajeros **(1 punto).**
- **checkinAeropuerto(pasajeros):** añade pasajeros a la lista de pasajeros mediante una interacción de preguntas y respuestas. En el caso de las maletas de cabina, solamente se permitirá añadir maletas si el volumen total en cm³ del equipaje de cabina de todos los pasajeros es menor que 120000. Consulta la página siguiente para ver un ejemplo de la interacción **(3,5 puntos).**
- **candidatoOverbooking(pasajeros):** devuelve el Pasajero que más peso total aporte al embarque del avión teniendo en cuenta el total de peso de su equipaje y su talla (0: 0 kg, 1: 20 kg, 2: 40 kg, 3: 60 kg, 4: 80 kg, 5: 100 kg). Por ejemplo, alguien de talla 4 con dos maletas de cabina de 5 kg cada una y otra facturada de 18.5 kg, pesaría $80+5+5+18.5 = 108.5$ **(2 puntos).**
- **resumenEmbarque(pasajeros):** imprime por pantalla un resumen del embarque que incluya el número de pasajeros, de sus menús, el peso del equipaje y el peso estimado de los pasajeros. Consulta la página siguiente para ver un ejemplo **(1 punto).**



El programa principal deberá contener algo similar a esto:

```
pasajeros = []

print('DeustoAir: de Bilbao al cielo.')

checkinOnline(pasajeros)
checkinAeropuerto(pasajeros)
overbooking = candidatoOverbooking(pasajeros)
print(f'El mejor candidato para proponerle una compensación por overbooking es {overbooking}')
resumenEmbarque(pasajeros)
```

Y deberá mostrar algo similar a esto:

DeustoAir: de Bilbao al cielo.

Cargando pasajeros online...

Registrando pasajeros en el aeropuerto...

```
¿Quieres añadir un pasajero? (s/n): s
Nombre: Borja
Apellido: Sanz
Talla (0-5): 4
Menú (normal/vegetariano/celiaco): normal
¿Quieres añadir equipaje de cabina? (s/n): s
Añadiendo nueva maleta...
Código: VL1113
Altura: 45.0
Anchura: 20.0
Largura: 40.5
Peso: 13.7
¿Quieres añadir equipaje de cabina? (s/n): n
¿Quieres añadir equipaje facturado? (s/n): s
Añadiendo nueva maleta...
Código: VL1114
Altura: 62.0
Anchura: 32.0
Largura: 60.5
Peso: 23.2
¿Quieres añadir equipaje facturado? (s/n): n
Pasajero añadido
```

```
¿Quieres añadir un pasajero? (s/n): n
```

El mejor candidato para proponerle una compensación por overbooking es Pablo Garaizar (43.5 kg)

Resumen del embarque:
126 pasajeros embarcados
101 menús normales
21 menús vegetarianos
4 menús celiacos
1477.5 kg de equipaje
7760 kg de pasajeros