

ASIGNATURA		Examen Global 2 Evaluación	
Número de curso: IFC302LOE			
Denominación: <b>TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA</b>			
Horas: 224	Modalidad: presencial	Fecha inicio: 13/09/2023	Fecha fin: 22/06/2021
Centro: AFA FORMACION CONTINUA			
Código y denominación del módulo formativo: <b>0485: Programación</b>			
Profesora: Mónica Alonso González			<b>CALIFICACIÓN</b>
Alumno/a:			
<b>Tiempo de realización:</b> 120 minutos			
<b>Descripción de la prueba:</b> <ul style="list-style-type: none"> <li>Examen practico</li> </ul>			
<b>Sistema de calificación:</b> <ul style="list-style-type: none"> <li><b>Cada parte evaluable viene identificada con su puntuación</b></li> <li><b>Se debe sacar un mínimo de 2 puntos sobre 5 en cada uno de los ejercicios para evaluar el examen</b></li> </ul>			
<b>Instrucciones:</b> <ul style="list-style-type: none"> <li><b>Desarrollar el enunciado sobre el proyecto suministrado en la tarea</b></li> </ul>			
Firma del Profesor:	Firma del Alumno/a:		Fecha:

## Ejercicio 1

La compañía Oceanic Airlines, debido a la escasez de pasajeros en su viejo 815 entre Sydney y Los ángeles (solo 6) ha puesto en oferta billetes de la clase turista para intentar completar el pasaje. El avión preparado para el viaje dispone de 42 asientos en la clase turista, distribuidos en 2 filas de 3 asientos cada una con el pasillo de por medio.

El software de la compañía representa el interior del avión con una matriz 7x7 con una fila intermedia para el pasillo marcada con dos \* en cada celda. Los asientos comienzan a numerarse descendentemente en columnas, tal y como muestra la figura:

1	7	13	19	25	31	37
2	8	14	20	26	32	38
3	9	15	21	27	33	39
**	**	**	**	**	**	**
4	10	16	22	28	34	40
5	11	17	23	29	35	41
6	12	18	24	30	36	42

Los 6 viajeros actuales son: Sylid Jarrah en el asiento 4, Kate Austen en el asiento 8, Sun Kwon en el 15, Hugo Reyes en el 16, Jack Shephard en el 23 y Ana lucia en el 42. . cuando un pasajero es asignado, se le colocan las iniciales en la posición correspondiente.

1	7	13	19	25	31	37
2	'KA'	14	20	26	32	38
3	9	'SK'	21	27	33	39
**	**	**	**	**	**	**
'SJ'	10	'HR'	22	28	34	40
5	11	17	'JS'	29	35	41
6	12	18	24	30	36	'AL'

Los nuevos pasajeros que aprovechan la oferta, no podrán solicitar asiento, por loque el software les asignará uno automáticamente. Por ejemplo, si el pasajero Benjamín Lopez comprara un billete, el sistema intenrá coloca las letras 'BL' en una de las posiciones vacías. Cuando una columna está completa, el sistema lo notifica imprimiendo el mensaje "COLUMNA COMPLETA", y cuando el avión está completo imprimiendo "AVION COMPLETO"

Se pide cumplimentar el esqueleto proporcionado en el ejer1 cumpliendo las siguientes características

- Generar una matriz como la descrita anteriormente para un avión vacío
- Generar 6 pasajeros aleatorios( dos letras aleatorias para las iniciales)
- Simular la llegada de clientes aleatorios y asignarles una posición hasta que el avión esté completo. Cuando una columna esté completa, el sistema lo notificará imprimiendo el mensaje "columna completa" y cuando el avión esté completo imprimiendo "AVION COMPLETO"
  - Pasajero.java (1 punto)
  - Principal (2 punto)
  - Vuelo (2 puntos)

## Ejercicio 2

Se pretende desarrollar un gestor de incidencias. El sistema debe registrar las incidencias que se van produciendo, anotando las operaciones realizadas para su solución, y su posible cierre, de todas estas operaciones se informa al sistema mediante mensajes. Cada mensaje contiene la información relacionada con una incidencia, y resultan de especial interés los campos "tiempo" (hora expresada en milisegundos) y la prioridad (ALTA o BAJA).

Existen 3 tipos de mensajes:

- **INFO:** Nueva incidencia o actualización de una existente.
- **CERRAR:** cierra la incidencia
- **ACTUACIÓN:** operación sobre una incidencia abierta

El proceso de tratar los mensajes se realizará para cada mensaje según los siguientes pasos:

1. Cada mensaje lleva una referencia al numero de incidencia al que va asociado.
2. Se debe utilizar dicho numero para localizar la incidencia
3. Si se trata de una nueva Incidencia (mensaje de tipo INFO con código de incidencia que no existe), se debe de crear un nuevo expediente (objeto incidencia) que contiene los datos básicos de la incidencia. Una incidencia tendrá el tipo de prioridad que tenga el mensaje a partir del que se crea.
4. Si se trata de información referente a una incidencia que se había recibido (mensaje tipo INFO (con código de incidencia que ya existe) , CERRAR o ACTUACIÓN) se debe considerar la siguiente posibilidad:
  - CERRAR: se trata de la solución definitiva que debe dar por cerrada la incidencia
  - INFO: Se trata de nueva información adicional sobre la incidencia, que será añadida al expediente como una **Nota**
  - ACTUACIÓN Se trata de intentos de soluciones temporales que no cierran la incidencia, pero pueden cambiar su prioridad y se debe almacenar como **Actualizaciones**.
5. El servicio de información de incidencias funciona de la forma bastante deficiente y por tanto es muy habitual que se envíen mensajes incorrectos y que serán registrados como **Error**. El sistema debe detectar como mensaje incorrecto como por ejemplo:
  - El cierre de una incidencia ya cerrada
  - La creación de incidencias con identificadores ya existentes.

Otro aspecto para tener en cuenta de la gestión de incidencias consiste en el cálculo del presupuesto del coste de cada incidencia.

Nota	50€
Actuación	200€

Únicamente se calculará el coste de incidencias cerradas. Si una incidencia no está cerrada su coste no se puede calcular y la función retorna -1 indicando un error.

Se pide:

- Diagrama de clases (1 punto)

- Completar el esqueleto proporcionado
  - Actuación.java (0.5 ptos)
  - Error.java (0.5 ptos)
  - GestorIncidencia.java (1 punto)
  - Incidencia.java (0.5 ptos)
  - Mensaje.java (0.5 ptos)
  - Nota.java (0.5 ptos)
  - Principal.java (0.5 ptos)