Examén Programación 1º DAW 12/03/2024

Ejer 1.- Debemos crear **GestAirport** software para la gestión de aeropuertos, el programa controlará la información de salida de los vuelos.

En primer lugar debemos logearnos en el sistema, para ello se pedira un usuario y contraseña que se validará con los usuarios almacenados (ver clase **Usuarios** de la librería). El usuario tendrá un máximo de 3 intentos para acceder al sistema.

Tendremos un menú con:

1. Mostrar cola de vuelos ordenada.
2. Dar salida a un vuelo.
3. Actualizar lista de vuelos.
4. Modificar horarios de un vuelo.
5. Salir

1. Mostrar cola de vuelos ordenada: Debe mostrar por pantalla la información de todos los vuelos que salgan en las próximas 24 horas. (Para ello es obligatorio usar alguna de las implementaciones del método sort de la clase **Collections**). La salida debe ser similar a la mostrada a continuación.

```
CODE: CS006 Origen: Santiago de Compostela Destino: Paris Salida: 2024-03-13T20:00 Llegada: 2024-03-13T23:00
CODE: CS004 Origen: Santiago de Compostela Destino: Madrid Salida: 2024-03-13T21:00 Llegada: 2024-03-13T22:30
CODE: CS002 Origen: Santiago de Compostela Destino: London Salida: 2024-03-13T22:00 Llegada: 2024-03-14T01:00
CODE: CS003 Origen: Santiago de Compostela Destino: Barcelona Salida: 2024-03-13T22:00 Llegada: 2024-03-13T23:30
CODE: CS005 Origen: Santiago de Compostela Destino: Oporto Salida: 2024-03-13T22:15 Llegada: 2024-03-13T23:45
CODE: CS007 Origen: Santiago de Compostela Destino: Barcelona Salida: 2024-03-14T22:00 Llegada: 2024-03-14T23:30
```

2. Dar salida a un vuelo: eliminar de la cola al primer vuelo de la lista.

3. Actualizar lista de vuelos: Comprobar la hora actual y eliminar de la lista aquellos vuelos cuya hora de salida ya haya sido superada.

4. Modificar horarios de un vuelo: pedir la cantidad de minutos que se debe retrasar o adelantar la salida de un vuelo y actualizar los valores de salida y llegada del minsmo.

5. Salir: Mostrar un mensaje de agradecimiento al usuario junto con el tiempo que ha estado ejecutando el programa y por último finalizar el programa.

Para la realización y pruebas se adjunta una librería que debéis utilizar como base, sino se utiliza se entenderá el programa como erróneo.

Para validar el funcionamiento se utilizarán los datos de de prueba suministrados.

La carga de datos debe realizarse automáticamente, al inicio de la ejecución del programa, convirtiendo el array bidimensional proporcionado por la clase **Datos** en objetos de tipo **Vuelo** (esta clase debe de estar relacionada con **VueloAbstract**)

En el caso de intentar modificar la hora de llegada o salida de un vuelo con una fecha/hora anterior a la actual, desde la clase Vuelo se lanzará una excepción del tipo **HorarioVueloException** que contenga el mensaje “La hora no puede ser anterior a la hora actual”. Esta excepción debe ser controlada desde el código, informando al usuario y no provocando la finalización del programa.

Para el almacenamiento de los vuelos podéis utilizar la estructura de almacenamiento que creáis conveniente (excepto arrays).

El main debe contener el mínimo de lineas de código posible y estará incluido en la clase **GestAirport** (nombre del proyecto).

Para el recorrido de los elementos de la colección o mapa que uséis, es obligatorio, usar como mínimo en una ocasión la clase **Iterator**.

El código debe seguir los paradigmas de la POO, por lo que toda la gestión de los datos se realizará en una clase llamada **Control**.

Entrega: un archivo .zip en el que se incluya el proyecto completo en formato **Netbeans**. Dicho proyecto, debe ser exportable a otras instancias de Netbeans, sin necesidad de modificar ningún archivo, para poder realizar la corrección.

Si la estructura de archivos no es correcta o se requieren modificaciones para poder ejecutar el código, se descontará un 20 % de la nota final del examen.

Todos los archivos java deben llevar una linea comentada (//Nombre_Apellido1_Apellido2) con el nombre y apellidos del alumno.

Calificación: En esta tabla se muestran las puntuaciones máximas que se pueden obtener en cada uno de los bloques.

Apartado 1: Cargar librería y uso de todas sus clases	0,75
Apartado 2: Carga de datos	1
Apartado 3: Logear usuarios	1,5
Apartado 4: Mostrar los vuelos ordenados	2
Apartado 5: Dar salida a un vuelo	0,75
Apartado 6: Modificar horario de un vuelo	1,5
Apartado 7: Uso de la interdace Iterator	1
Apartado 8: Salir	0,5
Apartado 9: Correcto funcionamiento del programa y ajustado a las condiciones solicitadas	1