Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI0202 - Principios de Informática
Examen parcial #3 (Reposición)
Ciclo I, 2019
Prof. Leonardo Villalobos Arias

Instrucciones: Escriba el código en Python 3 que funcione como describe el enunciado. Entregue su solución en un cuaderno de examen u hojas grapadas. Firme al entregar el examen. No se permite el uso de dispositivos electrónicos o material que contenga información relacionada al curso, a excepción de una hoja con materia del curso. Si realiza el examen en lápiz, pierde el derecho a reclamos.

Enunciado

La Dirección General de Aviación Civil desea generar información sobre los vuelos que parten de Costa Rica a diferentes destinos, desde los aeropuertos Juan Santamaría (SJO), Daniel Oduber (LIR) y Tobías Bolaños (SYQ). Esta institución desea saber cuántos vuelos y pasajeros parten de cada aeropuerto con el fin de programar la cantidad de empleados que debe asignar a cada aeropuerto por día, con especial interés en las diferencias que hay entre semana (de lunes a viernes) y el fin de semana (sábado y domingo).

Se cuenta con la información de vuelos y correspondientes pasajeros por vuelo, en un archivo de valores separados por coma (CSV) cuyo nombre es `salidas.csv`. La cantidad de filas o registros que contiene el archivo no es conocida de antemano. Cada registro (fila) del archivo tiene: el código del aeropuerto, el código de vuelo, el día del vuelo (L, K, M, J, V, S, D) y la cantidad de pasajeros que salieron del país en ese vuelo. El siguiente es un ejemplo de este archivo:

```
Aeropuerto;Vuelo;Dia;Pasajeros
SYQ;LI1215;S;90
SJO;RT8029;L;225
SJO;IX789;S;170
LIR;GH847;S;42
SYQ;MZ131;S;91
SJO;OQ9595;L;142
SJO;YE105;S;95
SJO;TW840;K;96
SYQ;QO393;K;69
LIR;NN5183;V;36
```

El programa debe inicialmente listar el detalle de vuelos de cada aeropuerto. Este detalle será parte de la información de cada aeropuerto y se implementará por medio de una ***matriz***. Cada fila de la matriz representaría un vuelo, y siempre tendría 3 columnas: el código del vuelo, el día y la cantidad de pasajeros. Esta sería la misma información que sería mostrada.

Adicionalmente el programa debe presentar un resumen de cantidad de vuelos por aeropuerto, diferenciando los vuelos que salen entre semana (L-V) de los que salen en fin de semana (S-D).

Finalmente, el programa debe presentar un consolidado de la cantidad de pasajeros que salió de cada aeropuerto, diferenciando los vuelos que salen entre semana (L-V) de los que salen en fin de semana (S-D). El siguiente sería un ejemplo de la salida del programa:

```
Detalle de vuelos por aeropuerto
Aeropuerto: Juan Santamaría
       Vuelo         Dia   Pasajeros
---------------------------------------
     RT8029          L        225
      IX789          S        170
     OQ9595          L        142
      YE105          S         95
      TW840          K         96

Aeropuerto: Daniel Oduber
       Vuelo         Dia   Pasajeros
---------------------------------------
      GH847          S         42
     NN5183          V         36

Aeropuerto: Tobías Bolaños
       Vuelo         Dia   Pasajeros
---------------------------------------
     LI1215          S         90
      MZ131          S         91
      QO393          K         69
```

```
Cantidad de vuelos
            Aeropuerto    L-V    S-D
---------------------------------------
       Juan Santamaría      3      2
         Daniel Oduber      1      1
        Tobías Bolaños      1      2

Cantidad de pasajeros
            Aeropuerto    L-V    S-D
---------------------------------------
       Juan Santamaría    463    265
         Daniel Oduber     36     42
        Tobías Bolaños     69    181
```

Finalmente, el programa debería estar programado en el paradigma orientado a objetos. Se utilizará un diseño de 2 clases: `Aeropuerto` y `Aviación`. `Aeropuerto` estaría encargada de almacenar la información correspondiente a los vuelos que salen de este lugar, y mostrar esta información y sus totales. `Aviación` estaría encargada de almacenar las instancias de `Aeropuerto`, leer la información del archivo, y mostrar los resultados.

Evaluación

1. **(60%)** Implementa una clase `Aeropuerto`, encargada de almacenar la información que corresponde con los vuelos que salen de ese aeropuerto.
    a. **(5%)** Implementa atributos que le permiten representar su `nombre` y la información que corresponde con sus `vuelos` (sugerencia: use una matriz de 3 columnas)
    b. **(55%)** Implementa los siguientes métodos:
        i. **(5%)** Un constructor para inicializar los atributos de la clase.
        ii. **(5%)** `agregar_vuelo(numero, dia, pasajeros)` que agrega el vuelo con estos datos en la matriz de vuelos.
        iii. **(15%)** `imprimir_vuelos()` que muestra los vuelos asociados al aeropuerto.
        iv. **(15%)** `imprimir_vuelos_periodo()` que muestra el nombre del aeropuerto, así como la cantidad total de vuelos asociados a los periodos L-J y S-D.
        Ejemplo: `Juan Santamaría  3  2`
        v. **(15%)** `imprimir_pasajeros_periodo()` que muestra el nombre del aeropuerto, así como la cantidad total de pasajeros asociados a los periodos L-J y S-D.
        Ejemplo: `Juan Santamaría  463  265`
2. **(40%)** Implementa una clase `Aviación`, encargada de representar los 3 aeropuertos, leer el archivo y mostrar los resultados.
    a. **(5%)** Implementa atributos que le permiten representar los 3 aeropuertos: Juan Santamaría (`SJO`), Daniel Oduber (`LIR`) y Tobías Bolaños (`SYQ`).
    b. **(35%)** Implementa los siguientes métodos:
        i. **(5%)** Un constructor para inicializar los atributos de la clase.
        ii. **(15%)** `leer_datos()` que abre el archivo de datos, lee la información de este, y la almacena en sus atributos.
        iii. **(15%)** `mostrar()` que muestra, para cada aeropuerto, sus vuelos, su cantidad de vuelos por periodo, y su cantidad de pasajeros por periodo.

Nota: En cada punto se evaluará la correctitud, completitud, y buenas prácticas de programación. Estas incluyen, pero no se limitan a identificadores significativos para variables e indentación adecuada.