Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
CI-0202 Principios de Informática
Examen parcial #2, Grupo 12
II Semestre 2020, 27 de noviembre 2020
Docente Sivana Hamer

Declaro bajo juramento que realizaré esta evaluación de forma **estrictamente individual**. En caso de que se sospeche lo contrario, estoy consciente de que se me aplicará el debido proceso estipulado en el Reglamento de orden y disciplina de los/las estudiantes de la Universidad de Costa Rica, el cual plantea sanciones de suspensión hasta por seis años.

Instrucciones

El examen se compone de dos partes **completamente individuales**. La primera parte vale 100 puntos y se realiza el viernes 27 de noviembre del 2020 de 11:00 am a 1:00 pm de manera sincrónica. La segunda con un puntaje de 10 puntos adicionales y se realiza desde la 1:05 pm el viernes 27 de noviembre del 2020 hasta la 1:05 pm el lunes 30 de noviembre del 2020 de manera asincrónica. **No se aceptarán entregas tardías.** 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. Además, si considera que requiere de más funciones para solucionar el examen, puede implementarlas.

Primera parte

Para esta parte, resolverá el problema planteado en el enunciado como si lo estuviera resolviendo en un cuaderno de examen físico. Escriba el código en Python 3 que soluciona el problema planteado con bolígrafo en su cuaderno o una hoja de papel. Si necesita hacer correcciones, puede tachar, siempre que el resultado sea claro. **Si la primera parte es realizada a computadora, se perderá la oportunidad de obtener puntos adicionales en la segunda parte del examen. Además, si realiza el examen a computadora todo el código entregado por el o la estudiante debe compilar sin errores, de lo contrario será calificado con 0.** Al finalizar el examen, debe tomar fotografías legibles de todo el examen, crear un archivo PDF con todas las imágenes ordenadas, y subir el archivo pdf el viernes 27 de noviembre del 2020 antes de las 1:05 pm al aula virtual. A menos que esté escrito explícitamente, no será necesario realizar validación de datos de entrada. Suponga que los datos que se ingresan siempre son válidos.

Enunciado

La compañía Dotify provee servicios de transmisión de audio. Específicamente su servicio más popular es que tienen disponible millones de canciones de distintos artistas que cualquier usuario/usuaria del servicio puede reproducir. Dado la gran cantidad de usuarios/usuarias que escuchan el servicio, están interesados en recolectar información sobre la popularidad de las canciones que se encuentran disponibles en sus servicios.

La información sobre el uso de la plataforma es registrada anualmente. Los archivos se crean automáticamente con un nombre `canciones-usuarios-AAAA.csv` donde AAAA es el año. A usted se le asigna implementar las estadísticas del archivo `canciones-usuarios-2020.csv`. Cada línea del archivo contiene información de las reproducciones de una canción por usuario/usuaria. Por consiguiente, las líneas de un archivo representan todas las reproducciones que los/las usuarios/usuarias realizaron durante el año en la plataforma. Una línea incluye el nombre de la canción, el artista que la interpreta, la calificación dada por el/la usuario/usuaria medido en estrellas de 1 a 5 y reproducciones. Cada canción tiene un nombre y un/una artista que funcionan como identificadores. Cada usuario/usuaria asigna una calificación en estrellas de la canción, que es un entero que va de 1 (calificación más baja) a 5 (calificación más alta). Por último, se guarda la cantidad de veces que el/la usuario/usuaria reproduce la canción por año. El siguiente ejemplo es como se vería el archivo:

Licencia: CC BY-NC-SA 4.0

```
nombre;artista;calificación;reproducciones
thank u, next;Ariana Grande;4;20
Sorry;Justin Bieber;3;3
Why not;LOONA;5;100
bad guy;Billie Eilish;1;12
TUSA;Karol G & Nicki Minaj;2;207
thank u, next;Ariana Grande;5;30
TUSA;Karol G & Nicki Minaj;3;51
Macarena;Los Del Rio;1;105
TUSA;Karol G & Nicki Minaj;5;34
Batalla Espiritual;Fabricio Alvarado;3;998
bad guy;Billie Eilish;2;29
Poker Face;Lady Gaga;3;101
Why not;LOONA;5;11
```

Cada canción se le debe calcular el promedio de estrellas (PE) dadas por los/las usuarios/usuarias. Para hacer esto, se calcula mediante la siguiente fórmula, donde e_i es una calificación de estrellas y n la cantidad de calificaciones que la canción recibió.

$$PE = \left\lfloor \frac{1}{n} \cdot \sum_{i=1}^{n} e_i \right\rfloor$$

Cuando es ejecutado, el programa deberá leer los datos del archivo y con base a eso mostrar para cada canción su nombre, artista, cantidad promedio de estrellas, cantidad total de reproducciones, y cantidad de usuarios/usuarias distintos que han escuchado la canción. Las canciones aparecen dado el orden de aparición del archivo. El siguiente es un ejemplo de cómo debería ser la salida del programa:

```
---------------------------------    ---------------------------------
Nombre: thank u, next                Nombre: TUSA
Artista: Ariana Grande               Artista: Karol G & Nicki Minaj
Estrellas: ☆ ☆ ☆ ☆                   Estrellas: ☆ ☆ ☆
Reproducciones: 50   Usuarios: 2     Reproducciones: 292   Usuarios: 3
---------------------------------    ---------------------------------

---------------------------------    ---------------------------------
Nombre: Sorry                        Nombre: Macarena
Artista: Justin Bieber               Artista: Los Del Rio
Estrellas: ☆ ☆ ☆                     Estrellas: ☆
Reproducciones: 3   Usuarios: 1      Reproducciones: 105   Usuarios: 1
---------------------------------    ---------------------------------

---------------------------------    ---------------------------------
Nombre: Why not                      Nombre: Batalla Espiritual
Artista: LOONA                       Artista: Fabricio Alvarado
Estrellas: ☆ ☆ ☆ ☆ ☆                 Estrellas: ☆ ☆ ☆
Reproducciones: 111   Usuarios: 2    Reproducciones: 998   Usuarios: 1
---------------------------------    ---------------------------------

---------------------------------    ---------------------------------
Nombre: bad guy                      Nombre: Poker Face
Artista: Billie Eilish               Artista: Lady Gaga
Estrellas: ☆ ☆                       Estrellas: ☆ ☆ ☆
Reproducciones: 41   Usuarios: 2     Reproducciones: 101   Usuarios: 1
---------------------------------    ---------------------------------
```

Licencia: CC BY-NC-SA 4.0

Su solución debe estar diseñada siguiendo el paradigma orientado a objetos. Se utilizó un diseño de 2 clases: `Canción` e `Dotify`. `Canción` estará encargada acerca de las reproducciones de las canciones. `Dotify` estará encargada de almacenar las instancias de `Canción`, leer información del archivo, y mostrar los resultados.

Evaluación

1. (45 puntos) Implementa la clase `Canción`.
   a) (10 puntos) `Canción` tiene atributos que le permiten almacenar las canciones con sus estadísticas.
   b) (35 puntos) `Canción` implementa los siguientes métodos:
      1) (5 puntos) Un constructor para inicializar los atributos de la clase.
      2) (3 puntos) `agregar_reproduccion(r)` que agrega `r` reproducciones al total de reproducciones de la canción.
      3) (3 puntos) `agregar_usuario(u)` que agrega `u` usuarios al total de usuarios de la canción.
      4) (4 puntos) `agregar_estrella(e)` que agrega `e` estrella a las estrellas de la canción.
      5) (10 puntos) `promedio_estrellas()` que calcula y retorna el promedio de estrellas totales de la canción.
      6) (10 puntos) `mostrar()` que muestra la información de la `Canción`.
      Ejemplo:
      ```
      ---------------------------------
      Nombre: thank u, next
      Artista: Ariana Grande
      Estrellas: ☆ ☆ ☆ ☆
      Reproducciones: 50   Usuarios: 2
      ---------------------------------
      ```

2. (55 puntos) Implementa la clase `Dotify`.
   a) (5 puntos) `Dotify` tiene un atributo que le permite guardar las distintas canciones.
   b) (50 puntos) `Dotify` implementa los siguientes métodos:
      1) (5 puntos) Un constructor para inicializar los atributos de la clase.
      2) (15 puntos) `buscar_cancion(nombre, artista)` que busca la instancia de la canción con el `nombre` y `artista`, y lo retorna. En el caso de que no existe retorna una instancia nueva de `Canción`.
      3) (25 puntos) `leer_archivo()` que abre el archivo `canciones-usuarios-2020.csv`, lee la información de este, y la agrega a cada una de las instancias de `Canción`.
      4) (5 puntos) `mostrar()` que muestra la información de cada `Canción`.

Licencia: CC BY-NC-SA 4.0

Segunda parte

La segunda parte del examen es para crédito adicional opcional en el cual se debe corregir la solución de la primera parte del examen. Inicia desde la 1:05 pm el viernes 27 de noviembre del 2020 hasta la 1:05pm el lunes 30 de noviembre del 2020 de manera asincrónica. Primero, debe transcribir todo el código realizado en la primera parte del examen en Python 3 en un archivo de código fuente (.py). Luego, debe realizar las modificaciones para que el programa funcione como indica el enunciado de la primera parte. **Todo el código entregado por el o la estudiante debe compilar sin errores, de lo contrario será calificado con 0.** Para esta parte, debe realizar validación de datos de entrada.

Evaluación

1. (10 puntos) El programa debe realizar y resolver el problema descrito por el enunciado.

Licencia: CC BY-NC-SA 4.0