PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada
1^{er} semestre 2017

Actividad 06

Funcional

Introducción

Llevas todo el día estudiando la materia de Programación funcional que ya nos das más, por lo que decides ver ~~muchas~~ una película, sin embargo hay tantas películas buenas como *'Lo que Mavrakis se llevó', 'Buscan do a Bastián', 'Bastian Rangers', 'Ba Ba Bastiand, etc.* que no logras decidirte. Luego de un rato llegas a la brillante idea de crear un programa que te ayude a resolverlo, para esto ingresas a *Prograflix* a ver como puedes comenzar.

El objetivo es realizar un sistema de consultas que te permita obtener información sobre películas y actores. Para esto, obtienes mágicamente la base de datos de *Prograflix* organizada en dos archivos especificados a continuación:

- **movies.txt**: Archivo de texto que contiene la información de las películas que puedes ver, de la forma: IDPelícula, Nombre Pelicula, Rating, Fecha de estreno, género_1, género_2, ..., género_n (es una cantidad variable de géneros).
- **cast.txt**: Archivo de texto que alberga la información de los artistas que trabajaron en las películas. Está distribuido de la forma: Nombre Película, Nombre Artista, Nombre Personaje.

Con esta información, debes ser capaz de resolver las consultas especificadas en los requerimientos de la actividad

Requerimientos

- Tu solución debe ser realizada con **programación funcional**
- Quieres que tu programa no almacene el archivo en memoria, para eso la lectura de ellos tendrá que ser sólamente mediante el uso de generadores. (0.5 ptos)

- Cada película debe tener su propio `id` el cual debe estar implementado con **generadores**. El `id` es distinto al que trae la base de datos, para que puedas distinguirla con tu programa. (0.5 ptos)
- Tu programa deberá realizar las siguientes consultas: (4.5 ptos)
  1. Crear el método `popular` que dado un número, retorne todas las películas que tienen un rating superior a dicho valor. (0.5 ptos)
  2. Crear el método `with_genres` que dado un número 'n', retorne todas las películas que tienen 'n' o más géneros. (0.5 ptos)
  3. Crear el método `tops_of_genre` que dado un género, retorne las 10 mejores películas ordenadas. (1.0 ptos)
  4. Crear el método `actor_rating` que dado el nombre de un actor, retorna el promedio del rating de las películas en las que ha participado. (1.0 ptos)
  5. Crear el método `compare_actors` que dado el nombre de dos actores, imprima cual de los dos esta mejor valorado según su promedio. (0.5 ptos)
  6. Crear el método `movies_of` que dado el nombre de un actor. retorne una lista con tuplas que contenga los pares (pelicula, personaje) en los que actuo. (0.5 ptos)
  7. Crear el método `from_year` que dado un año, retorne todas las peliculas que se estrenaron en ese año. (0.5 ptos)
- Deberás poblar tu sistema con consultas que prueben que tu programa funciona. (0.5 ptos)

Notas

- No debes modificar los archivos movies.txt y cast.txt
- No está permitido el uso de ciclos *for y while*, salvo dentro de **generadores, listas por comprensión o para imprimir resultados**.
- Para comparar fechas puedes usar `datetime.strptime` de la librería `datetime`
- Ejemplo de como **NO** leer archivos: En ningún momento puedes tener alguna estructura de datos con un formato similar a `[linea_1, linea_2, linea_3,...,linea_n]`
- Ejemplo de una opción como podría leerse un archivo:

```
with open(filename, 'r') as f:
   for line in f:
     # Aqui se usa un generador
```

Entrega

- **Lugar:** GIT - Carpeta: Actividades/AC06
- **Hora:** 16:55