OBSERVACIONES DEL LA PRACTICA

Cristian Armando Sánchez 202022112

Luis Felipe Dussán R. Cod 201912308

1. ¿Cuáles son los mecanismos de interacción (I/O: Input/Output) que tiene el **view.py** con el usuario?

En el archivo view.py podemos encontrar interacciones con el usuario cuando se imprime el menú, el usuario debe escoger una opción entre las cinco que hay. Cada una de estas opciones carga un archivo .csv e imprime lo que las funciones de cada opción solicita.

1. ¿Cómo se almacenan los datos de **GoodReads** en el **model.py**?

Primero se crea un catalogo, el cual es una lista donde se almacenarán todos los libros. A su vez, se crean otras listas para guardar los otros datos como autores, generos , entre otros. Después se van añadiendo los datos a las listas creadas por medio de sus respectivas funciones.

1. ¿Cuáles son las funciones que comunican el el **view.py** y el **model.py**?

Primero vamos a analizar esta función de getBestBooks(), para observar como se comunican el view.py y el model.py:

Esta es la parte que se muestra en el view, un elif si el usuriario selecciona la opción número 2. Aquí se hace un llamado al controller, para que se active la función getBestBooks.

Graphical user interface, text

Description automatically generated with medium confidence

Este es el codigo que se muestra en el controller, una función que llama a otra llamada getBestBooks, que se encuentra en el model.

Graphical user interface, text, application

Description automatically generated

Por último encontramos la función getBestBooks del model, la cúal extrae los best books.

Text

Description automatically generated

1. ¿Cómo se crea una lista?

En este archivo, ADT.list. Se crea una lista a apartir de una función llamada newList(), en aquella función se define una estrcutura básica la cual será usada cuando se quiera crear una lista. Para crear una lista con este tipo abstracto de dato hay una serie de parámetros que se pueden llenar para hacer de la lista más completa; como por ejemplo: definir la estructura, agregar una función de comparación, un identificador, el nombre de un archivo .csv por si se quiere crear una lista apartir de los elementos en el archivo, y por último un delimeter, el cual se usa para separar los campos en la lista. Aquí se puede observar el ejemplo:

Text

Description automatically generated

1. ¿Qué hace el parámetro **cmpfunction=None** en la función **newList()**?

Si se define este párametro se comparan los elementos de la lista.

1. ¿Qué hace la funció **addLast()**?

Es una función que agrega un elemento a la última posición de la lista, además actualiza el apuntador a la última posición, y por último hace que el tamaño de la lista se aumente en 1.

1. ¿Qué hace la función **getElement()**?

La función hace un recorrido por la lista hasta que encuentre el elemento pos, el cual debe ser mayor que 0 y no puede ser más grande que el tamaño de la lista. Cuando lo encuentra, lo retorna.

1. ¿Qué hace la función **subList()**?

Esta función retorna una sublista de la lista. Esta sublista contiente elemntos desde la posición pos, hasta una longitud dada.

1. ¿Observó algún cambio en el comportamiento del programa al cambiar la implementación del parámetro **“ARRAY\_LIST”** a **“SINGLE\_LINKED”**?

Cuando se encuentra con el formato “ARRAY\_LIST” se demora 58000.0000000000 nano seconds

Cuando se carga con el formato “SINGLE\_LINKED” se demora 46000.0000000000 nano seconds

Por lo que el cambio más notable es el tiempo que se demora. Siendo el formato SINGLE\_LINKED, el más eficiente en tiempo.