OBSERVACIONES DEL LA PRACTICA

Camilo Ortiz Cruz Cod 201821615

Kevin Fernando Gomez Camargo Cod 202015120

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

El mecanismo de Input que tiene el módulo view.py con el usuario es el input() que permite al usuario poner el numero de la opción que desea:

Texto

Descripción generada automáticamente



Por otro lado, el mecanismo de Output que se utiliza para comunicarse con el usuario es una serie de funciones que son PrintBestBooks(), PrintAuthorData() y funciones de print() normales que le muestran en consola la información que pidió el usuario.

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

Los datos de GoodReads se almacenan en un catálogo el cual es un diccionario con 4 llaves, las cuales son books, authors, tags, booktags, estas llaves tienen como valor listas las cuales pueden ser arreglos o listas enlazadas, la forma en que esta el programa utiliza arreglos para authors, tags y booktags y una lista enlazada para books.

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

Las funciones que comunican el view y el model son initCatalog() y loadData() pues estas son las únicas funciones que hacen referencia al controller el cual termina llamando a las funciones del model, algo importante para resaltar es que aunque no sean funciones del view, en la ejecución del programa view llama a tres funciones del controller que llaman a las funciones del model, esto sucede cuando el usuario da los inputs 2,3 y 4:

Texto

Descripción generada automáticamente

Sin embargo, no son funciones del view pero comunican al view con el model.

1. ¿Cómo se crea una lista?

Para crear una lista se debe llamar del módulo list.py que se encuentra en DISCLib/ADT, la función newList() a esta función se le pueden pasar opcionalmente 5 parametros con nombre los cuales son datastructure que determina que tipo del lista crear, cmpfunction la cual es la función que utilizara la lista para comparar los elementos de esta misma, key el cual se utiliza para acceder a una llave especifica de los elementos para compararlos, filename el cual recibe la ruta de un archivo para agregar los datos de este a la lista creada y delimiter el cual indica como están separados los datos si es que se pasa un filename.

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

Al pasar cmpfunction=None a la hora de crear la lista causa que la función de comparación de la lista sea la establecida por defecto

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

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

La función addLast() agrega un elemento en la última posición de la lista e incrementa el tamaño en 1

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

Esta función retorna el elemento que se encuentra en la posición pasada por parámetro.

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

La función subList() retorna una lista nueva con los elementos a partir de una posición dada por parámetro con un size de numelem que también es dada por parámetro

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

Al cambiar el tipo de lista de arreglo a lista enlazada no se percibieron cambios en la ejecución del programa, sin embargo, se debe resaltar que se utilizo los datos “-small” por lo cual el tiempo para realizar las operaciones no es significativo entre listas enlazadas y arreglos, no obstante, algunas opciones pueden hacer que se demore más, por ejemplo, la opción 4 tardaría mas tiempo ya que al cambiar la forma en como se guardan los tags a una lista enlazada, la función countBooksByTag que utiliza getElement podría tomar mas tiempo ya que en una lista enlazada getElement es O(n), sin embargo, habría que revisar la complejidad de tiempo de la función para ver si este cambio afecta o no.