OBSERVACIONES DEL LA PRACTICA

Blanca Valencia Aguirre Cod 202013797

Sergio Arango Arango Cod 201921814

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

El view.py se comunica inicialmente con el usuario con la función “print\_menu” que imprime la distintas funciones del programa, luego toma como entrada el numero de la función que el usuario ingrese, en cada caso el view imprime cierta confirmación de que está ejecutando la función que el usuario escogió y procede a pedirle más información al usuario si es necesario para configurar la función actual y la recibe para terminar de correr la función, finalmente imprime los resultados de la función y vuelve a mostrar el menú principal.

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

Se almacenan en lo que estamos refiriendo como un “catalogo” que en este caso es un diccionario nativo de Python. Sus llaves son: “books”, “authors”, “tags” y “book\_tags”. Donde respectivamente se guardan :

un TAD Lista representado por la estructura Lista Encadenada sin función de comparación,

un TAD Lista representado por la estructura Arreglo con la función de comparación “compareauthors”, un TAD Lista representado por la estructura Arreglo con la función de comparación “comparetagnames”,

un TAD Lista representado por la estructura Arreglo sin función de comparación.

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

initCatalog, loadData, controller.getBestBooks, controller.getBooksByAuthor, controller.countBooksByTag

1. ¿Cómo se crea una lista?

De la carpeta TAD de la librería DISClib, del módulo list importado como lt se llama a la función “newList”. Esta función toma como parámetros la estructura de datos a utilizar, la función de comparación que definira el órden del conjunto, si no se proporciona una función de comparación se usa la función por defecto pero se debe proveer el parámetro “key” para esta función. Opcionalmente se le puede dar el nombre de un archivo como parámetro para que tome los elementos del archivo como los datos a guardar, en este caso también se debe dar el parámetro de delimitador para que identifique correctamente la separación de los datos en el archivo (por defecto es “,”).

Esta función llama la función también llamada “newList” en el módulo List Structure ubicada en la carpeta Data Structures, la cuál dependiendo de la estructura de datos escogida llama al módulo Arraylist o Linkedlist dónde se define la estructura de datos y se guarda en memoria con los datos.

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

El parámetro toma una función “cmpfunction” de comparación. Es decir, esta función debe tener una estructura específica la cuál es tomar dos elementos (exactamente del tipo de elementos que se guardaran en la lista en la que esta función se asignara) y la función retorna un bool indicando si el primero (o el segundo) es “mayor” que el segundo (o el primero). Esta función define esencialmente cuál es el criterio para decir que un elemento es mayor que otro, por lo tanto define una estructura de orden de la lista (el cual puede ser parcial o total).

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

Esta función agrega un elemento en la última posición del TAD Lista. Toma como parámetro al elemento que agregará y la lista a la que lo agregará. No retorna parámetro

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

Esta función toma como parámetro una posición y la lista de la cuál debe revisar la posición. Retorna el elemento que se encuentra en la posición indicada de la lista indicada. No modifica la lista.

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

Esta función toma como parámetros una lista, un entero que representa una posición, y un entero que representa una cantidad. La función retorna una sublista (lista subconjunto) de la lista proporcionada, definida por el primer elemento siendo la posición parámetro de la lista original, y tomando los siguientes elementos hasta completar la cantidad ingresada como parámetro.

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

La función loaddata se toma más tiempo en cargar los datos cuando se cambia la estructura de datos de Array List a Single Linked List definida en la función initCatalog.