OBSERVACIONES DEL LA PRACTICA

Diego Acosta Corredor Cod 202110516

Mateo Cote Canal Cod 202022609

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

La primera interacción que tiene el código con el usuario es presentarle el menú de opciones o la interfaz seguido de un mensaje o sección para que el usuario introduzca una opción delimitada por un número (input). Posteriormente, dependiendo de la opción ingresada, se muestra un mensaje por consola y los resultados de la operación seleccionada (output).

Texto

Descripción generada automáticamente

En este ejemplo, en primer lugar se puede ver el menú que se le muestra al usuario (de “Bienvenido” hasta “Salir”). Posteriormente se habilita la sección de “seleccione una opción para continuar” donde el usuario introduce la opción que quiere ejecutar con un número como parámetro que coincida con las opciones del menú. Finalmente se ejecuta la opción seleccionada mostrando por consola un mensaje introduciendo la información y posteriormente la información cargada.

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

Por medio de la función newCatalog() lo que se hace es crear un diccionario de listas en el cual se almacenarán los datos en cuatro categorías: books, authors, tags y book-tags. Para añadir los datos a cada una de las listas se implementan otras cuatro funciones: addBook(), addBookAuthor(), addTag(), addBookTag(), cuya finalidad es añadir elementos al final de su respectiva lista/categoría. Esto de manera tal que la estructura final de los datos almacenados se organice en un diccionario de listas por categoría.

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

Primeramente, para inicializar un catálogo, la vista tiene una función llamada initCatalog():

Texto

Descripción generada automáticamente

Esta función a su vez se comunica con el controlador por medio de una función con ese mismo nombre:

Texto

Descripción generada automáticamente

Por último el controlador se comunica con el modelo por medio de la función newCatalog():



Otras funciones que siguen este mismo patrón son (view.py, model.py):

* SECCIÓN MENÚ PRINCIPAL → getBestBooks(catalog,number) → getBestBooks(catalog, number)
* SECCIÓN MENÚ PRINCIPAL → getBooksByAuthor(catalog,authorname) → getBooksByAuthor(catalog, authorname)
* SECCIÓN MENÚ PRINCIPAL → countBooksByTag(catalog, tag) → countBooksByTag(catalog, tag)

1. ¿Cómo se crea una lista?

Por medio de la función newList(), la cual puede recibir varios parámetros que influyen en la creación de la lista. Entre ellos está el tipo de lista a crear (puede ser tanto un arreglo como una lista simplemente encadenada). También existen otros parámetros como funciones de comparación entre elementos de la lista, o el nombre de un archivo CSV de donde se extraerán los elementos para crear la lista. De esta forma se crea una lista vacía bajo ciertas reglas que van a indexar los elementos incluidos desde un archivo CSV (filename) ya sea para compararlos (cmpfunction y key), delimitarlos (delimiter) o estructurarlos (datastructure).

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

Es un parámetro que indica una función de comparación entre los elementos existentes de la lista. Está ligado al parámetro key tal que se le asigne una llave a cada elemento de la lista y se utilice el valor del cmpfunction para compararlos unos a otros.

1. ¿Qué hace la funció **addLast()**?

Dados un elemento y una lista, la función añade dicho elemento en la última posición de la lista.

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

Recibe como parámetro una lista y una posición de tipo entero. Lo que hace es retornar el elemento que se encuentra en dicha posición de la lista ingresada.

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

Crea una sublista a partir de la lista original. Para esto, la sublista empezará desde la posición ingresada y seguirá hasta completar el número de elementos ingresado por el parámetro “numelem”.

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

Al aplicar el cambio de “ARRAY\_LIST” a “SINGLE-LINKED” el único cambio observable fue que las funciones corrían más lento, es decir, el código se demoraba más en retornar lo que pedía el usuario.