

Observaciones Lab 3

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

El mecanismo de interacción principal con el usuario es el menú principal, pues es el que permite dar la información sobre qué actividades se pueden realizar con el programa. Además, el input después de que se imprime el menú permite comunicar lo que quiere hacer el usuario con lo que debe ejecutar el algoritmo; estas órdenes son interpretadas y ejecutadas por el programa y luego se devuelven en forma de prints.

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

En model.py se almacena la información de GoodReads como un catálogo, más específicamente como un diccionario. Este diccionario está compuesto por diferentes listas; hay una lista diferente para el nombre de los libros, los autores, los tags, y los tags de los libros. Así mismo, cada lista tiene un tipo de dato distinto; algunas son arreglos y otras listas encadenadas.

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

Los archivos de view.py y model.py se comunican entre sí por medio del archivo controller.py. Específicamente las funciones que comunican estos documentos son: newController() y loadData(control).

4. ¿Cuál es la función que permite crear una lista?, ¿Qué datos son necesarios?

La función que permite crear una nueva lista se llama newList y los datos que necesita son: dataestructure, función de comparación, llave, nombre del archivo y delimitador.

5. ¿Para qué sirve el parámetro datastructure en la función newList()?, ¿Cuáles son los posibles valores para este parámetro?

El parámetro dataestructure en newList sirve para indicar qué tipo de estructura de datos se va a usar para crear la lista. Los tipos posibles de dataestructure son arreglo, lista enlazada y lista doblemente encadenada.

6. ¿Para qué sirve el parámetro cmpfunction en la función newList()? Cmpfunction sirve para establecer una función de comparación entre los elementos de la lista, según algún criterio deseado.

7. ¿Qué hace la función addLast()?

Con addlast() se agrega un elemento en la última posición de una lista.

8. ¿Qué hace la función getElement()?

getElement() retorna un elemento en una posición determinada de la lista. Con esta función se recorre la lista hasta la posición indicada, de manera que se retorna el valor asociado a esta posición.

9. ¿Qué hace la función `subList()`?

`SubList()` crea una copia de la lista y crea una sublista a partir de esta, la cual contiene los elementos desde una posición dada, hasta una longitud establecida. Así pues, se retorna esta 'nueva' lista.

10. Revise el uso de la función `iterator()` en las funciones `printAuthorData(author)` y `printBestBooks(books)` en la Vista que aplican a una lista de libros. ¿Qué hace la función `iterator()`?

Con esta función se recorre la lista, de manera que si se tiene un elemento pasa al siguiente para evaluarlo; de otra forma la sucesión para.

11. ¿Observó algún cambio en el comportamiento del programa al cambiar el valor del parámetro 'datastructure' en la creación de las listas?

Un cambio notable es el tiempo de lectura y organización de datos. A mi percepción, cambiando las estructuras de datos, el programa se demora un poco más. Así mismo, queda claro que la manera en que se organiza la información es diferente, lo que repercute la eficiencia de ciertas funciones.