OBSERVACIONES DEL LA PRACTICA

Estudiante 1 Cod XXXX

Estudiante 2 Cod XXXX

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

Los mecanismos de interacción del view.py con el usuario son:

* Input() - Mecanismo de tipo Input
* Print() - Mecanismo de tipo Output

En la siguiente captura de código se pueden evidenciar estos dos mecanismos integrados en una función:

Texto

Descripción generada automáticamente

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

Los datos del GoodReads son almacenados en distintos Arraylist que a su vez hacen parte del diccionario “catalogo”.

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

Varias funciones del view.py llaman a ciertas funciones del controller.py, que a su vez llaman a funciones del model.py.

Estas son:

* initCatalog()
* loadData()

Además, en el while del menu principal, también se llaman a distintas funciones del controller.py que llaman a las del model.

Estas son:

* getBestBooks() -> Ejemplo
* getBooksByAuthor()
* countBooksByTag()

Ejemplo:

Model:

Texto

Descripción generada automáticamente

Controller:

Imagen que contiene Texto

Descripción generada automáticamente

View:

Texto

Descripción generada automáticamente

1. ¿Cómo se crea una lista?

Primero se define la función, en este caso newList, y en los parámetros se deben definir: el datastructure, que es el tipo de estructura de la lista (SINGLE\_LINKED o ARRAYLIST); el cmpfunction, que es la función comparación; el key, que identificador del cmpfunction para comparar los elementos de la lista; el filename, que es el nombre del archivo sobre el que se crea la lista (recibe archivos tipo csv o utf8); y el delimiter, que es lo que separa cada campo del archivo, en general con una coma.

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

Define la función de comparación y además hace que key sea “None”.

1. ¿Qué hace la funció **addLast()**?

Agrega un elemento en la última posición de la lista y actualiza el apuntador de la última posición.

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

Retorna el elemento en una posición dada (sin eliminarlo). Es importante tener en cuenta que la posición del primer elemento es 1, la del segundo es 2 y así sucesivamente.

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

Se retorna una sublista a partir de una lista, que contiene los elementos de esta a partir de una posición específica y con una longitud definida por un número de elementos dado.

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

Sí, cuando se corrió el view.py fue más rápida la carga de los datos.