Manuel Villaveces

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

View.py presenta un menú de texto por consola con opciones y números 0-4. El usuario elige la opción ingresando un input por consola del numero que corresponde con la opción elegida. El programa ejecuta diferentes funciones dependiendo del valor del input, y devuelve texto por consola con la información solicitada.

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

En la función newCatalog() se crea la estructura de datos ”catalog”, los datos de goodreads se agregan a la estructura “catalog” con la función addBookTag().

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

View.py importa el controlador en la línea 29 “import controller”. Y lo usa en la función newController() en la línea 100 y desde el menú principal cuando se ingresa un input invoca las funciones.

El controlador importa el modelo en la línea 29 “import model”. Y la usa para cargar y manipular las estructuras en todas las funciones, completando el método MVC

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

La función newList() de DISClib crea la lista, necesita como parámetro datastructure.

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

El parámetro datastrucure sirve para elegir y definir la estructura de datos que se va a crear.

f. ¿Para qué sirve el parámetro cmpfunction en la función newList()?

La función newList() usa el parámetro cmpfuntion para comparar los elementos de la lista.

g. ¿Qué hace la función addLast()?

Agrega un elemento a una lista, en la última posición.

h. ¿Qué hace la función getElement()?

Devuelve el valor del elemento en la lista, que este en la posición entregada por parámetro.

i. ¿Qué hace la función subList()?

Crea una lista mas pequeña que la lista entregada como parámetro, de la posición y longitud también especificados por los parámetros.

j. 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()?

La función iterator() hace un recorrido completo de la estructura de datos buscando un elemento de valor específico y aumenta un contador cada vez que lo encuentra.

Sebastian Montoya

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

De acuerdo con los supuestos empleados en TAD lista, la función que nos permite crear una lista es:(newList), sin embargo, antes de hacer uso de ella es necesario haber importado el TAD,en este caso: list. Ahora bien, la función (newlist) ha de recibir dos parámetros:

1. La estructura de datos que vamos a usar (arreglos o lista encadenada)

2. Una función de comparación provista por el usuario para comparar elementos.

¿Para qué sirve el parámetro datastructure en la función newList()?, ¿Cuáles son

los posibles valores para este parámetro?

El valor datastructure nos otorga la capacidad de elegir el tipo de estructura que deseamos emplear para guardar la información. Los valores potenciales que puede aceptar son: ARRAY\_LIST o SINGLE\_LINKED. Ambas son organizaciones que deben ser seleccionadas dependiendo de las necesidades o beneficios que sean pertinentes en la situación o que percibamos como óptimas.

¿Para qué sirve el parámetro cmpfunction en la función newList()?

El parámetro cmpfunction cumple la función de llevar a cabo la comparación entre los elementos. En otras palabras, se trata de una "función" de comparación que se configura con el propósito de orientar la manera en que se organiza la lista. Dentro de una definición abstracta de tipo de dato (TAD) relacionada con listas, es habitual que se ofrezcan funciones para disponer los elementos de la lista de acuerdo con criterios específicos, como el valor de los elementos. La función de comparación (cmpfunction) establece el procedimiento mediante el cual se comparan dos elementos en la lista, lo que a su vez determina el orden que adquieren.

¿Qué hace la función addLast()?

La función addLast tiene la capacidad de introducir un elemento en la posición final de la lista. En otras palabras, si la lista contiene una cantidad determinada de elementos, esta función resultará en un incremento de un solo elemento en el tamaño total de la lista. El recién incorporado elemento ocupará la posición n-1 dentro de la lista.

¿Qué hace la función getElement()?

La función getElement devuelve el elemento ubicado en una posición concreta. Para lograr esto, es esencial proporcionar los parámetros correspondientes a la función, los cuales consisten en el nombre de la lista y la posición (representada por un valor entero) en la que se halla el elemento. En consecuencia, el tiempo de ejecución necesario para esta operación es O(1), lo que significa que su eficiencia no varía en función del tamaño de la lista.

¿Qué hace la función subList()?

Es posible que esta función se utilice para obtener una parte más pequeña de elementos de una lista más grande, aunque no hay información específica en el curso al respecto. En la mayoría de los lenguajes de programación, una sublista es una sección continua de elementos extraída de la lista original. Por lo general, la función subList requeriría parámetros como el índice de inicio y el índice de fin. Al utilizar estos parámetros, la función crearía y devolvería una nueva lista que contiene los elementos de la lista original que caen dentro de ese rango particular.

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()?

La función "iterator" en el contexto de la función "printAuthorData(author)" tiene la finalidad de recorrer la lista de libros asociados a un autor en particular. Básicamente, esta función se encarga de examinar la lista que contiene toda la información de los libros, incluyendo los nombres de los autores y sus respectivas calificaciones. Cada vez que identifica elementos relacionados con el autor al que el usuario hace referencia, incrementa un contador que registra cuántos libros han sido escritos por dicho autor, según los registros disponibles.

Por otro lado, "printBestBooks" nos brinda la capacidad de recorrer la lista completa de libros y resaltar aquellos con las calificaciones más altas otorgadas por los lectores.

¿Observó algún cambio en el comportamiento del programa al cambiar el valor del

parámetro 'datastructure' en la creación de las listas?, explique con sus propias

palabras los cambios que notó.

No se observó ninguna alteración en el funcionamiento del programa al realizar las pruebas en la consola; se obtuvieron los mismos resultados para las distintas opciones. Esto se debe a que el TAD NewCatalog está diseñado para operar de manera consistente sobre cualquiera de las dos estructuras de datos. A partir de esta observación, podemos inferir que en la función NewCatalog, donde se lleva a cabo la creación e inicialización del catálogo con la información completa de los libros, el comportamiento general del programa no se vería afectado simplemente al cambiar la forma en que los libros se almacenan.No obstante, es importante destacar que, dependiendo de la estructura de datos empleada, el orden de crecimiento temporal de las operaciones puede variar.