Annie Yen

Java Programming III

Assignment 1

August 5, 2018

**Software System Design Document**

Overview:

* InventoryMethods
* InventoryItem
* InventoryView
* InventoryController
* InventoryModel
* InventoryMain
* Multimedia
* Book
* CD
* DVD

**InventoryMethods**

InventoryMethods interface contain methods to get and perform CRUD operations (Create, Retrieve, Update and Delete) on string objects representing inventory items.

**InventoryItem**

Abstract class InventoryItem implements interface InventoryMethods. The interface methods being implemented are getItem, createItem, updateItem and deleteItem. The getItem method simply loads and prints the property files, which contain key-value strings representing media items in inventory. The other interface methods pass string parameters for adding new items to the inventory (createItem), updating items currently in the inventory (updateItem), and removing items from the inventory (deleteItem). The string key-value pairs being manipulated by these methods all treat the title of a media item as the key that is paired with other string values (such as name of artist or author, production, price, etc). Each key-value pair is viewed as a row in the property file that has every string value separated by vertical mark delimiter (‘|’) representing column in each row.

The forming of key-value pairs, columns representing each inventory item, are done by createItem and updateItem method. Both methods require the value string to be first delimited with vertical mark (‘|’), and then manipulated with, setProperty and getProperty method to manipulate properties object. Treat such object as the persistent Hashtable that stores key-value string pairs. The loadPropTable and storePropTable are helper methods for reading and writing the contents of property table. They are essentially the file manipulation method for writing an output stream and reading an input stream. The strings for forming this Hashtable as a text file are also the parameters of constructor.

The constructor has three parameters: a property object as persistent Hashtable, a string representation of text file, and a string representation of file header. The constructor is invoked in the subclasses that extended InventoryItem, which are Book, CD and DVD. Each subclass also implements its own retrieveItem method, which is a method from interface InventoryMethods not being implemented by the abstract InventoryItem class. This method retrieveItem is implemented separately by subclasses, so that each subclass has its own unique output string listing different columns (such as ISBN digits for Book, studio name for DVD, etc). These columns are specified in the enum Multimedia.

**Multimedia**

The enum class Multimedia has a constructor that sets string parameters to define the enum values of media type, whereas the media type can be Book, CD and DVD and the values are columns of each type. The class also contains private instances to set the accessor methods for accessing the media type and the corresponding columns. The accessors are for the InventoryModel class to accessed when communicating with InventoryView, ensuring that the accessing of key-value property strings is hidden in the model component of MVC design. The other two components of MVC, view and controller, can only use methods to get the key-value property pairs from the model, ensuring that only model can accessed the property information.

**InventoryView, InventoryController, InventoryModel**

InventoryView uses Scanner to read user input, using setModelSelection in InventoryController to call the setSelection in InventoryModel. The setSelection in InventoryModel method sets Multimedia enum objects of Book, CD, and DVD according to the user’s selection; in which, the enum contains the type of media and the corresponding columns. In response, the newView method in InventoryView calls model.getItem to load the properties. The model.getSelection and model.getColumns are called from InventoryModel class to pass the user selection of media and its columns.

Once the properties of selected media are listed, the view then prompts the user for operation to create, retrieve, update, or delete. The controller.setOperation found in InventoryView passes the string representing key entered by the user to the InventoryModel via InventoryController. It has an overloading method that takes another string along with an integer, indicating the column selected by the user. For instance, when the user selects to update, string corresponding to new column entry and an integer corresponding to the column, are passed to method setOperation along with the string parameter corresponding to the key.

Once the InventoryModel receives the integer from controller, indicating that the user has selected an operation to perform, the InventoryView is also notified with the integer. In response, the newView method of InventoryView calls the perfromOperation() method of InventoryModel to perform interface methods createItem, retriveItem, updateItem, or deleteItem.

**InventoryMain**

InventoryMain contains the main method for the inventory application, using the viewMain method of InventoryView to initialize. It connects the view, controller and model components of design pattern, completing the MVC design architecture. The connection between these components are formed as the following: using the setModel method of InventoryView, InventoryController reference in InventoryView, InventoryModel reference in InventoryController,and setView method of InventoryModel.