|  |  |
| --- | --- |
| Student ID | **D10126532** |
| Student Name | **John Warde** |
| Student Email | [**john.warde@gmail.com**](mailto:john.warde@gmail.com)  [**john.warde@mydit.ie**](mailto:john.warde@mydit.ie) |
| Student Phone | **086-353-7309** |
| Title | **Continuous Assessment – Part 1 – Library System** |

# Background

As I have never programmed in Java professionally and having only competed a training course for Sun Certified Java Programmer – I decided to use the console/command line for user interaction. On hindsight, it may have been better to learn basic Swing because in this project I created a “Console UI Framework”.

# Running

Follow these steps to run the library system:

1. Expanding the zip file to a local folder
2. Open up a command window,
3. Change folder to the root folder of expanded files.
4. Paste in the following command:  
   **java -classpath bin/library.jar library.Library**
5. Javadoc generated HTML documentation is located within the **javadoc** folder

# Class and Interface Descriptions

Below are brief descriptions of each class in this project.

## Library Class

Stub class for application start-up.

## LibraryItem Class

Abstract class which Book, Dvd and Periodical concrete classes are derived from. Implements the OutputStrategy interface to provide a consistent calling mechanism for display to the console – an example of the Strategy pattern.

## Book Class

Inherits from LibraryItem parent class, contains methods to specific to it properties.

## Periodical Class

Inherits from LibraryItem parent class, contains methods to specific to it properties.

## Dvd Class

Inherits from LibraryItem parent class, contains methods to specific to it properties.

## Catalog Class

Maintains a list of all library items. Implements the Traverser interface, an example of the Iterator pattern, needs a better name and one that doesn’t clash with Java’s Iterator interface.

## CatalogViewer Class

Creates views or sub-sets of a catalog can be created by this class i.e. for listing all items on loan

## Loans Class

Records individual loans for library items, for each loaned out item and entry exist with the associated user/library ID. A separate loans class was used in the design to reflect more flexible design if data persistence were to be implemented in the furure.

## LibraryRepository Class

This is a singleton class, it is the data store for all the elements of this library system i.e. users, library items and loans.

## LibraryItemFactory Class

This is used to help populate the initial library Catalog, it takes input from a Comma Seperated Values type structure and returns one of the concrete child classes of LibraryItem, it is an example of the Factory pattern.

## Members Class

This is the “data store” for all users of the library it also uses the Traverser interface to iteration over users.

## User Class

Contains details for a single user.

## OutputStrategy Interface

Implemented by the concrete child classes of the LibraryItem class – forces them to implement method calls for two display modes. If a new child class of Library Item was introduced then, it would also need to implement these methods.

## Traverser Interface

Uses the iterator pattern to provide a consistent iterable access to Library and User objects, the name needs to be reviewed.

## LibraryManagerConsole class

Manages the main console application including displaying menus, lists, forms and managing the user input to these. This class and the rest of the classes listed below combine to provide a console user interface to the library “business objects”. I believe there is enough separation to be able to create a Graphics User Interface with the library “business objects”.

The following classes are used to manage the console UI elements such as Menus, due to time constraints; I cannot list out the full details of all these classes.

ConsoleComponent – abstract class which the following classes derive from  
Confirmation  
Form  
FormField  
FormFieldDate  
FormFieldInt  
FormFieldString  
Confirmation  
ListChoice  
ListContextMenu  
Menu  
Prompt

# Summary & Conclusions

Due to time constraints I was not able to complete this project and documentation to my satisfaction, including all the jUnit tests.