

**MSc Computer Science**

**School of Computing, Engineering and Digital technologies**

**Object-oriented Programming**

**ICA Element 2: Development Report**

**Name: Yaswanth Sai Chinthakayala**

**Student ID: W9640628**

Table of Contents

[1. Discussion of GUI Application 3](#_Toc134483152)

[1.1Overview 3](#_Toc134483153)

[1.2 GUI Application 3](#_Toc134483154)

[1.3 Abstract Table Model 4](#_Toc134483155)

[1.4 Main Class Code 6](#_Toc134483156)

[1.5 Testing 6](#_Toc134483157)

[2. Discussion of Enhanced Application 7](#_Toc134483158)

[2.1 Overview 7](#_Toc134483159)

[2.2 Displaying an Image 7](#_Toc134483160)

[2.3 Adding and buying multiple items. 10](#_Toc134483161)

[2.4 Testing 12](#_Toc134483162)

[3. Discussion of merged application 13](#_Toc134483163)

[3.1 Overview 13](#_Toc134483164)

[3.2 Adapter Pattern 13](#_Toc134483165)

[3.3 Testing 15](#_Toc134483166)

[4. Discussion of Improvement 17](#_Toc134483167)

[4.1 Overview 17](#_Toc134483168)

[4.2 Adding to Cart 17](#_Toc134483169)

[5. References: 18](#_Toc134483170)

# 1. Discussion of GUI Application

## 1.1Overview

In this step, GUI application needs to be developed and modified from task 3 in element 1.

* The task 3 is a console-based application, to transform it into GUI, most of the code remain same and some of the code sections need to change.
* The arraylist data should be loaded into JTable and displayed.
* Adding items, buying items from the shop system needs to be enabled.

Table : Use of components from previous element

|  |  |  |
| --- | --- | --- |
| Console component | Functionality | Change for GUI Version |
| ASCStockItem.java | Data class | No change |
| stockItemsList | ArrayList for ASCStockItems | No change |
| loadData () | Loads input data to arraylist | No change |
| saveData () | Saves the arraylist data to a file | No change |
| buyItem () | Buy an item from stock | Needs to be modified according to JTable |
| addStock () | Add an item to the stock | Needs to be modified according to JTable |

## 1.2 GUI Application

Graphic User Interface design needs to be created in the design tab of NetBeans. It shows how our application will be displayed to the users in the end. The designed GUI is shown in the figure below.

Graphical user interface, application

Description automatically generated

Figure : Blueprint of GUI

The elements present in the GUI are:

Table : GUI Elements

|  |  |
| --- | --- |
| Swing Items | Names |
| JTable | ascStockItem |
| JButtons | buyButton,  addButton,  buyXButton,  quitButton |
| Jpanel | photoPanel |
| JLabels | PhotoLabel,  itemLabel |

First, a JFrame class is created in the package and then on top of it all the components are placed. A JTable is created next, then all the jButtons are laid down. Next, a panel named photoPanel is built, In the panel two labels are used. PhotoLabel is used to display the image of the selected item and itemLabel is used to show the name of the item that got chosen.

## 1.3 Abstract Table Model

In the previous tasks, arraylist is used for storing and displaying items. But the JTable cannot display the items in arraylist directly. AbstractTableModel is needed for interpreting the values of arraylist into the JTable. The abstract class has many default methods that TableModel have, and all the listeners are maintained in this model. The abstract table model only needs three methods to be overridden to work.

Graphical user interface, text, application

Description automatically generated

Figure : ASC Table Model

The model that is used here is ascTableModel which extends to AbstractTableModel abstract class. The code is shown in the figure above. The constructor also needs the column names for the table and arraylist as parameters.

Graphical user interface, text, application

Description automatically generated

Figure : Updating rows for Table model.

The data is updated row wise by looping through the arraylist. Rows will be incremented until there are no items left.

## 1.4 Main Class Code

Graphical user interface, text, application, email

Description automatically generated

Figure : Setting abstract model to JTable

In the main class, a table model named ascModel is created. This model has set to the JTable ascStockItem that is designed in GUI. By looping through the column array the table headers are being set.

## 1.5 Testing

* The JTable is populated with data of AsherSportsConsortium3.csv.

Graphical user interface

Description automatically generated

Figure : JTable with data

* The button functionality for Add, Buy and Quit button is also added. Screenshots are attached below by performing all the operations.

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, application

Description automatically generated

Figure : Sale Confirmation Figure : Added Stock Confirmation Figure : Quit Confirmation

# 2. Discussion of Enhanced Application

## 2.1 Overview

After developing the GUI, enhancements need to be done to include more features to the application. The overview of the improvements is:

* Images needs to be displayed in the photoLabel.
* Low stock Items should be notified if there are less than 5 units left for an item.
* With the help of buyX button and addY buttons, a feature needs to be enabled where multiple items can be bought and added to stock respectively.

## 2.2 Displaying an Image

**Loading Images:**

A method named loadData() is created, which preloads the images. The preloading of images helps the images to loader faster in the application. Initially, all the photos are placed in folder and that location is used.

Graphical user interface

Description automatically generated with medium confidence

Figure : Load image method

The method is started by looping through the length of arrayList. Each file name is same as the product Id with jpg extension. The photo path is loaded into filename and then loaded into an arrayList.

Text, application

Description automatically generated

Figure : Adding photos to photoList

The first block will try to read the images and for any exceptions caught, simply the logger reports the error. Finally, all the images will be loaded to photoList array.

**Display Method:**

The display image method is called in listener event. The parameter that passed is the current selected row in the JTable. The photo and item labels are cleared first with setText. Then the image is selected from photoList.

**Graphical user interface, text, application, email

Description automatically generated**

Figure : Displaying Image

The image and text labels are set accordingly If the image is not null, the same will be displayed or else image is not available is displayed.

Graphical user interface, text, application

Description automatically generated

Figure : Adding text to photo and Item Labels

**List Event Listener**

To display images of any selected row, firstly the event called “list selection listener” should be implemented. This listener event is used when the value of a table or list is modified.

As shown in the figure, getSelectionModel method is used to add this listener to the table. If the selection gets changed the row can be selected and then with the help of the help of displayImage () method, the image is displayed.

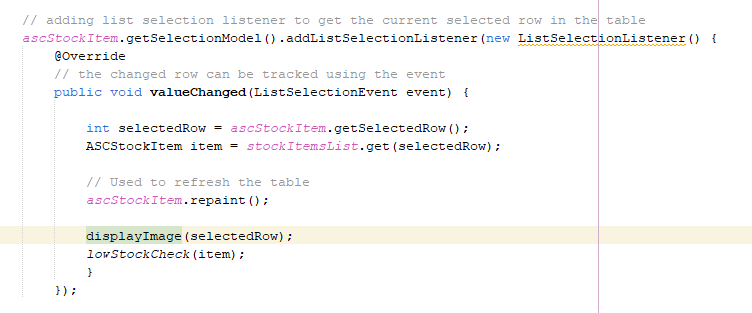


Figure : List selection listener

## 2.3 Adding and buying multiple items.

The features like adding and buying multiple items can be achieved by methods buyX and addY. Most of the code is same for both adding and buying items. The only change is the stock is decremented in buyX() and the stock is added in addY().

Graphical user interface, text, application

Description automatically generated

Figure : BuyX method

If the buyX button is pressed without selecting an item, an error message is displayed as shown in the above code. Next, if an item which has zero stock is bought popup notification is displayed “Item out of Stock”.

Graphical user interface, text, application

Description automatically generated

Figure : Updating arrayList and JTable

In the above figure, Initially the range of values is limited to length of arrayList. Next, popup is displayed using the JOptionPane and the option is selected. The selected option is decremented, and the value is set to table and arrayList.

Text

Description automatically generated

Figure : Sale Confirmation

In the end, an information message is displayed about the confirmation of sale and low stock check is done.

**Low Stock Check:**

The stock is checked by using getStock() getter function. If the item stock falls less than 5 then low stock warning is displayed.

Text

Description automatically generated

Figure : Low stock check

## 2.4 Testing

* The image is displayed when a row is selected, if that item stock is less, warning message will also be displayed.

Graphical user interface, application

Description automatically generated

Figure : Image and warning are displayed.

* Using buyX button displays a popup message with number of items to buy. After buying information message is displayed.

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application, chat or text message

Description automatically generated

Figure : Buy multiple items. Figure : Sale Confirmation

* Adding stock

Graphical user interface, text, application, chat or text message

Description automatically generated Graphical user interface, text, application

Description automatically generated

Figure : Add multiple items. Figure : Added Stock Confirmation

# 3. Discussion of merged application

## 3.1 Overview

* Asher Sport’s Consortium needs to be integrated with Teesside Skates which is a basic application that only have basic features.
* Program needs to be modified in such a way that it works for both asc and Teesside Skate products.

## 3.2 Adapter Pattern

A design pattern needs to be used for solving the problem of integrating the two different classes. Adapter pattern will be used in this merge. This pattern is a structural pattern which combines two incompatible classes.

Primarily, the TSProduct.java class needs to be refactor copied to the ASCSystem project. Then, Intermediary class is created, it is AdaptedTSProduct which extends to ASCStockITem as shown in the below figure.

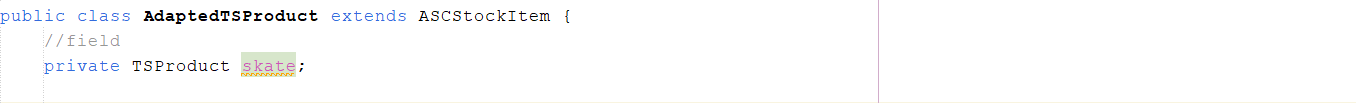


Figure : AdaptedTSProduct class

For the constructor all the parameters of TSProducts.java must be present in this as well. The base class needs to be initialized first. The parameters can be matched in this case, if all the parameters cannot be matched, override methods also can be used.

Table : ASC Parameters in terms of TS Parameters

|  |  |
| --- | --- |
| ASCStockItem parameters match | TSProduct parameters |
| Product Id | num |
| Title | Make + mdl |
| Description | notes |
| Price in Pounds | (int) price |
| Price in pence | (Int) (price – (int)price) \* 100 |

The super class can be initiated with the values as shown in the above table. The implementation can be seen in below figure. The tricky part of code is to convert the price (float) into pounds(int) and pence(int). The pounds will be equal to integer part of price. For pence, the pounds can be subtracted and multiplied by 100.

Text

Description automatically generated

Figure : Initialization of base class

The skate product is similar to TSProduct. All parameters which are in ascSystem are defined. Therefore, overriding methods is not necessary. AdapterTSProduct is ready to apply.

**Main class code:**

The loading of data uses two classes, one for ASC data and other for TS data, the only change that needed is when trying to load items into the arraylist, AdaptedTSProduct should be used.

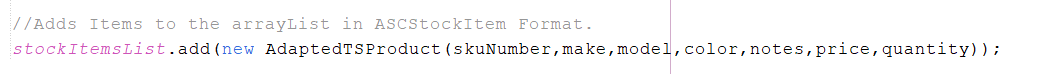


Figure : Adding items to arrayList using Adapted Class

## 3.3 Testing

* The ts\_products.txt and AsherSportsConsortium3.csv data is loaded into the table and images should be displayed.

Input data in files:

ASC Data

A screenshot of a computer

Description automatically generated with low confidence

Figure : ASC Data

TS Data

Text, letter

Description automatically generated

Figure : TS Data

The combined data files loaded in GUI.

Graphical user interface, application

Description automatically generated

Figure : Merged Application

* The output file format does not change, only new entries from TS Products are added.

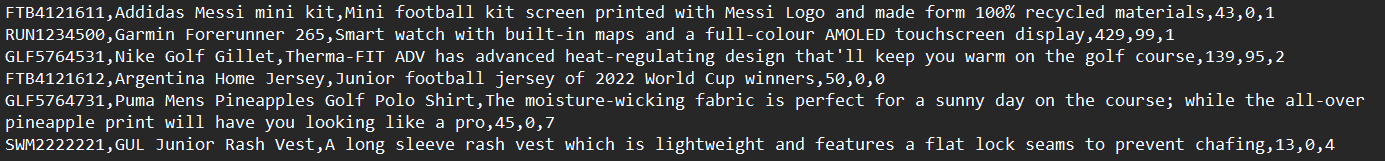


Figure : Output file

# 4. Discussion of Improvement

## 4.1 Overview

More features can be added to Asher Sports Consortium. The first improvement that comes to mind when thinking of improvements is cart function which enables users to add various items to cart and they can buy all the items at once.

## 4.2 Adding to Cart

A button will be created and placed at the bottom of JTable with name addToCart. When an item is selected, the item can be added to cart as many times as needed. Various items can also be selected and added to cart.

All the items in the cart can be bought at once. The hardest part is there should be track of all the items that are adding into the cart. An approach that can be used is arrayList named cartList. When the items are added to cart, they are automatically added to cartList.

Text

Description automatically generated

Figure : Rough Code

After the cart is selected, when the items are bought the cartList will be used to update both arrayList of StockItemsList as well as JTable.

# 5. References:

Oracle Help Center. (n.d.). *JDK 20 Documentation*. [online] Available at: https://docs.oracle.com/javase/8/docs/api/javax/swing/table/AbstractTableModel.html. [Accessed 1 May 2023].

netbeans.apache.org. (n.d.). *Handling Images in a Java GUI Application*. [online] Available at: https://netbeans.apache.org/kb/docs/java/gui-image-display [Accessed 1 May 2023].

‌

docs.oracle.com. (n.d.). *How to Write a List Selection Listener* [online] Available at: https://docs.oracle.com/javase/tutorial/uiswing/events/listselectionlistener.html [Accessed 1 May 2023].

‌www.tabnine.com. (n.d.). *ListSelectionListener java code examples | Tabnine*. [online] Available at: https://www.tabnine.com/code/java/classes/javax.swing.event.ListSelectionListener [Accessed 1 May 2023].

‌