**Documentation on File Explorer.**

Kazi Samin Yeaser (1405103)

**Introduction:**

The assignment was to make a File Explorer using java and Design pattern. Name of my project is myFileExplorer. It contains total eleven classes including one main class(pro3.java) and a controller class(FXMLDocumentController.java) which controls the FXML file (FXMLDocument.fxml) . Apart from those I wrote 9 classes in this project. Some of them follow one or more of the four design patterns we were taught in the class. The description of the classes and applications of the design patterns are given below.

**Classes:**

**1. myClass.** Created for combining necessary information to create TableView.

**Included methods.** Only getters;

**2. myViewInterface.** An Interface. Created for make switching between views easier. It has two sub classes.

**Included methods.** alterView();

**3. EditDetailsView.** A sub class of myViewInterface. Created for creating Details View.

**Included methods. 1.** setColumns() : initialize columns.

**2.** alterView() : override function.

3. getCNG() : creating or getting the singleton object.

**4. EditTilesView.** A sub class of myViewInterface. Created for creating Tiles View.

**Included methods. 1.** setTilePane() : initialize tilepane

**2.** alterView() : override function.

**3**. getCNG() : creating or getting the singleton object.

**4**. creatMyVbox(File,FileName): create adapter class.

**5.factoryVewChanger.** A factory class deciding which subclass of myViewInterface is to creat.

**Included methods. 1.** getFactory(): creating or getting the singleton object.

**2.** getView : factoryfunction.

**6.myIconView.** Singleton class to create Imageview.

**Included methods. 1.** fileIconView(File,Size) : Creating imageview from files.

**2.** getView(File f, String S): creating or getting singleton object. Also used as a factory function returning three types of imageview.

**7.myTreeItems.** Adapter class for TreeItem class.  
 **Included methods. 1.** getChildren(): overriding function.

**2.** isLeaf(): overriding function.

**3**. toString() :overriding function.

**4**. buildChildren(TreeItem<String> TreeItem,File f) : Used for creating and adding children in treeitems.

**8.MyTableCell.** Adapter class for TableCell class.

**Included methods. 1.** updateItem(ImageView , boolean): Overriding function.

**9.myVbox.** Adapter class for Vbox.

**Included methods. 1.** toString() : Overriding function.

**Design Patterns Used:**

**Factory design pattern:**

1. Factory design pattern is used in “factoryVewChanger” class. It Takes a Boolean variable as input and returns the “EditDetailsView” class or the “EditTilesView” class.

2. Factory design is used in “myIconView” class. It takes input a file and a String. Based on the string it creates three types of ImageView extracted from file icon.

**Singleton pattern:**

1.Singleton pattern is used in myIconView class as I didn’t need multiple objects of this class. And need to distinguish between it’s work done in first creation and latter.

2. Singleton pattern is used in factoryVewChanger class as I didn’t need multiple objects of this class.

3. Singleton pattern is used in “EditDetailsView” class as I didn’t need multiple objects of this class. And need to distinguish between work done in it’s first creation and latter uses.

4. Singleton pattern is used in “EditTilesView” class as I didn’t need multiple objects of this class. And need to distinguish between work done in it’s first creation and latter .

**Composite pattern:**

1. TreeView API uses this pattern.
2. TilePane API uses this pattern.
3. ScrollPane API uses this pattern.

**Adapter pattern:**

1. MyTableCell Class Adapts ImageView into TableCell. So I used Adapter pattern in myTableCell where MyTableCell is the adapter, ImageView is the Adoptee, and TableCell API is the target.

2. Adapter pattern is used in MyVbox, as adapter where VBox is target and ImageView is the adoptee.

3. Adapter pattern is used in myTreeItems class where treeItem was the target and File was adoptee.