BANGLADESH UNIVERSITY OF ENGINEERING & TECHNOLOGY

Topics: ***Design Pattern Implementation***

Project : ***File Explorer***

Prepared By,

Name : Rifat Rahman

Roll : 1405007

Department : CSE

Level : 3 Term : 1

Date Of Submission

30/04/2017

Classes & Methods

1.FileInformation:

This class store the information about size, name , Icon, modify_date of folder. The folder works as the Node of the tree. This class has some methods:

a.FileInformation() [Constructor]

b. public FileInformation(ImageView icon, double size, String modify_date, String name)

c.String getFileName()

d. ImageView getFileIcon()

e. Double getFileSize()

f. String getModify_ddate()

g. void setFileName(String name)

h. void setFileSize(Double size)

I . void setFileIcon(ImageView icon)

j. void setModifyDate(String modify_date)

2. ***Controller_ Select:***

Creates new action listeners for new vboxes. In TileView we can control by clicking on the folder. It creates new action for clicking in Tile View .The methods are:

a. controller_Select(VBox vbox)

b. void run_action_listener()

3.***Update_Modify:***

Contains singleton observable list and also provides the only function for updating elements in tiles view or table view in file explorer.

The functions are:

a. private Update_Modify()

b. static void update_items(File f,TreeItem<File> temp)

4.Shape:

This is an interface. This interface contains empty method. The method is:

a. ImageView sel (Image img)

This method is for changing the shape of icon.

5.Short:

This class implements Shape interface. In this class “sel” method is overridden. This class is for small icon. Method :

a. ImageView sel (Image img)

6.Medium:

This class implements Shape interface. In this class “sel” method is overridden. This class is for medium icon. Method :

a. ImageView sel (Image img)

7.Large:

This class implements Shape interface. In this class “sel” method is overridden. This class is for large icon. Method :

a. ImageView sel (Image img)

8.ImageShape:

This class contains a method which returns the image of the folder. In Tile View there are three kinds of views. Large icon, medium icon, small icon are different kinds of view.

a. Shape get_image(int point)

9.Tile_Vbox:

This class is used for to get the icon and name to situate in Vbox. This kind of class is for the tile view.

a. Tile_Vbox(VBox box,ImageView image,Label l)

10.File_Explorer:

The main method includes in this class. In main method we declare objects of scrollpane, tableview, tilePane etc classes. We also declare objects of button class for tilesView and tableView. The localhost name is the root of the tree of treePane.

a. Void main(String args[])

Design Patterns

1.Factory pattern:

Factory pattern is one of most used design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

In Factory pattern, we create objects without exposing the creation logic to the client and refer to newly created object using a common interface.

Implementation:

Class ***ImageShape***

We build a ***Shape*** interface for the shape of icon. Class ***Short, Medium, Large*** implement the ***Shape*** class and override ***sel(Image img)*** function which returns ImageView. In function get_image(int point) takes point as different size code and returns the desired shape class. By using sel(Image img) method we can get the desired icon. We use this to see different shape in tile view.

2.Singleton pattern:

Singleton pattern is one of the simplest design patterns in Java. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

This pattern involves a single class which is responsible to create an object while making sure that only single object gets created. This class provides a way to access its only object which can be accessed directly without instantiating the object of the class.

Implementation:

Class ***Update_Modify***

As the constructor of this class is private we can use only one observable list. As a result only one Observable list can be shared by tiles view, table view and tree view.

Class ***File_Explorer***

In this class we declare object of treePane and TilePane as static, As a result we can use only one tree and tablepane in our explorer.

3.Composite Pattern:

Composite pattern is used where we need to treat a group of objects in similar way as a single object. Composite pattern composes objects in term of a tree structure to represent part as well as whole hierarchy. This type of design pattern comes under structural pattern as this pattern creates a tree structure of a group of objects.

This pattern creates a class that contains a group of its own objects. This class provides ways to modify its group of same objects.

We are demonstrating use of composite pattern via following example in which we will show employees hierarchy of an organization.

Implementation:

Class ***File_Explorer***

If we observe the treePane a TreeItem is composed to some other TreeItems. Some TreeItem can be leaf in this pattern.

Again folders are composed to some other folders. There are also some leaf folders. By using isDirectory() method we can assure whether a folder has children or not.

4.Adapter Pattern:

Adapter pattern works as a bridge between two incompatible interfaces. This type of design pattern comes under structural pattern as this pattern combines the capability of two independent interfaces.

This pattern involves a single class which is responsible to join functionalities of independent or incompatible interfaces. A real life example could be a case of card reader which acts as an adapter between memory card and a laptop. You plugin the memory card into card reader and card reader into the laptop so that memory card can be read via laptop.

We are demonstrating use of Adapter pattern via following example in which an audio player device can play mp3 files only and wants to use an advanced audio player capable of playing vlc and mp4 files.

Class ***Tile_Vbox***

In this class when we declare any object, the image and text get in to a Vbox. The Vbox interface turns into an interface.

Class ***File_Explorer***

By using the function tableColumn.setCellValueFactory(new PropertyValueFactory<>) we change the interface of a table colum manually. This is a kind of adapter pattern.