**Documentation on File Explorer**

**Classes:**

**1. App (main class):**

This is the main class containing only the main function *public static void main(String[] args)* where the constructor of the class Frame is called.

**2. Frame**

This class returns a single instance of the Jframe used for creating the interface of the software. The constructor is private so no more than one instance can be created. The components of the frame such as scrollpane, buttons and textfields are all added in the private constructor. This class represents **Singleton Pattern** as it does not allow more than one instance of the class to be created.

The only function of this class is:

*get\_instance()*

This method returns only one single jframe when this method is called in the main class;

**3. TreeView:**

The purpose of the TreeView class is to build up a tree starting from a given root folder. The tree includes all the subfolders present inside the root folder, their subfolders and so on. The nodes of the tree are created and their corresponding responses on clicking are set in this class.

The child folders are obtained and added to a parent folder through a depth first search in the *CreateNodes()* method. Objects and composition of objects are treated uniformly, hence the **Composite Pattern** is used here.

The methods of this class are:

*a)* *get\_tree()*

The private variable tree is accessed by other classes through this method.

*b)* *mousePressed (MouseEvent evt)*

This method writes the operation that takes place when a folder node in the tree is clicked with a mouse. This basically means the subfolders under the selected folder are displayed in a tabular or list format beside the tree.

*c)* *CreateNodes (DefaultMutableTreeNode top, File file)*

This method traces all the folders and subfolders in a node using depth first search. This method is a recursive function. The folders of the parent file is saved in an array, then the function is recursively called for each of the folders in the array.

**4. TableView:**

This class is responsible for tabular view of folders and the operations of the table created.

The methods are:

*a) get\_table()*

The private variable table is accessed by other classes through this method.

*b) mousePressed(MouseEvent evt)*

Whenever a row of the table is double clicked with a mouse, this method is called. The purpose of the method is to identify the row where user has clicked, get information from the row and show its child folders in the current view.

**5. ListView:**

This class is responsible for listed view of folders and the operations of the list created.

This class implements the API class *ListSelectionListener* and overrides the method *valueChanged (ListSelectionEvent lse)* to make it compatible with responses of list view, thus following the **Adapter Pattern.**

The methods are:

*a) get\_list()*

The private variable list is accessed by other classes through this method.

*b) valueChanged (ListSelectionEvent lse)*

Whenever a folder in the list is clicked with a mouse, this method is called. The purpose of the method is to identify the index in the list where user has clicked, get necessary information and show its child folders in the current view.

**6. L\_Renderer:**

The purpose of this class to place icons beside files in the list by overriding the *getListCellRendererComponent()* method of Java API class *DefaultListCellRenderer*. The *DefaultListCellRenderer* classdoes not allow icons by default. Here the class has been modified to adapt it to icons, following the **Adapter Pattern**.

The methods are:

*a) getListCellRendererComponent (JList list, Object value, int index, boolean isSelected, boolean cellHasFocus)*

This method in the *DefaultListCellRenderer* class is overridden here to add the appropriate icon beside each folder in list view. The method returns a label with the desired icon and text.

**Design Patterns:**

|  |  |
| --- | --- |
| **Design Pattern** | **Class** |
| Singleton Pattern | Frame |
| Composite Pattern | TreeView |
| Adapter Pattern | ListView |
| Adapter Pattern | L\_Renderer |
| Adapter Pattern | MouseAdapter (API) |
| Adapter Pattern | ActionListener (API) |
| Composite Pattern | JTable (API) |
| Composite Pattern | JFrame (API) |
| Composite Pattern | JScrollPane (API) |
| Composite Pattern | JSplitPane (API) |
| Composite Pattern | Container (API) |

**Submitted By:**

Name : Tasfia Zahin

Student ID: 1405041

Section : A2

Dept. of CSE

BUET