

# FILE EXPLORER

Fardous Ahmed(1405099)

## Introduction

File Explorer is an application that provides a graphical interface to access or browse the files and folders of a file system.

## Classes & corresponding functions

- **FileExplorer**

This is the main class. It contains several functions.

1. **createJMenu()** - creates a JMenu, which name is “Options”.
2. **displayTableView()** - creates a JMenuItem, which name is “List View” and adds an *actionListener* for it.
3. **displayListView()** - creates a JMenuItem, which name is “Tile View” and adds an *actionListener* for it.
4. **selectTableOrListView()** - selects between *List View* & *Tile View* by clicking *JMenuItem*.
5. **showFileDetails()** - shows file name and file path of the current directory.
6. **fileGUI()** - it is a container that holds all the components (Tree, List, Tile).
7. **main()** - main class, runs the project.

- **FileDetails**

This class contains a function. It sets file name & file path of the current directory.

1. **setFileDetails()** - sets file name & file path of the current directory.

- **FileTree**

This class creates a Tree on the left side to show file-folder hierarchy of the file system. This class extends **FileDetails** class to set the file name and file path of the current directory.

1. **getTreeObj()** - returns an object of *FileTree* class.
2. **createJTree()** - uses *DefaultTreeModel* to create the tree of the file system, sets the file system roots and nodes, puts the tree into *JTree*.
3. **showChildren()** - implements *SwingWorker* to add the files in the tree that are contained within the directory of the currently selected node.

- **FileTreeCellRenderer**

This class extends *DefaultTreeCellRenderer* class. It is a *TreeCellRenderer* for a file.

- **FileTable**

This class creates the table view. This class extends **FileDetails** class to set the file name and file path of the current directory.

1.**getTableObj()** - returns a object of *FileTable* class.

2.**createJTable()** - uses *FileTableModel* to create the table view of the file system, creates & initializes table rows, puts the rows into *JTable*.

3.**updateTableData()** - updates the data on the table.

- **FileTableModel**

This class extends *AbstractTableModel* to hold files. *FileTable* class uses it to set icon, name, size & last modified date of the files.

- **FileList**

This class creates the tile view. This class extends **FileDetails** class to set the file name and file path of the current directory.

1.**getListObj()** - returns a object of *FileList* class.

2.**createJList()** - uses *DefaultListModel* to create the tile view of the file system, creates & initializes it.

3.**updateListData()** - updates the data on the tile view.

- **FileListCellRenderer**

This class extends *DefaultListCellRenderer*. It is a *ListCellRenderer* For a file.

## Used Design Patterns

- **SINGLETON PATTERN**

The following classes use this design pattern. It ensures that any class has only one object.

1. **FileTree**
2. **FileTable**
3. **FileList**

- **FACTORY PATTERN**

*selectTableOrListView()* function in the **FileExplorer** class chooses any of the classes from **FileTable**, **FileList** which extends the class **FileDetails**.

- **COMPOSITE PATTERN**

The following class use this design pattern.

1. **FileTree**

- **ADAPTER PATTERN**

The following classes use this design pattern.

1. Client - **FileTree**  
Adapter - **JTree**  
Adaptee - **DefaultTreeModel**.
2. Client - **FileTable**  
Adapter - **JTable**  
Adaptee - **FileTableModel**.
3. Client - **FileList**  
Adapter - **JList**  
Adaptee - **DefaultListModel**.