Documentation On File Explorer

YEASEEN ARAFAT
#1405091
April 18,2016

1 *Description of Classes:*

Here naames and uses of each class.

1. **Main**:
   (a) Runs a Graphical User Interface.
   (b) Loads a FXML file.
2. **Controller**: Functions and uses of this class.
   (a) initialize();
      i. Creates a TreeView of Folders and Files of a File System.
      ii. Creates a TilesView of Folders and Files corresponding to that TreeView.
      iii. Describing setOnAction functions of different types of Buttons.
   (b) setExpandProperty();
      i. Expands the tree items successors by clicking them on.
   (c) treeIdChildren();
      i. Adds the tree items children.
   (d) adress();
      i. Finds the adress of a tree item.
   (e) expandFolder();
      i. Provides a TilesView or TableView of a tree item by clicking on it.
3. **Views:** Interface
   (a) Definition of a draw() function for GoGrid and Tables classes.
4. **GoGrid:** Functions and uses of this class.
   (a) draw();
      i. Implementation of the Views class by exposing draw() function.
   (b) gridView();
      i. Creates a TilesView of a File System corresponding the Path.
5. **Folder** Basically it extends the VBox class of JAVA API.
6. **Tables:** Functions and uses of this class.
   (a) draw();
      i. Implementation of the Views class by exposing draw() function.
   (b) showFilesTableView();
      i. Creates a TableView of a File System corresponding the Path.
7. **FactoryView:** Functions and uses of this class.
   (a) getView();
      i. Provies a GoGrid object or Tables object corresponding an input instruction.
8. **AllInfo:** Stores the TableView components in this class.

2 *Design Patterns:*

In this File System I am using two different Design Patterns. These are,

1. Factory Pattern.
2. Compsite Pattern.
3. Adapter Pattern.

Here, I am describing where I used these.

2.1 Uses of Factory Pattern

In class-based programming, the factory method pattern is a creational pattern that uses factory methods to deal with the problem of creating objects without having to specify the exact class of the object that will be created. This is done by creating objects by calling a factory methodeither specified in an interface and implemented by child classes, or implemented in a base class and optionally overridden by derived classesrather than by calling a constructor.

Classes related to Factory Pattern:

- Views: Interface
- GoGrid:
- Tables:
- FactoryView:
- Controller:

Process of using Factory Pattern:

1. Created a Views:Interface and defines a function called draw();
2. Then created a class named GoGrid which implements the Views:Interface and overrides draw() function for showing TilesView of File System in GUI.
3. Then created a class named Tables which implements the Views:Interface and overrides draw() function for showing TableView of File System in GUI.
4. After that, I created a FactoryView: class to get a Views: object and to return it to the Controller class according to a input instruction.
5. In Controller class by clicking TilesView button or TableView button, I send a instruction by calling getView() function of FactoryView, then get a Views object corresponding input instruction and show this in the GUI.

2.2 Uses of Composite Pattern

The composite pattern describes a group of objects that is treated the same way as a single instance of the same type of object. The intent of a composite is to "compose" objects into tree structures to represent part-whole hierarchies.

I m using this pattern in Controller: Class.

Process of using Composite Pattern

1. Firstly, I created a root of a Folder tree.
2. Then I added its Children by creating treeIdChildren() fuction.
3. Again I added selected items successors by clicking on a selected treeItem.

2.3 Uses of Adapter Pattern

Adapter pattern provides to make existing classes work with others without modifying their source code.

I m using this pattern in GoGrid: Class.

Process of using Adapter Pattern

1. For this, I created a Folder: class which extends VBox which is a API of java.
2. Now, here I declare a String which stores the filename of each component of TilesView.
3. By adapting VBox, I used this String to explore the sub Folder of that Folder if it is a directory.

The END();
