



# 软件体系结构

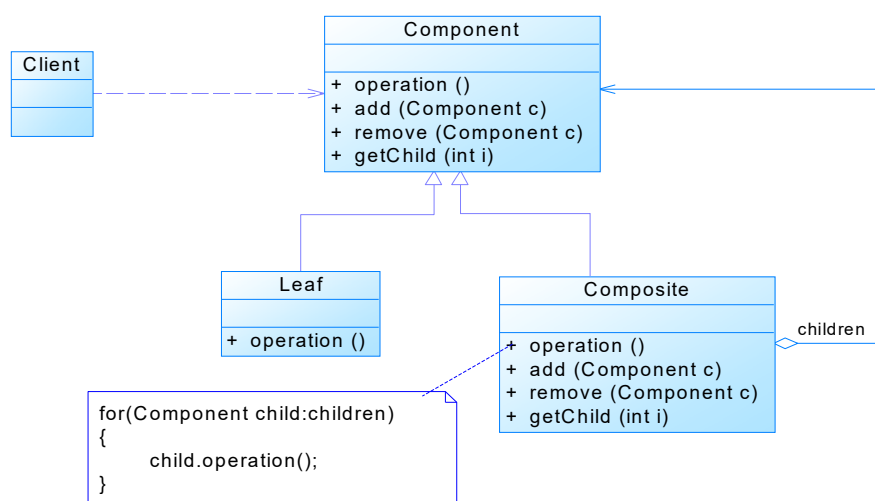
## 《软件体系结构作业十九》

学 号 22920212204396

姓 名 黄子安

2024 年 5 月 23 日

## 1、用 GUI 改写本例并访问你的计算机某个子目录



该例子使用了组合设计模式，该模式核心理念主要有两点：

- 定义了一个抽象构件类（Component），具体的元素（Leaf）和容器（Composite）都继承于它，所以它既可以代表叶子，又可以代表容器，而客户端针对该抽象构件类进行编程，无须知道它到底表示的是叶子还是容器，可以对其进行统一处理。
- 容器对象与抽象构件类之间还建立一个聚合关联关系，在容器对象中既可以包含叶子，也可以包含容器，以此实现递归组合，形成一个树形结构。

文件系统就是该例子的很好体现，代码可以对于给定的路径无须担心其是文件还是文件夹，只需要执行对应的操作（printList）即可，如果是文件则会输出对应的文件名，是文件夹就递归输出；另一方面文件夹里可以放文件，也可以再放文件夹，构成一个聚合关系。

之后使用 GUI 修改本例子，依旧采用 JavaFX 框架，先编写一个控制器类，这里需要使用 JavaFx 框架提供的 DirectoryChooser 来递归读取本地计算机指定目录并继续递归读取其中文件和文件夹，之后在 Composite Tree 中加入对应的结点，再调用 printList 即可通过组合设计模式完成输出操作，具体代码如下：

```

@FXML
public void selectButtonClick(ActionEvent actionEvent) {
    DirectoryChooser directoryChooser = new DirectoryChooser();
    directoryChooser.setTitle("Select Directory");
    File selectedDirectory = directoryChooser.showDialog(stage);

    if (selectedDirectory != null) {
        currentFolderPath = selectedDirectory.getAbsolutePath();
        label.setText("Current Directory: " + currentFolderPath);
        root = new Directory(selectedDirectory.getName());

        buildTree(selectedDirectory, root);

        textArea.clear();
        textArea.setText(root.printList());
    }
}

```

```

private void buildTree(File file, Directory directory) {
    File[] files = file.listFiles();
    if (files != null) {
        for (File f : files) {
            if (f.isDirectory()) {
                Directory subDirectory = new Directory(f.getName());
                directory.add(subDirectory);
                buildTree(f, subDirectory);
            } else {
                directory.add(new cn.xmu.edu.composite.
                    File(f.getName(), (int) f.length()));
            }
        }
    }
}

```

对于新建文件夹和新建文件，需要查询 Composite Tree，找到对应的节点

```

private Entry findFolder(Entry entry, String folderName) {
    if (entry instanceof Directory directory) {
        if (directory.getName().equals(folderName)) {
            return directory;
        } else {
            for (Object item : directory.getDirectory()) {
                if (item instanceof Entry) {
                    Entry found = findFolder((Entry) item, folderName);
                    if (found != null) {
                        return found;
                    }
                }
            }
        }
    }
    return null;
}

```

新建文件操作通过输入框输入文件名，对于文件已存在或者文件名为空的特殊情况提示对应的报错，之后通过 Java IO 提供的文件操作来新建一个文件

```
@FXML
public void onCreateFileButtonClick(ActionEvent actionEvent) {
    if (currentFolderPath.isEmpty()) {
        label.setText("Please select a directory first");
        return;
    }

    TextInputDialog fileNameDialog = new TextInputDialog();
    fileNameDialog.setTitle("Create File");
    fileNameDialog.setHeaderText("Enter file name");
    Optional<String> fileNameResult = fileNameDialog.showAndWait();

    if (fileNameResult.isPresent()) {
        String fileName = fileNameResult.get();
        File newFile = new File(currentFolderPath, fileName);

        if (newFile.exists()) {
            label.setText("File already exists");
            return;
        }

        try (FileWriter fileWriter = new FileWriter(newFile)) {
            fileWriter.write("");
            fileWriter.flush();
            int fileSize = 0;

            // Find the current folder in the composite tree and add the new file
            Entry currentFolder = findFolder(root,
                new File(currentFolderPath).getName());
            if (currentFolder instanceof Directory) {
                ((Directory) currentFolder)
                    .add(new cn.xmu.edu.composite.File(fileName, fileSize));
            } else {
                label.setText("Could not find the current folder");
                return;
            }

            textArea.clear();
            textArea.setText(root.printList());
        } catch (IOException e) {
            e.printStackTrace();
        }
    } else {
        label.setText("File name cannot be empty");
    }
}
```

## 新建文件夹操作同理

```
@FXML
public void onCreateFolderButtonClick(ActionEvent actionEvent) {
    String currentFolderPath = label.getText().replace("Current Directory: ", "");
    if (currentFolderPath.isEmpty()) {
        label.setText("Please select a directory first");
        return;
    }

    TextInputDialog folderNameDialog = new TextInputDialog();
    folderNameDialog.setTitle("Create Folder");
    folderNameDialog.setHeaderText("Enter folder name");
    Optional<String> folderNameResult = folderNameDialog.showAndWait();

    if (folderNameResult.isPresent()) {
        String folderName = folderNameResult.get();
        File newFolder = new File(currentFolderPath, folderName);

        if (!newFolder.exists()) {
            if (newFolder.mkdir()) {
                Directory newDirectory = new Directory(folderName);

                Entry currentFolder = findFolder(root,
                                                    new File(currentFolderPath).getName());
                if (currentFolder instanceof Directory) {
                    ((Directory) currentFolder).add(newDirectory);
                } else {
                    label.setText("Could not find the current folder");
                    return;
                }

                textArea.clear();
                textArea.setText(root.printList());
            } else {
                label.setText("Failed to create folder");
            }
        } else {
            label.setText("Folder already exists");
        }
    } else {
        label.setText("Folder name cannot be empty");
    }
}
```

最后通过 FXML 文件设计对应的界面 UI，另外使用一个 CSS 文件美化

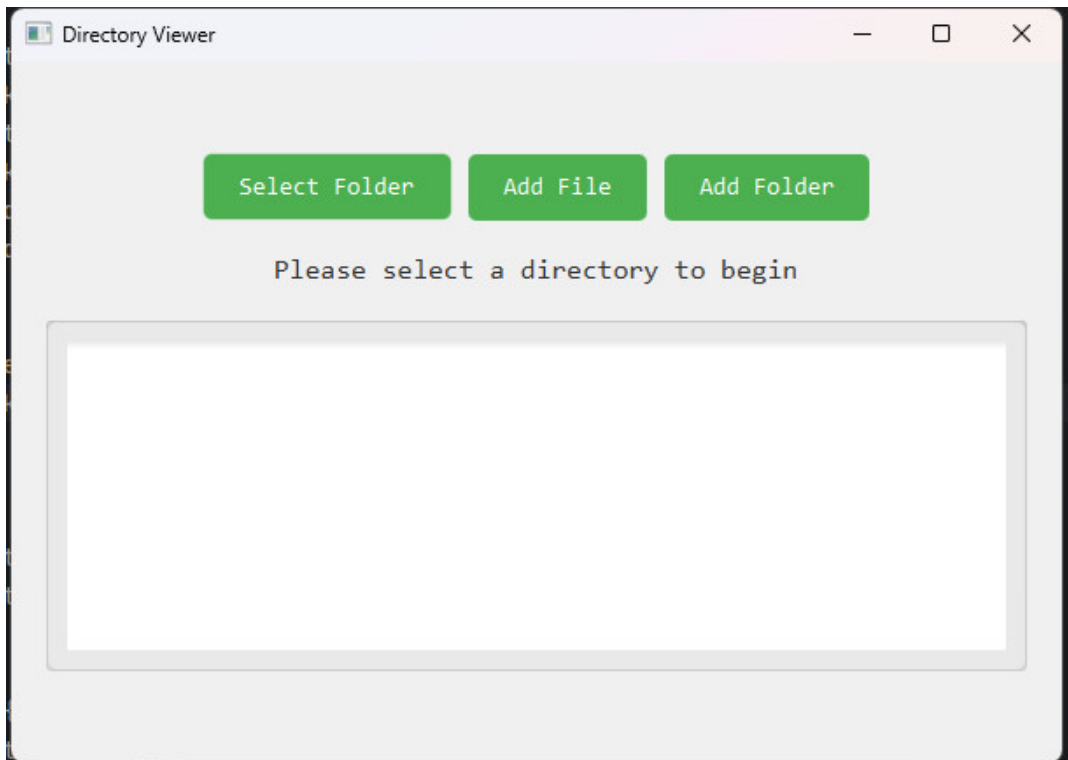
```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.*?>
<?import javafx.scene.control.*?>
<?import javafx.scene.layout.*?>
<?import javafx.scene.text.*?>

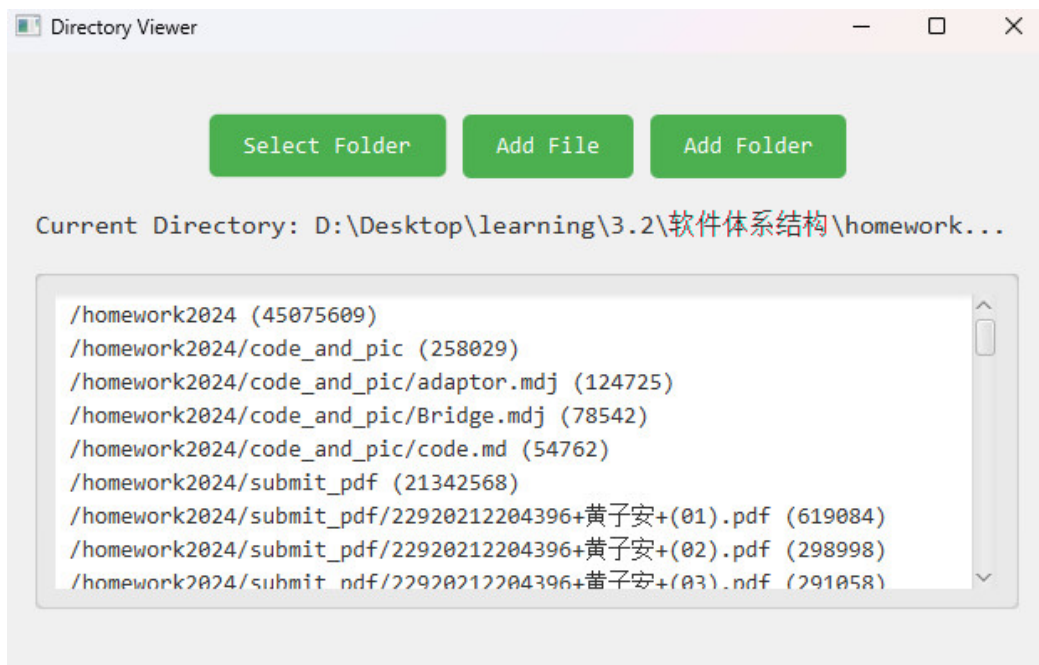
<?import java.net.URL?>
<VBox alignment="CENTER" spacing="20.0" xmlns="http://javafx.com/javafx/17.0.2-ea"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="cn.xmu.edu.composite.CompositeController">
    <padding>
        <Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />
    </padding>
    <stylesheets>
        <URL value="@styles.css" />
    </stylesheets>
    <HBox prefHeight="20.0" prefWidth="200.0" alignment="CENTER" spacing="10">
        <children>
            <Button onAction="#selectButtonClick" text="Select Folder" />
            <Button onAction="#onCreateFileButtonClick" text="Add File" />
            <Button onAction="#onCreateFoldButtonClick" text="Add Folder" />
        </children>
    </HBox>
    <Label fx:id="label" text="Please select a directory to begin" />
    <TextArea fx:id="textArea" prefHeight="200.0" prefWidth="400.0" wrapText="true" />
</VBox>
```

```
.root {
    -fx-font-family: 'Consolas';
    -fx-background-color: #f0f0f0;
}
.button {
    -fx-font-size: 14px;
    -fx-background-color: #4CAF50;
    -fx-text-fill: white;
    -fx-background-radius: 5;
    -fx-border-radius: 5;
    -fx-padding: 10 20;
}
.button:hover {
    -fx-background-color: #45a049;
}
.label {
    -fx-font-size: 16px;
    -fx-text-fill: #333;
}
.text-area {
    -fx-font-size: 14px;
    -fx-border-color: #ccc;
    -fx-border-width: 1;
    -fx-border-radius: 5;
    -fx-padding: 10;
}
```

运行后的最终界面如下所示：



递归输出 SA 作业的目录，输出其中的子文件夹和文件





在当前目录下新增一个文件夹

