

Homework:

1、 用 GUI 改写本例并访问你的计算机某个子目录。

答：对于本题，我使用了 JavaFX 进行图形化界面设计，我设计的程序运行逻辑如下：

- 1) 用户点击“打开目录”按钮，通过 DirectoryChooser 选择一个目录。
- 2) 程序遍历所选目录及其子目录和文件，生成类似于 Linux tree 命令的目录结构字符串，并显示在文本区域中。

程序主界面如图 1 所示：



图 1 目录查看器主界面

UIController 类调用了其他类来实现目录和文件结构的显示。

- 用户选择目录：当用户点击“打开目录”按钮时，handleOpenDirectory 方法被调用。
- 创建目录和文件对象：handleOpenDirectory 方法使用 DirectoryChooser 选择目录，并创建一个 Directory 对象表示根目录。
- 递归添加子目录和文件：addFilesToDirectory 方法递归地将文件和子目录添加到根目录对象中。
- 显示目录结构：调用根目录对象的 toTreeString 方法生成目录结构字符串，并将其显示在文本区域中。

本程序使用的设计模式是组合模式，它允许将对象组合成树形结构来表示“部分-整体”的层次结构。组合模式使得用户对单个对象和组合对象的使用具有一致性。在这个程序中，Directory 和 MyFile 类共同实现了组合模式。

- Component（组件）：抽象类 Entry 定义了目录和文件的公共接口。
- Leaf（叶子节点）：MyFile 类是叶子节点，表示文件。
- Composite（组合节点）：类 Directory 是组合节点，表示目录，它可以包含子目录或文件。

UIController 类主要方法代码如下所示：

```
16 private void handleOpenDirectory() {
17     DirectoryChooser directoryChooser = new DirectoryChooser();
18     directoryChooser.setTitle("选择目录");
19     File selectedDirectory = directoryChooser.showDialog(new Stage());
20
21     if (selectedDirectory != null) {
22         Directory root = new Directory(selectedDirectory.getName());
23         addFilesToDirectory(root, selectedDirectory);
24         textArea.setText(root.toTreeString( prefix: ""));
25     }
26 }
```

```

28 @ private void addFilesToDirectory(Directory directory, File file) {
29     for (File f : file.listFiles()) {
30         if (f.isDirectory()) {
31             Directory subDirectory = new Directory(f.getName());
32             directory.add(subDirectory);
33             addFilesToDirectory(subDirectory, f);
34         } else {
35             directory.add(new MyFile(f.getName(), (int) f.length()));
36         }
37     }
38 }

```

程序运行结果如图 2 和图 3 所示，可以看出程序正常运行，并计算出目录和文件大小：

```

C:\Users\ayu>tree F:\大三下学期\软件体系结构\课件
卷 新加卷 的文件夹 PATH 列表
卷序列号为 00000052 C63D:6458
F:\大三下学期\软件体系结构\课件
├─阅读材料
│   └─test
C:\Users\ayu>

```

图 2 使用 cmd 的 tree 指令打印访问的子目录结构

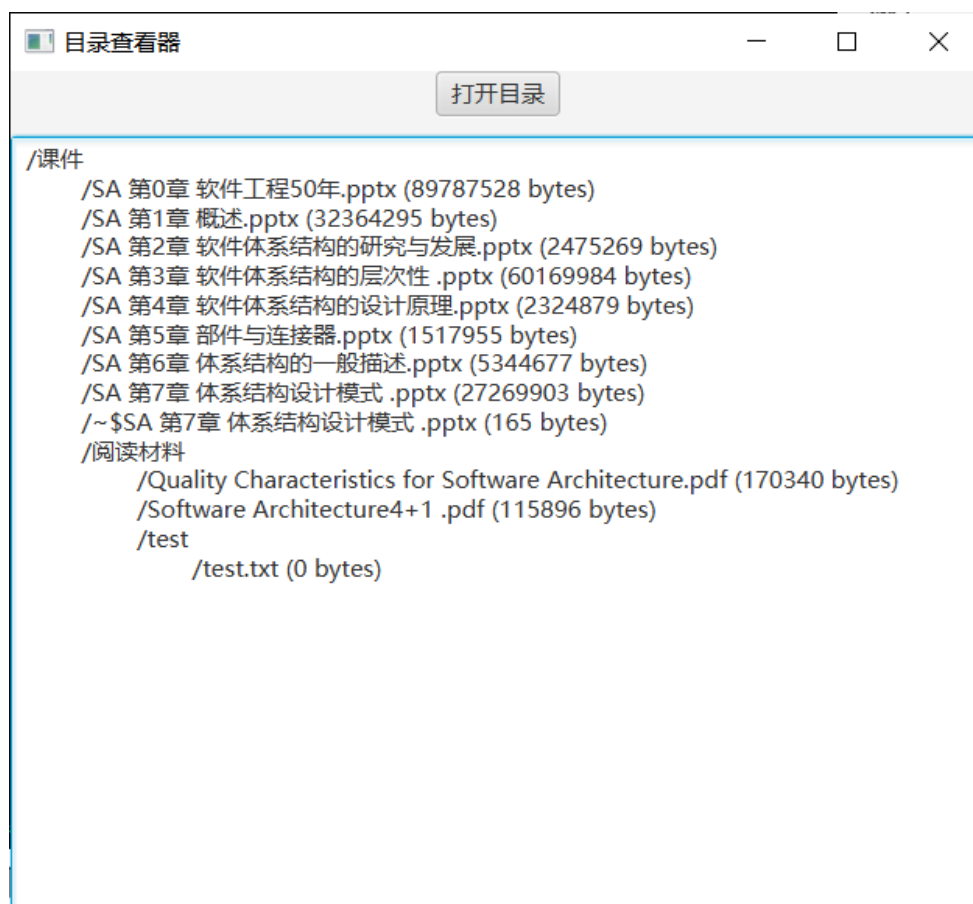


图 3 使用程序打开访问的子目录，观察结构

2、 附录

1) Main.java

```
package com.example.composite;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.io.IOException;

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) throws IOException {
        FXMLLoader loader = new
FXMLLoader(getClass().getResource("ui.fxml"));
        Scene scene = new Scene(loader.load());
        primaryStage.setScene(scene);
        primaryStage.setTitle("目录查看器");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```

2) UIController.java

```
package com.example.composite;

import javafx.fxml.FXML;
import javafx.stage.DirectoryChooser;
import javafx.stage.Stage;
import javafx.scene.control.TextArea;

import java.io.File;

public class UIController {

    @FXML
    private TextArea textArea;

    @FXML
    private void handleOpenDirectory() {
```

```

        DirectoryChooser directoryChooser = new DirectoryChooser();
        directoryChooser.setTitle("选择目录");
        File selectedDirectory = directoryChooser.showDialog(new
Stage());

        if (selectedDirectory != null) {
            Directory root = new
Directory(selectedDirectory.getName());
            addFilesToDirectory(root, selectedDirectory);
            textArea.setText(root.toTreeString(""));
        }
    }

    private void addFilesToDirectory(Directory directory, File file)
    {
        for (File f : file.listFiles()) {
            if (f.isDirectory()) {
                Directory subDirectory = new Directory(f.getName());
                directory.add(subDirectory);
                addFilesToDirectory(subDirectory, f);
            } else {
                directory.add(new MyFile(f.getName(), (int)
f.length()));
            }
        }
    }
}

```

3) ui.fxml

```

<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.TextArea?>
<?import javafx.scene.layout.VBox?>

<VBox spacing="10" alignment="CENTER"
xmlns="http://javafx.com/javafx/8"
xmlns:fx="http://javafx.com/fxml/1"
fx:controller="com.example.composite.UIController">
    <Button text="打开目录" onAction="#handleOpenDirectory"/>
    <TextArea fx:id="textArea" prefWidth="400" prefHeight="400"/>
</VBox>

```

4) Entry.java

```
package com.example.composite;

public abstract class Entry {
    public abstract String getName();
    public abstract int getSize();
    public Entry add(Entry entry) throws FileTreatmentException {
        throw new FileTreatmentException();
    }
    public void printList() {
        printList("");
    }
    protected abstract void printList(String prefix);
    public abstract String toTreeString(String prefix);
    public String toString() {
        return getName() + " (" + getSize() + ")";
    }
}
```

5) Directory.java

```
package com.example.composite;

import java.util.Iterator;
import java.util.Vector;

public class Directory extends Entry {
    private String name;
    private Vector<Entry> directory = new Vector<>();

    public Directory(String name) {
        this.name = name;
    }

    public String getName() {
        return this.name;
    }

    public int getSize() {
        int size = 0;
        Entry entry;
        for (Iterator<Entry> it = this.directory.iterator();
it.hasNext(); size += entry.getSize()) {
            entry = it.next();
        }
    }
}
```

```

    }
    return size;
}

public Entry add(Entry entry) {
    this.directory.add(entry);
    return this;
}

protected void printList(String prefix) {
    System.out.println(prefix + "/" + this);
    Iterator<Entry> it = this.directory.iterator();
    while (it.hasNext()) {
        Entry entry = it.next();
        entry.printList(prefix + "/" + this.name);
    }
}

public String toTreeString(String prefix) {
    StringBuilder builder = new StringBuilder();
    builder.append(prefix).append("/").append(getName()).append("\n");
    for (Entry entry : directory) {
        builder.append(entry.toTreeString(prefix + "\t"));
    }
    return builder.toString();
}

public String toString() {
    return getName() + " (" + getSize() + " bytes)";
}
}

```

6) Myfile.java

```

package com.example.composite;

public class MyFile extends Entry {
    private String name;
    private int size;

    public MyFile(String name, int size) {
        this.name = name;
    }
}

```

```

        this.size = size;
    }

    public String getName() {
        return this.name;
    }

    public int getSize() {
        return this.size;
    }

    protected void printList(String prefix) {
        System.out.println(prefix + "/" + this);
    }

    public String toTreeString(String prefix) {
        return prefix + "/" + getName() + " (" + getSize() + "
bytes)\n";
    }

    public String toString() {
        return getName() + " (" + getSize() + " bytes)";
    }
}

```

7) FileTreatmentException.java

```

package com.example.composite;

public class FileTreatmentException extends RuntimeException {
    public FileTreatmentException() {
    }

    public FileTreatmentException(String msg) {
        super(msg);
    }
}

```

8) 程序结构

