Министерство науки и высшего образования Российской Федерации

Муромский институт (филиал)

Федерального государственного бюджетного образовательного учреждения высшего образования

«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»

Факультет ИТР

Кафедра ПИн

*Лабораторная №7*

По Разработка кроссплатформенных приложений

Тема Файловый менеджер

Руководитель

Кульков Я.Ю.

(фамилия, инициалы)

(подпись) (дата)

Студент ПИн-121

(группа)

Ермилов М.В.

(фамилия, инициалы)

(подпись) (дата)

Муром 2024

**Лабораторная №7**

**Тема:** Файловый менеджер

**Цель:** изучить дополнительные компоненты JavaFX, а также способы объединения разметок на примере разработки файлового менеджера.

**Задачи:**   
1. Изучить работу полученного из примера приложения

1. Реализовать обработчик кнопки удаления файлов
2. Реализовать обработчик кнопки перемещения файлов
3. Реализовать все действия через меню приложения

**Ход работы:**

**Код из файла MainApplication.java:**package com.example.work;

import javafx.application.Application;

import javafx.fxml.FXMLLoader;

import javafx.scene.Scene;

import javafx.stage.Stage;

import java.io.IOException;

public class MainApplication extends Application {

@Override

public void start(Stage stage) throws IOException {

FXMLLoader fxmlLoader = new FXMLLoader(MainApplication.class.getResource("main-view.fxml"));

Scene scene = new Scene(fxmlLoader.load(), 1 , 1);

stage.setMinWidth(1400);

stage.setMinHeight(400);

stage.setTitle("FileManager");

stage.setScene(scene);

stage.show();

}

public static void main(String[] args) {

launch();

}

}

**Код из файла MainController.java:**package com.example.work;

import javafx.application.Platform;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.Alert;

import javafx.scene.control.ButtonType;

import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

public class MainController {

public PanelController leftPanelController, rightPanelController;

public PanelController src = null, dst = null;

void setSrcAndDst(){

if(leftPanelController.getSelectedFileName()==null &&

rightPanelController.getSelectedFileName() == null){

Alert alert = new Alert(Alert.AlertType.ERROR, "File is not choice", ButtonType.OK);

alert.showAndWait();

return;

}

if(leftPanelController.getSelectedFileName() != null){

src = leftPanelController;

dst = rightPanelController;

}

else {

src = rightPanelController;

dst = leftPanelController;

}

}

@FXML

void onCopy(ActionEvent event){

setSrcAndDst();

Path srcPath = Paths.get(src.getCurrentPath(), src.getSelectedFileName());

Path dstPath = Paths.get(dst.getCurrentPath()).resolve(srcPath.getFileName());

try {

Files.copy(srcPath, dstPath);

dst.updateList(Paths.get(dst.getCurrentPath()));

src.updateList(Paths.get(src.getCurrentPath()));

} catch (IOException e) {

Alert alert = new Alert(Alert.AlertType.ERROR, "Error of copying", ButtonType.OK);

alert.showAndWait();

}

}

@FXML

void onMove(){

setSrcAndDst();

Path srcPath = Paths.get(src.getCurrentPath(), src.getSelectedFileName());

Path dstPath = Paths.get(dst.getCurrentPath()).resolve(srcPath.getFileName());

try {

Files.move(srcPath, dstPath);

dst.updateList(Paths.get(dst.getCurrentPath()));

src.updateList(Paths.get(src.getCurrentPath()));

} catch (IOException e) {

Alert alert = new Alert(Alert.AlertType.ERROR, "Error of moving", ButtonType.OK);

alert.showAndWait();

}

}

@FXML

void onDelete(){

setSrcAndDst();

Path srcPath = Paths.get(src.getCurrentPath(), src.getSelectedFileName());

try {

Files.delete(srcPath);

src.updateList(Paths.get(src.getCurrentPath()));

} catch (IOException e) {

Alert alert = new Alert(Alert.AlertType.ERROR, "Error of deleting", ButtonType.OK);

alert.showAndWait();

}

}

@FXML

void onExit(){

Platform.exit();

}

}

**Код из файла PanelController.java:**package com.example.work;

import javafx.beans.property.SimpleObjectProperty;

import javafx.beans.property.SimpleStringProperty;

import javafx.event.ActionEvent;

import javafx.fxml.FXML;

import javafx.scene.control.\*;

import javafx.scene.input.MouseEvent;

import java.io.IOException;

import java.nio.file.FileSystems;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.time.format.DateTimeFormatter;

import java.util.stream.Collectors;

public class PanelController {

@FXML

private ComboBox<String> diskBox;

@FXML

private TableColumn<FileInfo, String> fileNameColumn;

@FXML

private TableColumn<FileInfo, Long> fileSizeColumn;

@FXML

private TableColumn<FileInfo, String> fileTimeColumn;

@FXML

private TableColumn<FileInfo, String> fileTypeColumn;

@FXML

private TableView<FileInfo> filesTable;

@FXML

private TextField pathField;

@FXML

void onPassUp(ActionEvent event) {

Path upperPath = Paths.get(pathField.getText()).getParent();

if(upperPath != null){

updateList(upperPath);

}

}

@FXML

void onSelectDisk(ActionEvent event) {

updateList(Paths.get(diskBox.getSelectionModel().getSelectedItem()));

}

@FXML

public void initialize(){

fileTypeColumn.setCellValueFactory(param -> new

SimpleStringProperty(param.getValue().getType().getName()));

fileNameColumn.setCellValueFactory(param -> new

SimpleStringProperty(param.getValue().getFileName()));

fileSizeColumn.setCellValueFactory(param -> new

SimpleObjectProperty<Long>(param.getValue().getSize()));

fileSizeColumn.setCellFactory(column -> new TableCell<>(){

@Override

protected void updateItem(Long size, boolean empty) {

super.updateItem(size, empty);

if(size==null || empty){

setText(null);

setStyle("");

}

else{

String text = String.format("%,d байт", size);

if(size==-1){

text = "[DIR]";

}

setText(text);

}

}

});

DateTimeFormatter dtf = DateTimeFormatter.ofPattern("dd.MM.YY HH:mm");

fileTimeColumn.setCellValueFactory(param -> new

SimpleStringProperty(param.getValue().getLastModified().format(dtf)));

filesTable.getSortOrder().addAll(fileTypeColumn,fileNameColumn,fileSizeColumn,fileTimeColumn);

diskBox.getItems().clear();

for(Path p : FileSystems.getDefault().getRootDirectories()){

diskBox.getItems().add(p.toString());

}

diskBox.getSelectionModel().select(1);

updateList(Paths.get("."));

}

protected void updateList(Path path) {

filesTable.getItems().clear();

pathField.setText(path.normalize().toAbsolutePath().toString());

try {

filesTable.getItems().addAll(

Files.list(path).map(FileInfo::new)

.collect(Collectors.toList())

);

} catch (IOException e) {

Alert alert = new Alert(Alert.AlertType.WARNING, "Unable to refresh list of files",ButtonType.OK);

alert.showAndWait();

}

filesTable.sort();

}

@FXML

void onTable(MouseEvent event) {

if(event.getClickCount() == 2){

Path path = Paths.get(pathField.getText())

.resolve(filesTable.getSelectionModel().getSelectedItem().getFileName());

if(Files.isDirectory(path)){

updateList(path);

}

}

}

public String getSelectedFileName(){

if(!filesTable.isFocused()){

return null;

}

return filesTable.getSelectionModel().getSelectedItem().getFileName();

}

public String getCurrentPath(){

return pathField.getText();

}

}

**Код из файла FileInfo.java:**package com.example.work;

import java.io.File;

import java.io.IOException;

import java.nio.file.Files;

import java.nio.file.Path;

import java.time.LocalDateTime;

import java.time.ZoneOffset;

public class FileInfo {

public enum FileType {

FILE("F"), DIRECTORY("D");

private String name;

FileType(String name) {

this.name = name;

}

public String getName() {

return name;

}

}

private FileType type;

public FileInfo(Path path) {

try {

this.fileName = path.getFileName().toString();

this.size = Files.size(path);

this.type = Files.isDirectory(path) ? FileType.DIRECTORY : FileType.FILE;

if(type==FileType.DIRECTORY){

size=-1L;

}

this.lastModified = LocalDateTime.ofInstant(Files.getLastModifiedTime(path).toInstant(),

ZoneOffset.ofHours(3));

} catch (IOException e) {

throw new RuntimeException("Unable to get file info");

}

}

private String fileName;

private long size;

private LocalDateTime lastModified;

public FileType getType() {

return type;

}

public void setType(FileType type) {

this.type = type;

}

public String getFileName() {

return fileName;

}

public void setFileName(String fileName) {

this.fileName = fileName;

}

public long getSize() {

return size;

}

public void setSize(long size) {

this.size = size;

}

public LocalDateTime getLastModified() {

return lastModified;

}

public void setLastModified(LocalDateTime lastModified) {

this.lastModified = lastModified;

}

}

**Код из файла main-view.fxml:**<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.Menu?>

<?import javafx.scene.control.MenuBar?>

<?import javafx.scene.control.MenuItem?>

<?import javafx.scene.input.KeyCodeCombination?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.VBox?>

<VBox maxHeight="-Infinity" maxWidth="-Infinity" xmlns="http://javafx.com/javafx/21"

xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.work.MainController">

<children>

<MenuBar>

<menus>

<Menu mnemonicParsing="false" text="File">

<items>

<MenuItem mnemonicParsing="false" onAction="#onCopy" text="Copy">

<accelerator>

<KeyCodeCombination alt="UP" code="C" control="DOWN" meta="UP" shift="UP"

shortcut="UP" />

</accelerator></MenuItem>

<MenuItem mnemonicParsing="false" onAction="#onMove" text="Move">

<accelerator>

<KeyCodeCombination alt="UP" code="X" control="DOWN" meta="UP" shift="UP"

shortcut="UP" />

</accelerator></MenuItem>

<MenuItem mnemonicParsing="false" onAction="#onDelete" text="Delete">

<accelerator>

<KeyCodeCombination alt="UP" code="DELETE" control="DOWN" meta="UP"

shift="UP" shortcut="UP" />

</accelerator></MenuItem>

<MenuItem mnemonicParsing="false" onAction="#onExit" text="Exit">

<accelerator>

<KeyCodeCombination alt="UP" code="Q" control="DOWN" meta="UP" shift="UP"

shortcut="UP" />

</accelerator></MenuItem>

</items>

</Menu>

</menus>

</MenuBar>

<HBox prefHeight="100.0" prefWidth="200.0" VBox.vgrow="ALWAYS">

<fx:include fx:id="leftPanel" source="panel-view.fxml" HBox.hgrow="ALWAYS"

VBox.vgrow="ALWAYS" />

<fx:include fx:id="rightPanel" source="panel-view.fxml" HBox.hgrow="ALWAYS"

VBox.vgrow="ALWAYS" />

</HBox>

<HBox>

<children>

<Button focusTraversable="false" maxHeight="1.7976931348623157E308"

maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#onCopy" text="Copy"

HBox.hgrow="ALWAYS" />

<Button focusTraversable="false" maxHeight="1.7976931348623157E308"

maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#onMove" text="Move"

HBox.hgrow="ALWAYS" />

<Button focusTraversable="false" maxHeight="1.7976931348623157E308"

maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#onDelete" text="Delete"

HBox.hgrow="ALWAYS" />

<Button maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308"

mnemonicParsing="false" onAction="#onExit" text="Exit" HBox.hgrow="ALWAYS" />

</children>

</HBox>

</children>

</VBox>

**Код из файла panel-view.fxml:**<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>

<?import javafx.scene.control.Button?>

<?import javafx.scene.control.ComboBox?>

<?import javafx.scene.control.TableColumn?>

<?import javafx.scene.control.TableView?>

<?import javafx.scene.control.TextField?>

<?import javafx.scene.layout.HBox?>

<?import javafx.scene.layout.VBox?>

<VBox minHeight="250.0" minWidth="565.0" spacing="5.0" xmlns="http://javafx.com/javafx/21"

xmlns:fx="http://javafx.com/fxml/1" fx:controller="com.example.work.PanelController">

<children>

<HBox spacing="20.0">

<children>

<ComboBox fx:id="diskBox" minWidth="300.0" onAction="#onSelectDisk" HBox.hgrow="ALWAYS"

/>

<TextField fx:id="pathField" editable="false" HBox.hgrow="ALWAYS" />

<Button mnemonicParsing="false" onAction="#onPassUp" text="Вверх" HBox.hgrow="ALWAYS" />

</children>

</HBox>

<TableView fx:id="filesTable" onMouseClicked="#onTable" prefHeight="200.0" prefWidth="200.0"

VBox.vgrow="ALWAYS">

<columns>

<TableColumn fx:id="fileTypeColumn" minWidth="50.0" prefWidth="50.0" resizable="false"

text="Тип" />

<TableColumn fx:id="fileNameColumn" minWidth="100.0" prefWidth="-1.0" text="Имя файла" />

<TableColumn fx:id="fileSizeColumn" minWidth="100.0" prefWidth="100.0" resizable="false"

text="Размер" />

<TableColumn fx:id="fileTimeColumn" minWidth="150.0" prefWidth="150.0" resizable="false"

text="Время" />

</columns>

<columnResizePolicy>

<TableView fx:constant="CONSTRAINED\_RESIZE\_POLICY" />

</columnResizePolicy>

</TableView>

</children>

<padding>

<Insets bottom="5.0" left="5.0" right="5.0" top="5.0" />

</padding>

</VBox>

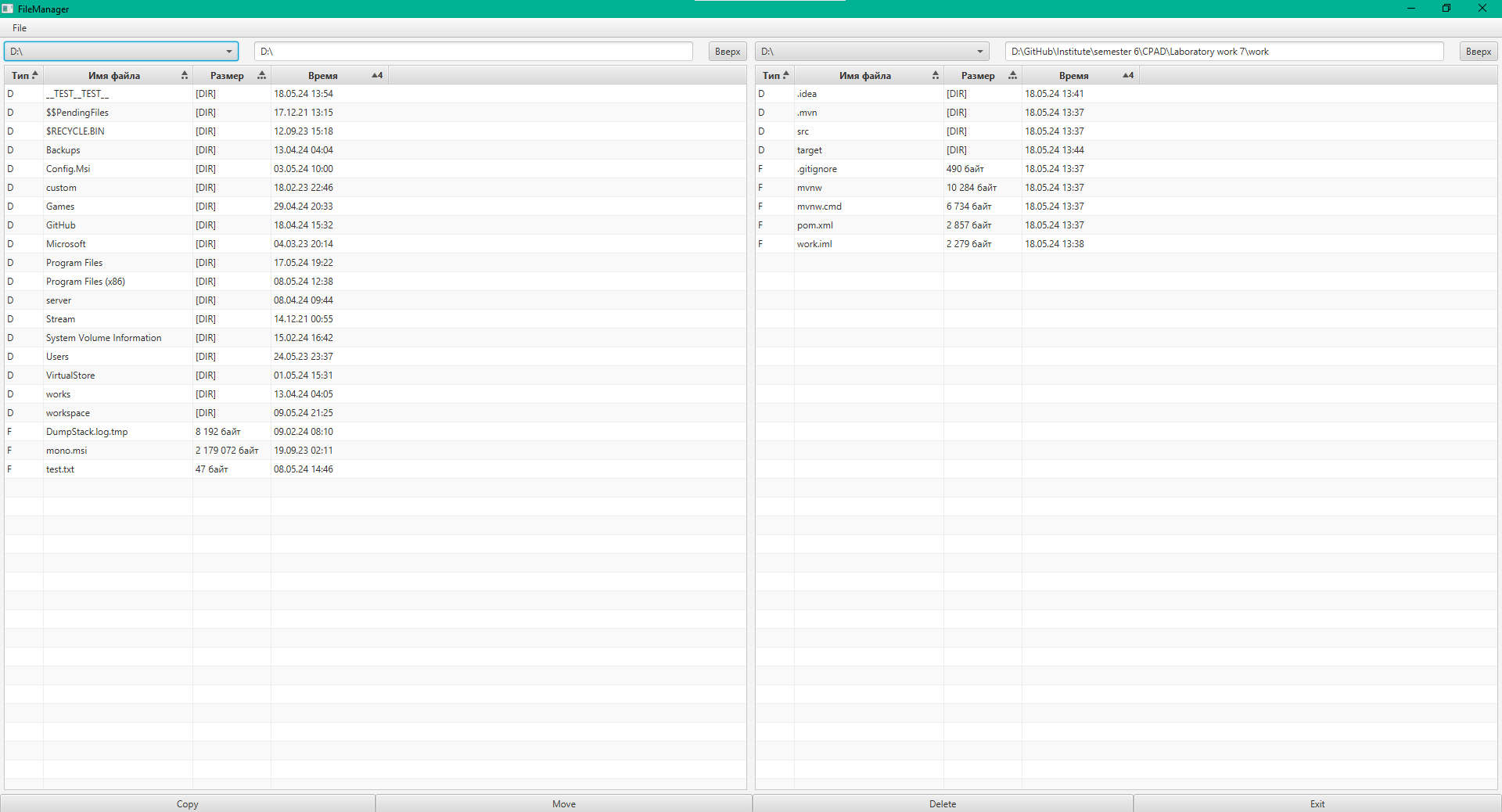


Рисунок 1 - пример работы программы