МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Севастопольский государственный университет»

кафедра Информационные системы

Сирота Марина Романовна

Институт информационных технологий и управления в технических системах

курс 3 группа ИС/б-32-о

09.03.02 Информационные системы и технологии (уровень бакалавриата)

ОТЧЕТ

по лабораторной работе №5

по дисциплине: «Платформа Java»

по теме: «Исследование возможностей разработки приложений с использованием платформы JAVAFX 2»

Отметка о зачете \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

ст. пр. Кузнецов С.А.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2018

1. ЦЕЛЬ РАБОТЫ

В ходе выполнения данной лабораторной работы необходимо ознакомиться с особенностями платформы JavaFX2 и приобрести практические навыки создания насыщенных пользовательских интерфейсов Java-программ.

1. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

С использованием компонентовJavaFX2 необходимо создать Java приложение реализующее добавление, редактирование и удаление данных заданного по варианту типа информации T(см. табл. 4.1).Данные отображать в виде таблицы. Реализовать поля ввода для добавления новых записей. Редактирование записей реализовать в таблице (использовать CellValueFactory). Предусмотреть возможность загрузки информации из текстового файла и сохранения в текстовый файл. Данные столбца N отображать в виде автоматически обновляющегося графика/диаграммы G(см. табл. 4.1).

|  |  |  |  |
| --- | --- | --- | --- |
| 2 | B | 3 | PieChart |

B: Ноутбук (Идентификатор модели, Производитель процессора, Тактовая частота процессора, Объем ОЗУ);

1. ВЫПОЛНЕНИЕ РАБОТЫ
   1. Листинг

package JV5;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.util.Scanner;

import javafx.application.Application;

import javafx.beans.property.SimpleStringProperty;

import javafx.collections.FXCollections;

import javafx.collections.ListChangeListener;

import javafx.collections.ObservableList;

import javafx.event.ActionEvent;

import javafx.event.EventHandler;

import javafx.geometry.Insets;

import javafx.scene.Group;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.control.Label;

import javafx.scene.control.TableColumn;

import javafx.scene.control.TableColumn.CellEditEvent;

import javafx.scene.control.TableView;

import javafx.scene.control.TextField;

import javafx.scene.control.cell.PropertyValueFactory;

import javafx.scene.control.cell.TextFieldTableCell;

import javafx.scene.layout.HBox;

import javafx.scene.layout.TilePane;

import javafx.scene.layout.VBox;

import javafx.scene.text.Font;

import javafx.stage.FileChooser;

import javafx.stage.Stage;

import javafx.scene.chart.\*;

public class TableViewExample extends Application {

private TableView<PC> table = new TableView<PC>();

private final ObservableList<PC> data = FXCollections.observableArrayList();

final HBox hb = new HBox();

ObservableList<PieChart.Data> pcData = FXCollections.observableArrayList();

PieChart chart = new PieChart();

public void UpdateChart() {

int i;

pcData.clear();

for (i = 0; i < data.size(); i++) {

pcData.add(new PieChart.Data(data.get(i).CPU.getValue().toString(),

Double.parseDouble(data.get(i).Fr.getValue().toString())));

}

chart.setData(pcData);

}

public static void main(String[] args) {

launch(args);

}

@Override

public void start(Stage stage) {

Scene scene = new Scene(new Group());

stage.setTitle("Table");

stage.setWidth(1050);

stage.setHeight(550);

final Label label = new Label("Ноутбуки");

label.setFont(new Font("Arial", 20));

label.setMaxWidth(300);

table.setEditable(true);

data.addListener(new ListChangeListener<PC>() {

@Override

public void onChanged(javafx.collections.ListChangeListener.Change<? extends PC> arg0) {

// TODO Auto-generated method stub

UpdateChart();

}

});

TableColumn idNameCol = new TableColumn("id");

idNameCol.setMinWidth(100);

idNameCol.setCellValueFactory(new PropertyValueFactory<PC, String>("ID"));

idNameCol.setCellFactory(TextFieldTableCell.forTableColumn());

idNameCol.setOnEditCommit(new EventHandler<CellEditEvent<PC, String>>() {

@Override

public void handle(CellEditEvent<PC, String> t) {

((PC) t.getTableView().getItems().get(t.getTablePosition().getRow()))

.setID(t.getNewValue());

}

});

TableColumn cpuNameCol = new TableColumn("Пр-ль");

cpuNameCol.setMinWidth(100);

cpuNameCol.setCellValueFactory(new PropertyValueFactory<PC, String>("CPU"));

cpuNameCol.setCellFactory(TextFieldTableCell.forTableColumn());

cpuNameCol.setOnEditCommit(new EventHandler<CellEditEvent<PC, String>>() {

@Override

public void handle(CellEditEvent<PC, String> t) {

((PC) t.getTableView().getItems().get(t.getTablePosition().getRow()))

.setCPU(t.getNewValue());

}

});

TableColumn frNameCol = new TableColumn("Частота");

frNameCol.setMinWidth(100);

frNameCol.setCellValueFactory(new PropertyValueFactory<PC, String>("Fr"));

frNameCol.setCellFactory(TextFieldTableCell.forTableColumn());

frNameCol.setOnEditCommit(new EventHandler<CellEditEvent<PC, String>>() {

@Override

public void handle(CellEditEvent<PC, String> t) {

((PC) t.getTableView().getItems().get(t.getTablePosition().getRow()))

.setFr(t.getNewValue());

}

});

TableColumn ramNameCol = new TableColumn("ОЗУ");

ramNameCol.setMinWidth(100);

ramNameCol.setCellValueFactory(new PropertyValueFactory<PC, String>("RAM"));

ramNameCol.setCellFactory(TextFieldTableCell.forTableColumn());

ramNameCol.setOnEditCommit(new EventHandler<CellEditEvent<PC, String>>() {

@Override

public void handle(CellEditEvent<PC, String> t) {

((PC) t.getTableView().getItems().get(t.getTablePosition().getRow()))

.setRam(t.getNewValue());

}

});

table.setItems(data);

table.getColumns().addAll(idNameCol, cpuNameCol,frNameCol, ramNameCol );

table.setMaxWidth(650);

final TextField addName = new TextField();

addName.setPromptText("ID");

addName.setMaxWidth(idNameCol.getPrefWidth());

final TextField addCPU = new TextField();

addCPU.setMaxWidth(cpuNameCol.getPrefWidth());

addCPU.setPromptText("Производитель");

final TextField addFr = new TextField();

addFr.setMaxWidth(frNameCol.getPrefWidth());

addFr.setPromptText("Частота");

final TextField addram = new TextField();

addram.setMaxWidth(ramNameCol.getPrefWidth());

addram.setPromptText("ОЗУ");

final Button addButton = new Button("Add");

addButton.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent e) {

data.add(new PC(addName.getText(), addCPU.getText(),addFr.getText(), addram.getText()));

addName.clear();

addCPU.clear();

addFr.clear();

addram.clear();

}

});

final Button downloadButton = new Button("Download");

downloadButton.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent e) {

FileChooser fileChooser = new FileChooser();

fileChooser.setTitle("Open Resource File");

File file = fileChooser.showOpenDialog(stage);

System.out.println("Открытие файла - : " + file.getAbsolutePath());

Scanner in = null;

try {

in = new Scanner(file);

in.useDelimiter(";");

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

while (in.hasNext()) {

PC tyty = new PC(in.next(), in.next(), in.next(), in.next());

data.add(tyty);

}

}

});

final Button saveButton = new Button("Save");

saveButton.setOnAction(new EventHandler<ActionEvent>() {

@Override

public void handle(ActionEvent e) {

FileChooser fileChooser = new FileChooser();

fileChooser.setTitle("Файл для сохранения");

File file = fileChooser.showOpenDialog(stage);

System.out.println("Сохранение в файл - : " + file.getAbsolutePath());

PrintWriter out = null;

try {

out = new PrintWriter(file);

} catch (FileNotFoundException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

PC[] myArray = {};

myArray = data.toArray(new PC[data.size()]);

for (int i = 0; i < myArray.length; i++) {

out.print(myArray[i].id.get() + ";" + myArray[i].CPU.get() + ";" + myArray[i].Fr.get() + ";"

+ myArray[i].RAM.get() + ";");

}

out.close();

}

});

hb.getChildren().addAll(addName, addCPU, addFr, addram, addButton, saveButton,downloadButton);

hb.setSpacing(5);

hb.setMaxWidth(700);

final VBox vbox = new VBox();

vbox.setSpacing(7);

vbox.setPadding(new Insets(10, 0, 0, 10));

vbox.getChildren().addAll(label, table, hb);

vbox.setMaxWidth(700);

chart.setTitle("Ноутбуки");

chart.setPadding(new Insets(10, 0, 0, 10));

chart.setMaxWidth(400);

UpdateChart();

TilePane tilePane = new TilePane();

tilePane.setStyle( "-fx-background-color: #f5c9df;");

tilePane.getChildren().addAll(vbox, chart);

((Group) scene.getRoot()).getChildren().add(tilePane);

stage.setScene(scene);

stage.show();

}

public static class PC {

private final SimpleStringProperty id;

private final SimpleStringProperty CPU;

private final SimpleStringProperty Fr;

private final SimpleStringProperty RAM;

private PC(String iName, String cName,String fName, String rName ) {

this.id = new SimpleStringProperty(iName);

this.CPU = new SimpleStringProperty(cName);

this.Fr = new SimpleStringProperty(fName);

this.RAM = new SimpleStringProperty(rName);

}

public String getID() {

return id.get();

}

public void setID(String iName) {

id.set(iName);

}

public String getCPU() {

return CPU.get();

}

public void setCPU(String iName) {

CPU.set(iName);

}

public String getFr() {

return Fr.get();

}

public void setFr(String iName) {

Fr.set(iName);

}

public String getRAM() {

return RAM.get();

}

public void setRam(String iName) {

RAM.set(iName);

}

}

}

* 1. Результаты выполнения

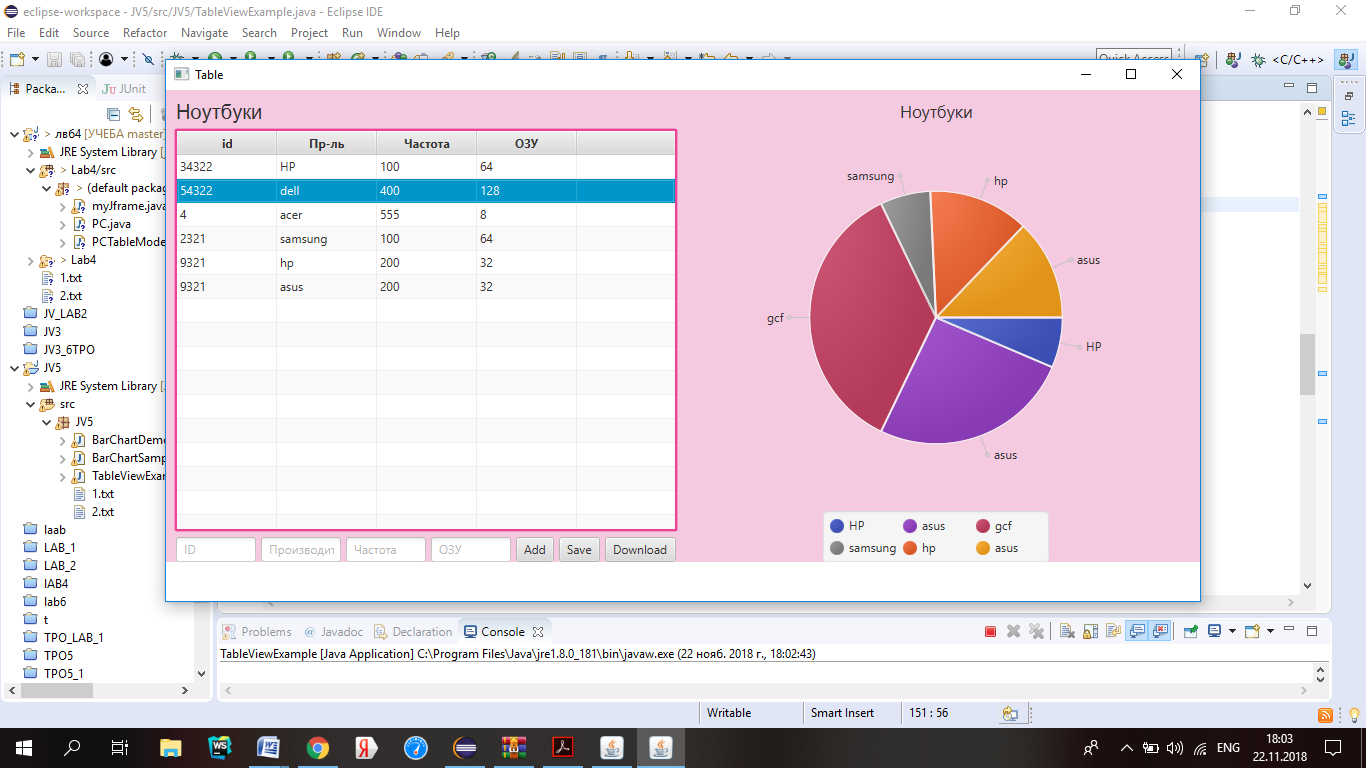


Рисунок 3.1 – Результаты работы программы

ВЫВОДЫ

В ходе четвертой лабораторной работы по дисциплине «Платформа Java» были изучены особенностями инструментария библиотеки платформы JAVAFX 2для создания графического интерфейса приложений на языке Java.

Были приобретены практические навыки создания Java-программ с графическим интерфейсом, позволяющим пользователю осуществлять взаимодействие с программой: задавать исходные данные, просматривать результаты работы программы в удобном виде.

Была написана программа в соответствии с вариантом задания.