МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Отчет по лабораторной работе №1

по дисциплине

Информационные технологии в системах управления

РУКОВОДИТЕЛЬ:

Миндров А.Е.

СТУДЕНТ:

Меженин М.Д.

17-АС

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2020

1. **Цель лабораторной работы**:

Изучить работу с классами в языке Java.

1. **Задание**:

Написать программу, проверяющую знания таблицы умножения.

1. **Листинг:**

**Lab1.java**

package sample;  
  
public class Lab1 {  
 private int x;  
 private int y;  
 private int c;  
  
 public Lab1() {  
 this.x = 1 + (int) (Math.*random*() \* 9);  
 this.y = 1 + (int) (Math.*random*() \* 9);  
 this.c = this.x \* this.y;  
 }  
  
 public int getX() {  
 return this.x;  
 }  
  
 public int getY() {  
 return this.y;  
 }  
  
 public int getC() {  
 return this.c;  
 }  
}

**Main.java**

package sample;  
  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
  
public class Main extends Application {  
 @Override  
 public void start(Stage primaryStage) throws Exception {  
 Parent root = FXMLLoader.*load*(getClass().getResource("MainPage.fxml"));  
 primaryStage.setTitle("Test");  
 primaryStage.setScene(new Scene(root, 700, 400));  
 primaryStage.show();  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

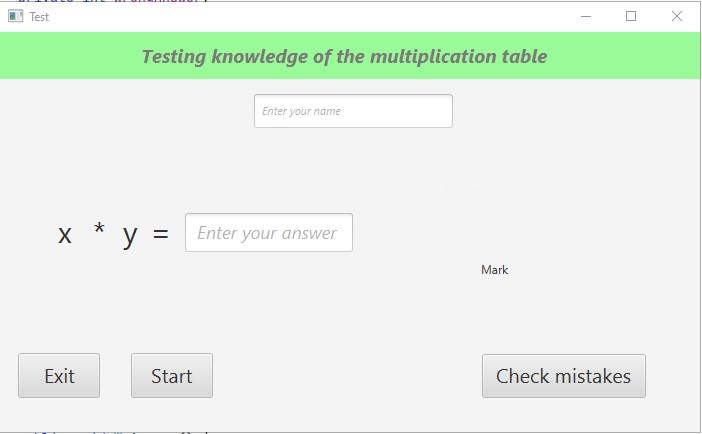
**MainPageController**

package sample;  
  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.\*;  
import javafx.stage.Stage;  
  
import java.util.ArrayList;  
  
public class MainPageController {  
 @FXML  
 public TextField name;  
  
 @FXML  
 public TextField answer;  
  
 @FXML  
 public Button start;  
  
 @FXML  
 public Label x;  
  
 @FXML  
 public Label y;  
  
 @FXML  
 public Label corranswer;  
  
 @FXML  
 public Label control;  
  
  
 User user = new User();  
 ArrayList<String> list = new ArrayList<>();  
 Stage stage;  
  
 @FXML  
 public void Start(ActionEvent event) {  
 Lab1 var = new Lab1();  
 x.setText(Integer.*toString*(var.getX()));  
 y.setText(Integer.*toString*(var.getY()));  
 corranswer.setText(Integer.*toString*(var.getC()));  
 start.setDisable(true);  
 user.setName(name.getText());  
 }  
 public void next(ActionEvent event) {  
 Alert input\_err = new Alert(Alert.AlertType.*INFORMATION*);  
 input\_err.setTitle("Error");  
 input\_err.setHeaderText(null);  
 input\_err.setContentText("Error!\nInvalid input!\n" + "Must be an integer.");  
 int Answer = 0;  
  
 try {  
 Answer = Integer.*parseInt*(answer.getText());  
 } catch (NumberFormatException e) {  
 input\_err.showAndWait();  
 return;  
 }  
  
 if (Answer == Integer.*parseInt*(corranswer.getText())) {  
 user.CorrAnswer();  
 } else {  
 user.WrAnswer();  
 list.add("x = " + x.getText() + " y = " + y.getText() + "\nYour answer: " + Answer + "\nCorrect answer: " + corranswer.getText());  
 }  
 user.setMark();  
 user.OutputResults();  
 control.setText(user.Result);  
 Lab1 var = new Lab1();  
 x.setText(Integer.*toString*(var.getX()));  
 y.setText(Integer.*toString*(var.getY()));  
 corranswer.setText(Integer.*toString*(var.getC()));  
 }  
  
 public void errors(ActionEvent event) {  
 String errors\_list = " ";  
  
 for (int i = 0; i < list.size(); i++) {  
 errors\_list += list.get(i) + "\n";  
 if (i == 4) {  
 break;  
 }  
 }  
  
 Alert errors\_alert = new Alert(Alert.AlertType.*INFORMATION*);  
 errors\_alert.setTitle("Your mistakes: ");  
 errors\_alert.setHeaderText(null);  
 errors\_alert.setContentText(errors\_list);  
 errors\_alert.showAndWait();  
 }  
  
 public void stop(ActionEvent event) {  
 stage = (Stage) ((Button) event.getSource()).getScene().getWindow();  
 stage.close();  
 }  
}

**User**

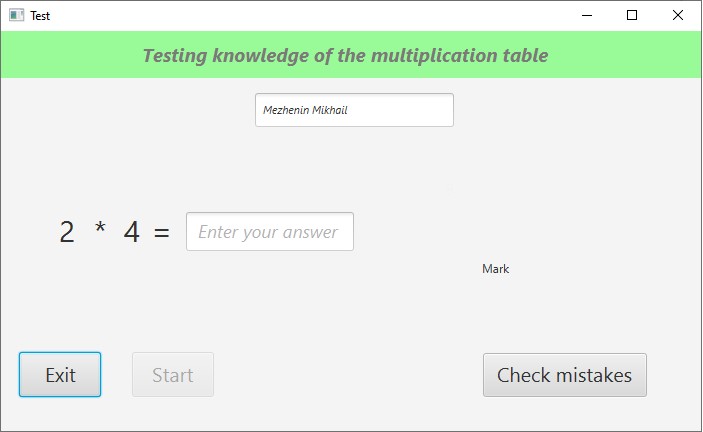
package sample;  
  
public class User {  
 private String name;  
 private int CorrectAnswer;  
 private int WrongAnswer;  
 private int Mark;  
 public String Result;  
 public int Error;  
  
 public User() {  
 this.CorrectAnswer = 0;  
 this.WrongAnswer = 0;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
  
 void CorrAnswer() {  
 int i = 0;  
 this.CorrectAnswer++;  
 i++;  
 }  
  
 public void WrAnswer() {  
 int j = 0;  
 this.WrongAnswer++;  
 j++;  
 }  
  
 public void setMark() {  
 int a = 51;  
 int b = 71;  
 int c = 81;  
 int percent = 100 \* this.CorrectAnswer / (this.CorrectAnswer + this.WrongAnswer);  
 this.Error = 1;  
 if (percent <= a) {  
 this.Mark = 2;  
 } else if (percent > a && percent <= b) {  
 this.Mark = 3;  
 } else if (percent > b && percent <= c) {  
 this.Mark = 4;  
 } else {  
 this.Mark = 5;  
 }  
 if (a >= b || b >= c || a >= c) {  
 this.Error = 0;  
 }  
 }  
  
 public void OutputResults() {  
 this.Result = ("Name: " + this.name  
 + "\nCorrect answers: " + this.CorrectAnswer  
 + "\nWrong answers: " + this.WrongAnswer  
 + "\nMark: " + this.Mark);  
 }  
}

1. **Скриншоты выполнения программы**



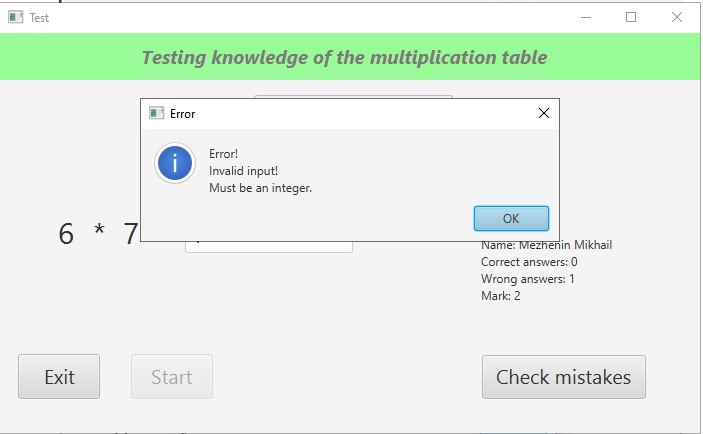
Это исходное окно приложения.

Далее вводим имя и нажимаем старт.



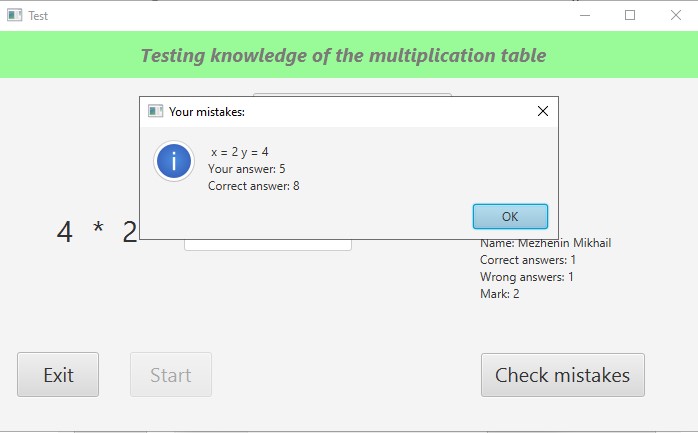
Видим, что вместо x и y сгенерировались цифры. Введём ответ и нажмём Enter.

Видим, что в правом нижнем углу появился контроль оценки с введенным именем.



Ответ, который не является числом воспринимается как неверный формат, но программа будет ждать, пока пользователь не введёт корректный ответ.

Далее нажмём кнопку “Показать ошибки”.



Видим, что ошибок у нас 7, но программа, как и планировалось, выводит только первые 5. Теперь остаётся только выйти из программы.