**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**федеральное государственное образовательное бюджетное учреждение высшего образования**

**«Вологодский государственный университет»**

**(ВоГУ)**

* 1. **ОБЪЕКТНО-ОРЕНТИРОВАННОЕ программирование**
  2. **лабораторнАЯ РАБОТА №5**

**«Язык программирования Java. Проектирование пользовательского интерфейса с использованием библиотеки Java FX»**

**Направление подготовки: 09.03.01 Информатика и вычислительная техника**

**Направленность (профиль): Программное обеспечение средств вычислительной техники и автоматизированных систем**

**Форма обучения: очная**

**Институт: Математики, естественных и компьютерных наук**

**Кафедра: Автоматики и вычислительной техники**

**Группа: 4Б09 ПО-41**

**Студент: Махонин М.Н.**

**Руководитель: Сорокин А.Н.**

Вологда

2021 г.

**Цель работы:** научиться создавать приложения на языке программирования Java, освоить базовые элементы графической библиотеки Java FX.

**Вариант 2**

Создайте диалоговое приложение «Тестирование», позволяющее проходить набор тестовых заданий. Варианты заданий хранятся в файле или в базе данных.

Класс *Questions*:

package sample;  
  
public class Questions {  
  
 private String question;  
 private String[] answers;  
  
 public Questions(){  
  
 }  
  
 public Questions(String question, String[] answers) {  
 this.question = question;  
 this.answers = answers;  
 }  
  
 public String correctAnswer() {  
 return this.answers[answers.length - 1];  
 }  
  
 public String getQuestion() {  
 return question;  
 }  
  
 public String[] getAnswers() {  
 return answers;  
 }

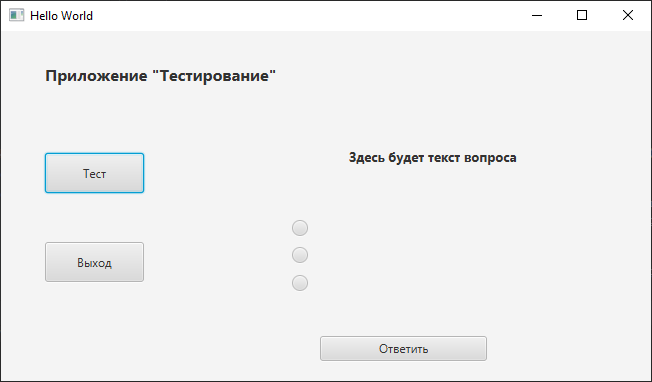
Класс *Controller:*

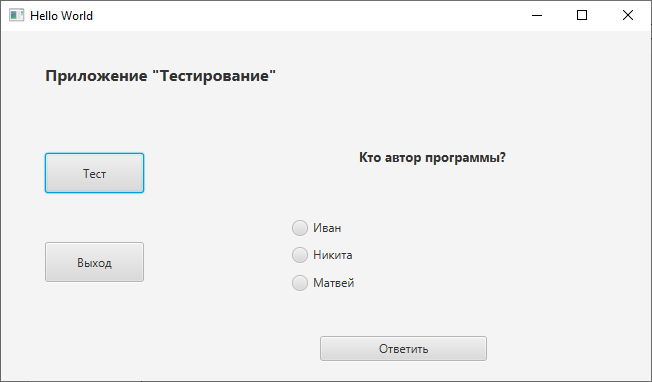
package sample;  
  
import com.fasterxml.jackson.core.JsonProcessingException;  
import com.fasterxml.jackson.databind.ObjectMapper;  
import javafx.event.ActionEvent;  
import javafx.fxml.FXML;  
import javafx.scene.control.\*;  
  
  
import javax.swing.\*;  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
  
  
public class Controller {  
  
 @FXML  
 private RadioButton var1;  
  
 @FXML  
 private RadioButton var2;  
  
 @FXML  
 private RadioButton var3;  
  
 @FXML  
 private ToggleGroup answers;  
  
 @FXML  
 private Label questionslbl;  
  
 private Questions[] questions;  
 private RadioButton[] l = new RadioButton[3];  
  
 private String nowCorrectAnswer;  
 private String result = "";  
 private String path;  
 private int nowQuestion = 0, correctAnswers;  
  
 //Выход из приложения  
 public void exit(ActionEvent actionEvent) {  
 System.*exit*(1);  
 }  
  
 //Нажатие на кнопку "Ответить"  
 public void Click(ActionEvent actionEvent) throws FileNotFoundException {  
 nowCorrectAnswer = questions[nowQuestion].correctAnswer();  
  
 RadioButton selected = (RadioButton) answers.getSelectedToggle();  
 if(selected != null){  
 String toggleGroupeValue = selected.getText();  
 if(toggleGroupeValue.equals(nowCorrectAnswer)){  
 correctAnswers++;  
 result += "№" + nowQuestion + " Ответ верный\n";  
 }  
 else {  
 result += "№" + nowQuestion + " Ответ неверный, правильный ответ " + nowCorrectAnswer +"\n";  
 }  
  
 if(nowQuestion + 1 == questions.length){  
 for(int i = 0; i < 3; i++){  
 l[i].setVisible(false);  
 questionslbl.setText(result);  
 }  
 }else{  
 nowQuestion++;  
 nowCorrectAnswer = questions[nowQuestion].correctAnswer();  
 questionslbl.setText(questions[nowQuestion].getQuestion());  
 String[] answers = questions[nowQuestion].getAnswers();  
 for(int i = 0; i < 3; i++){  
 l[i].setText(answers[i]);  
 }  
 selected.setSelected(false);  
 }  
 }  
 }  
  
 //Загрузка теста из JSON файла  
 public void LoadTest(ActionEvent actionEvent) {  
 l[0] = var1;  
 l[1] = var2;  
 l[2] = var3;  
 JFileChooser chooser = new JFileChooser();  
 int returnVal = chooser.showOpenDialog(null);  
 if(returnVal == JFileChooser.*APPROVE\_OPTION*){  
 ObjectMapper mapper = new ObjectMapper();  
 try {  
 path = chooser.getSelectedFile().getAbsolutePath();  
 questions = mapper.readValue(new File(chooser.getSelectedFile().getPath()), Questions[].class);  
 questionslbl.setText(questions[0].getQuestion());  
 for(int i = 0; i < 3; i++) {  
 l[i].setSelected(false);  
 l[i].setVisible(true);  
 l[i].setText(questions[0].getAnswers()[i]);  
 }  
 } catch (JsonProcessingException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

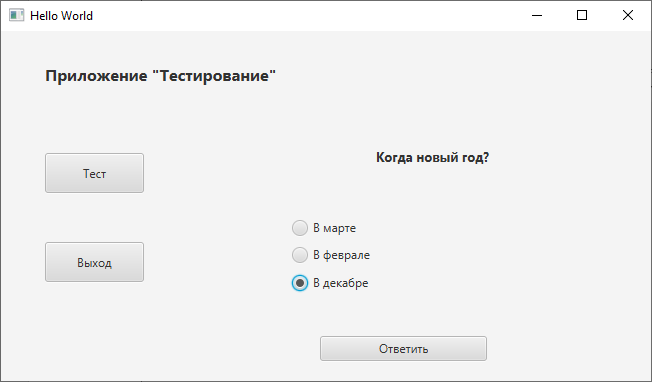
Класс *main:*

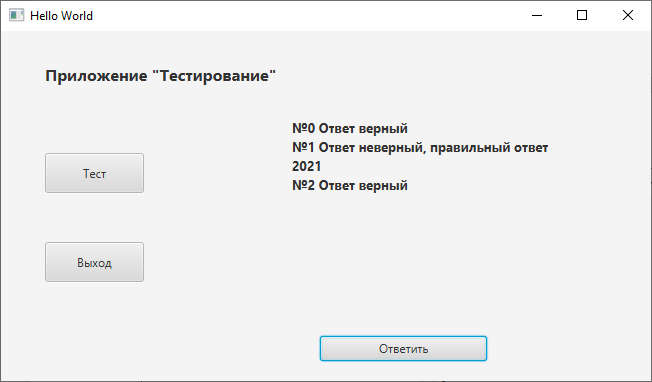
package sample;  
  
import com.fasterxml.jackson.databind.ObjectMapper;  
import javafx.application.Application;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
import javafx.stage.Stage;  
import java.io.File;  
  
  
public class Main extends Application {  
  
 static String *path* = "c:\\Users\\acer\\Documents\\";  
 @Override  
 public void start(Stage primaryStage) throws Exception{  
 Parent root = FXMLLoader.*load*(getClass().getResource("sample.fxml"));  
 primaryStage.setTitle("Hello World");  
 primaryStage.setScene(new Scene(root, 650, 350));  
 primaryStage.show();  
 Questions[] q = new Questions[3];  
 q[0] = new Questions("Кто автор программы?", new String[]{"Иван", "Никита", "Матвей"});  
 q[1] = new Questions("Какой сейчас год?", new String[]{"1990", "0", "2021"});  
 q[2] = new Questions("Когда новый год?", new String[]{"В марте", "В феврале", "В декабре"});  
 ObjectMapper mapper = new ObjectMapper();  
 mapper.writeValue(new File(*path* + "result.json"), q);  
 }  
  
 public static void main(String[] args) {  
 *launch*(args);  
 }  
}

Тестирование программы:









**Контрольные вопросы**

1. **Опишите основные возможности библиотеки JavaFX.**

JavaFX представляет инструментарий для создания кроссплатформенных графических приложений на платформе Java.

JavaFX позволяет создавать приложения с богатой насыщенной графикой благодаря использованию аппаратного ускорения графики и возможностей GPU.

С помощью JavaFX можно создавать программы для различных операционных систем: Windows, MacOS, Linux, Android, iOS и для самых различных устройств: десктопы, смартфоны, планшеты, встроенные устройства, ТВ. Приложение на JavaFX будет работать везде, где установлена исполняемая среда Java (JRE).

JavaFX предоставляет большие возможности по сравнению с рядом других подобных платформ, в частности, по сравнению со Swing. Это и большой набор элементов управления, и возможности по работе с мультимедиа, двухмерной и трехмерной графикой, декларативный способ описания интерфейса с помощью языка разметки FXML, возможность стилизации интерфейса с помощью CSS, интеграция со Swing

1. **Опишите основные элементы языка разметки FXML.**

В FXML элемент XML представляет одно из следующего:

* Экземпляр class
* Свойство экземпляра class
* "Статическое" свойство
* "Определить" блок

Блок кода сценария

Экземпляры класса, свойства экземпляра, статические свойства, и определяют блоки, обсуждаются в этом разделе ниже. Сценарии обсуждаются в более позднем разделе.

*Экземпляр класса*

Экземпляры класса могут быть созданы в FXML несколькими способами. Наиболее распространенное через элементы объявления экземпляра, которые просто создают новый экземпляр class по имени. Другие способы создать экземпляры class включают ссылающиеся существующие значения, копируя существующие значения, и включая внешние файлы FXML.

*Свойство экземпляра класса:*

Элементы, имена тега которых начинаются со строчной буквы, представляют объектные свойства. Элемент свойства может представить одно из следующего:

* Метод set свойства
* Свойство списка только для чтения
* Свойство карты только для чтения

*«Статические» Свойства:*

Элемент может также представить "статическое" свойство (иногда названный "присоединенным свойством"). Статические свойства являются свойствами, которые только имеют смысл в определенном контексте. Они не являются внутренними class, к которому они применяются, но определяются другим class (часто, родительский контейнер управления).

*Блок define:*

<fx:define> элемент используется, чтобы создать объекты, которые существуют за пределами иерархии объектов, но, возможно, должны быть упомянуты в другом месте.