**Лабораторная работа №5**

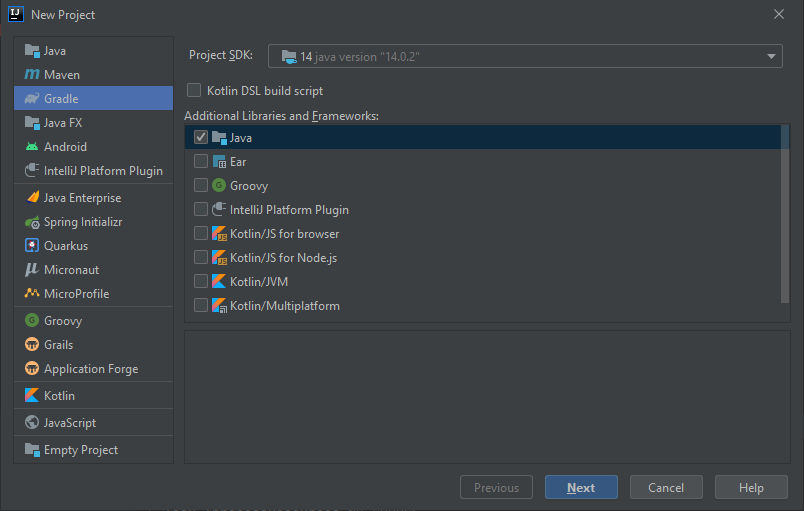
*Лабораторная работа №5 выдаётся студентам в одном варианте. Срок выполнения – 2 недели. Форма представления отчёта – программа, написанная на языке Java (код и скрины примера работы программы).*

**Тема:** разработка графического интерфейса на языке Java с использованием библиотеки JavaFX.

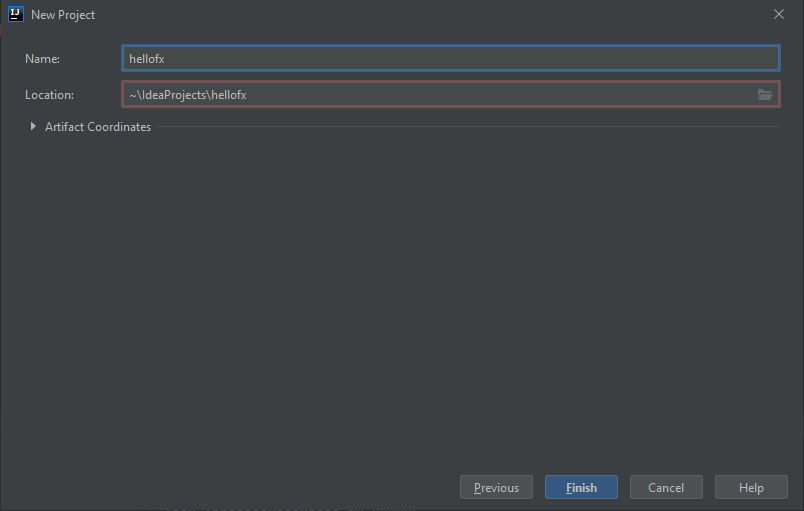
**Задание №5:**

1. Откройте *IntellijIDEA*

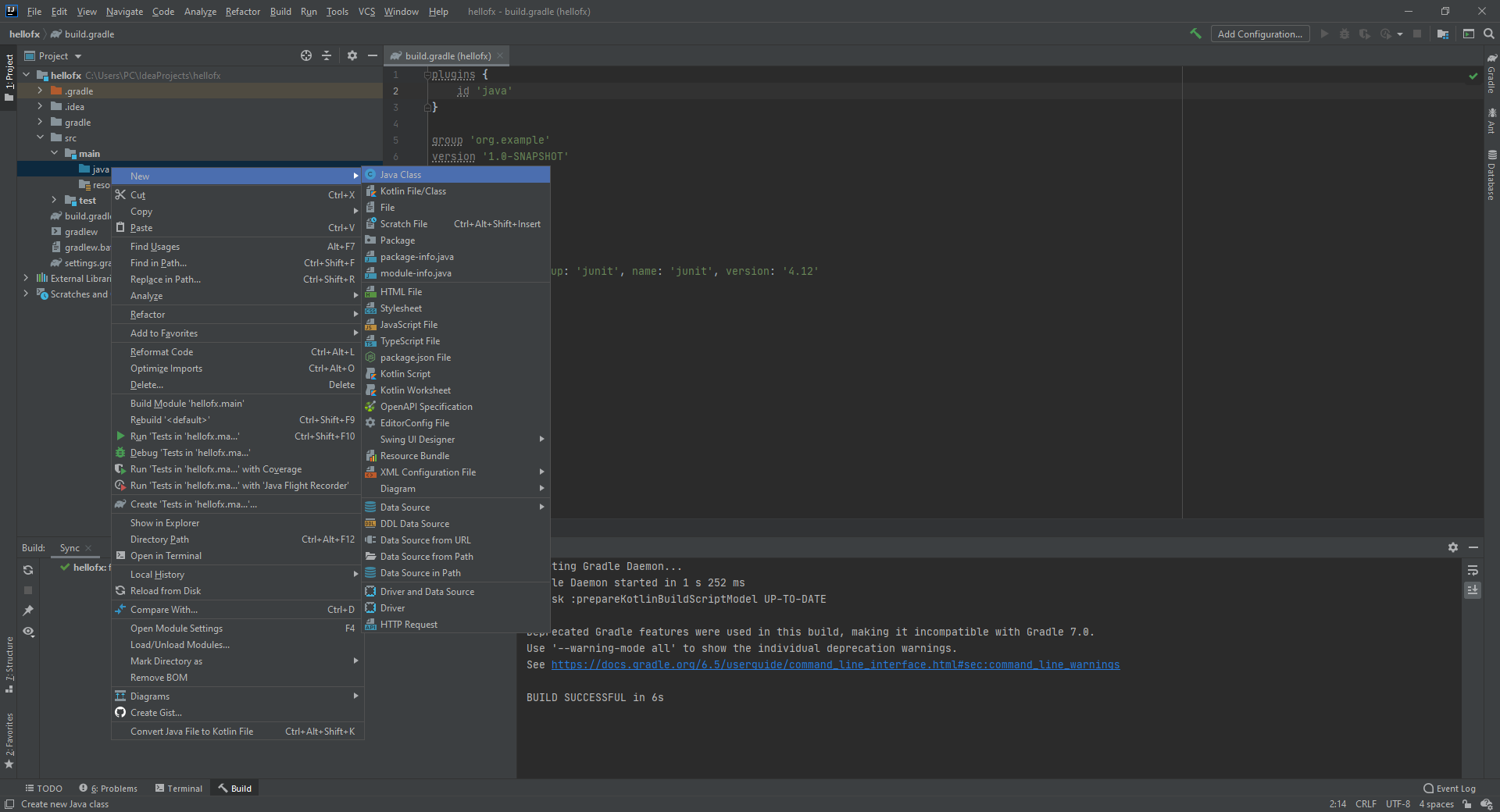
2. Создайте новый *Gradle* проект, нажмите далее:



3. В следующем окне укажите название проекта и выберите его расположения, например:



4. В папке *src/main/java* создайте новый класс, назовите его *MainApp*:



5. Откройте файл *build.gradle* и замените его содержимое на следующее:

plugins {

id 'application'

id 'org.openjfx.javafxplugin' version '0.0.8'

}

repositories {

mavenCentral()

}

dependencies {

}

javafx {

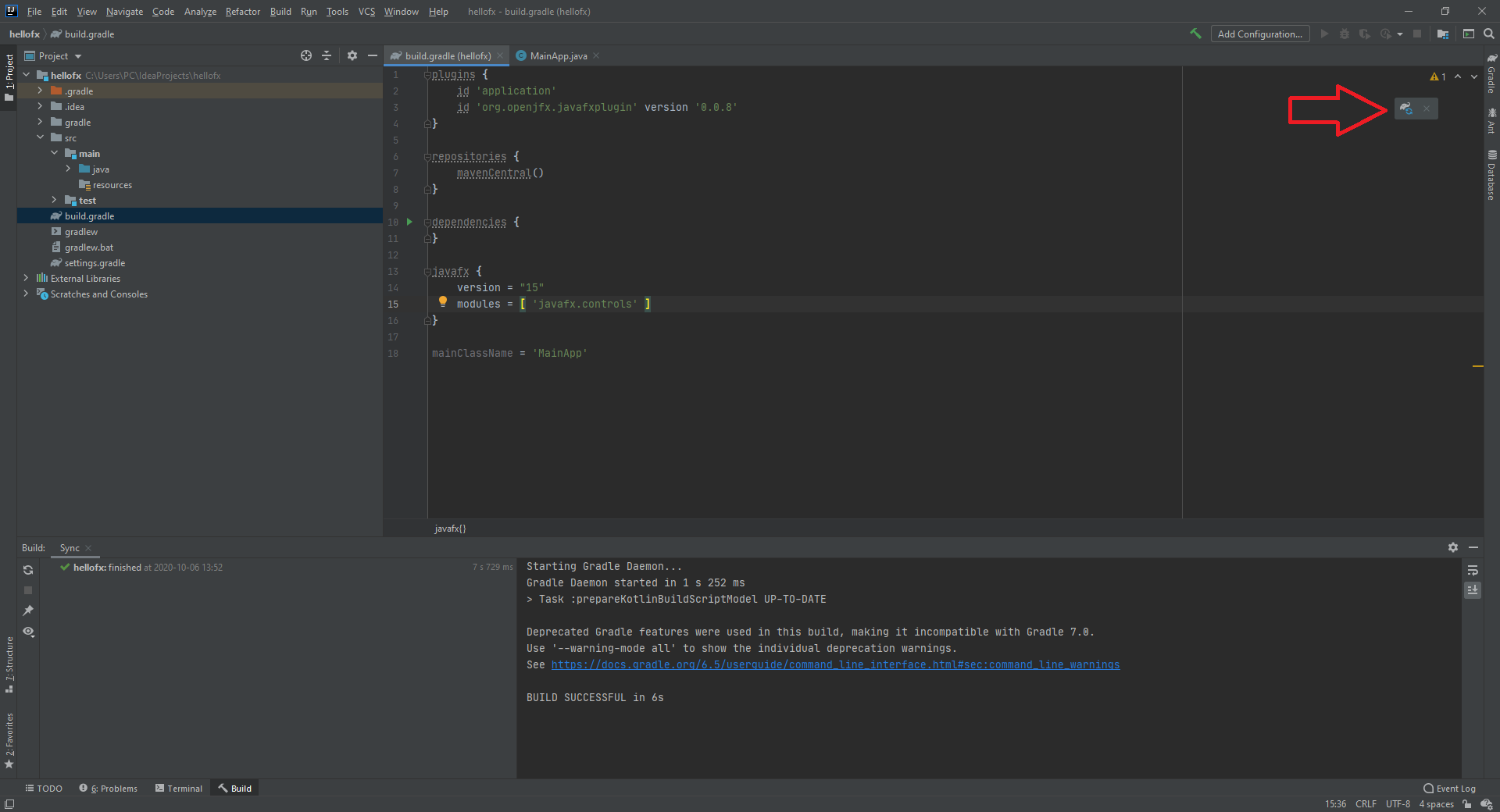
version = "15"

modules = [ 'javafx.controls' ]

}

mainClassName = 'MainApp'

6. Послечегообновитеконфигурацию*Gradle*:

****

7. Вернитесь к файлу *MainApp.java* и замените его содержимое на:

import javafx.application.Application;

import javafx.stage.Stage;

import javafx.scene.Scene;

import javafx.scene.control.Button;

import javafx.scene.layout.BorderPane;

public class MainApp extends Application {

public static void main(String[] args) {

Application.launch(args);

}

@Override

public void start(Stage stage) {

Button btn = new Button("Click");

BorderPane root = new BorderPane(btn);

Scene scene = new Scene(root, 800, 600);

stage.setScene(scene);

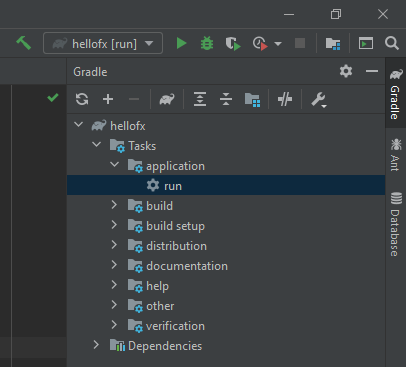
stage.setTitle("HelloFX");

stage.show();

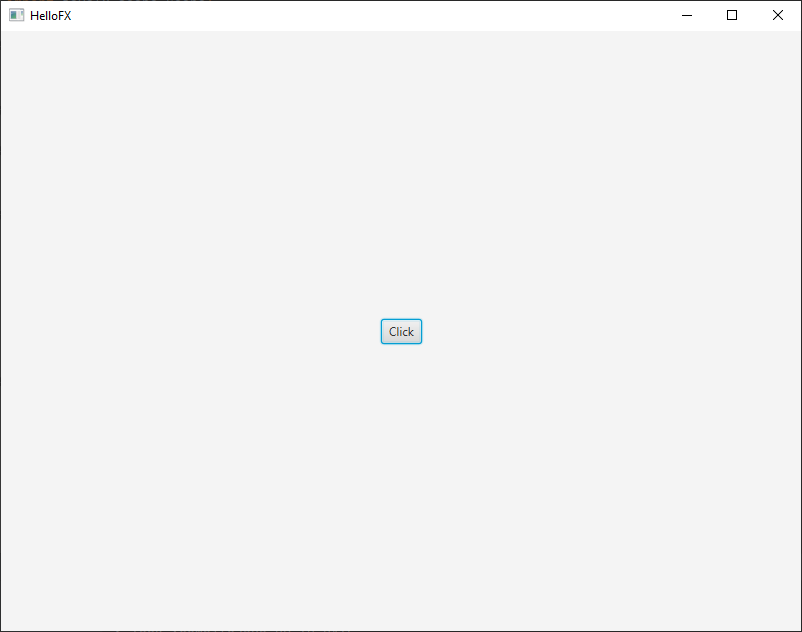
}

}

8. Откройте вкладку *Gradle*, она находится справа. Найдите пункт *application/run* и кликните по нему дважды.



9. После этого должно появиться окно нашего приложения:

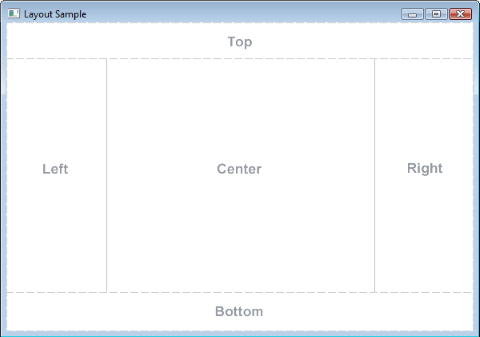


10. Разберём код класса *MainApp*.

Выполнение начинается с метода *main*, он запускает внутренние механизмы *JavaFX*, после чего выполняется метод *start*, в нём и происходит построение интерфейса нашего приложения.

В первой строке создаётся кнопка с надписью *Click*. Пока она не выполняет никаких действий.

В следующей строке мы создаём контейнер *BorderPane*. Он позволяет размещать элементы интерфейса по следующей схеме:



Мы передаём кнопку в конструктор *BorderPane*, она автоматически помещается в центр.

В 3 строке создаётся окно приложения. Мы указываем содержимое окна и задаём его размеры.

Затем мы задаём название нашего окна и вызываем метод *show*, чтобы отобразить окно.

11. Измените код, создав контейнер *VBox* и поместив в него несколько элементов:

Label label1 = new Label("Label1");

Label label2 = new Label("Label2");

Label label3 = new Label("Label3");

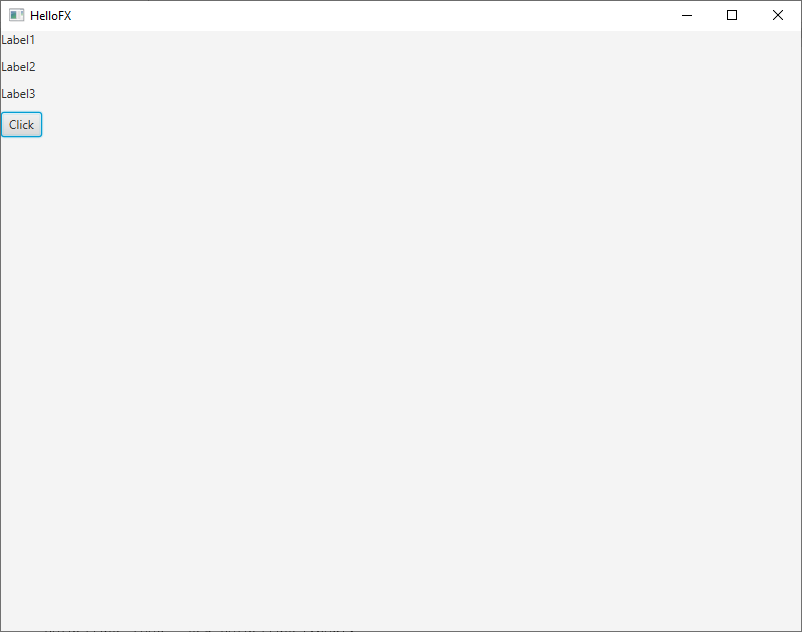
Button btn = new Button("Click");

VBoxvbox = new VBox(10, label1, label2, label3, btn);

BorderPane root = new BorderPane(vbox);

Контейнер *VBox*позволяет расположить элементы в столбик. Первый параметр задаёт расстояние между элементами.

12. Запустите приложение. Вы должны увидеть следующее:



Контейнер *VBox*занимает весь экран, поэтому он не центруется. Существует аналогичный контейнер HBoxдля горизонтального расположения элементов.

13. Создадим Todo-приложение. Поместите в метод *start*следующий код:

VBoxtodos = new VBox();

ScrollPanescrollPane = new ScrollPane(todos);

TextFieldtextField = new TextField();

Button addButton = new Button("+");

HBox controls = new HBox(8, textField, addButton);

controls.setPadding(new Insets(8));

BorderPane root = new BorderPane();

root.setCenter(scrollPane);

root.setBottom(controls);

Scene scene = new Scene(root, 550, 800);

stage.setScene(scene);

stage.setTitle("HelloFX");

stage.show();

Здесь мы создаём контейнер *todos*, в котором будут лежать задачи;*textField*–поле, для ввода текста новой задачи; *addButton*–кнопка для добавления новой задачи.

14. Добавьте действие при нажатии на кнопку addButton:

addButton.setOnAction((event) -> {

Label text = new Label(textField.getText());

Button removeButton = new Button("x");

HBoxtodo = new HBox(8, text, removeButton);

todo.setPadding(new Insets(8));

todos.getChildren().add(todo);

textField.setText("");

});

При нажатии на кнопку формируется новый блок задачи, он добавляется в список задач, а поле ввода - очищается.

15. Самостоятельно реализуйте удаление задач из списка при нажатии на *removeButton*.