

Оглавление

1 ОПИСАНИЕ ИСПОЛЬЗОВАННЫХ ПРОГРАММНЫХ ТЕХНОЛОГИЙ И СРЕДСТВ.....	2
2 ОПИСАНИЕ ПРОВЕДЕННЫХ ЭКСПЕРИМЕНТОВ ПО РАЗРАБОТКЕ И ИСПОЛЬЗОВАНИЮ И ПРОГРАММНЫХ СРЕДСТВ	2
2.1 Измерение времени, затраченного на разработку программных продуктов.....	2
2.2 Измерение количества файлов, необходимых для функционирования программных продуктов.....	3
2.3 Измерение количества строк кода, написанных при разработке программных продуктов.....	3
2.4 Описание визуальных различий элементов графический интерфейсов приложений	3
2.5 Описание времени отклика элементов графический интерфейсов приложений	4
2.6 Выборочное измерение времени отклика элементов интерфейсов на основании результатов п. 5.....	5
3 ПРЕДСТАВЛЕНИЕ ПОЛУЧЕННЫХ ВЫВОДОВ В НАГЛЯДНОЙ ФОРМЕ	6
4 АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ.....	14
Список использованных источников	16

1 ОПИСАНИЕ ИСПОЛЬЗОВАННЫХ ПРОГРАММНЫХ ТЕХНОЛОГИЙ И СРЕДСТВ

С целью разработки на языке Java использовалась кросс-платформенная среда разработки IntelliJ IDEA Ultimate (версия 2023.1) производства компании JetBrains [4]. При этом использовался OpenJDK (Java Development Kit) версии 22.0.2 [5].

Для разработки текстового редактора «Блокнот» в варианте «Swing» использовались библиотеки `javax.swing` и `java.awt`, которые по умолчанию входят в используемый JDK [2].

Для разработки текстового редактора «Блокнот» в варианте «JavaFX» использовалась библиотека `javafx`, которая не входит в используемый JDK и требует дополнительной установки. В данном случае использовалась JavaFX SDK версии 22.0.1 [1, 3].

2 ОПИСАНИЕ ПРОВЕДЕННЫХ ЭКСПЕРИМЕНТОВ ПО РАЗРАБОТКЕ И ИСПОЛЬЗОВАНИЮ И ПРОГРАММНЫХ СРЕДСТВ

Для сравнительной оценки GUI-фреймворков Swing [2] и JavaFX [1, 3], используемых при разработке текстового редактора «Блокнот» на языке Java, в соответствии с планом исследований были использованы эксперименты как на этапе разработки программных продуктов, так и на этапе их тестирования.

2.1 Измерение времени, затраченного на разработку программных продуктов

Для разработки текстового редактора «Блокнот» в варианте «Swing» было потрачено 7 дней.

Для разработки текстового редактора «Блокнот» в варианте «JavaFX» было потрачено 8 дней.

Более точное измерение времени, затраченного на разработку программных продуктов, не проводилось по причине сложности фиксации.

Сложность фиксации связана с использованием в разработке не применявшихся ранее программных технологий и средств.

2.2 Измерение количества файлов, необходимых для функционирования программных продуктов

Для разработки текстового редактора «Блокнот» в варианте «Swing» были использованы 14 файлов, содержащих программный код (рис. 1, табл. 1).

Для разработки текстового редактора «Блокнот» в варианте «JavaFX» были использованы 9 файлов, содержащих программный код (рис. 2, табл. 2).

При этом в обоих случаях не учитываются одни и те же графические файлы для иконок окон и кнопок, а также системные файлы программной среды, необходимые для сборки проектов.

2.3 Измерение количества строк кода, написанных при разработке программных продуктов

Для разработки текстового редактора «Блокнот» в варианте «Swing» были использованы 808 строк кода (табл. 1).

Для разработки текстового редактора «Блокнот» в варианте «JavaFX» были использованы 733 строки кода (табл. 2).

При этом необходимо отметить, что с целью удобства восприятия и сопровождения кода для отделения смысловых блоков кода в разработке могут использоваться пустые строки. В данной работе пустые строки не учитывались, подсчет выполнялся только в отношении действительных строк, имеющих собственно код.

2.4 Описание визуальных различий элементов графический интерфейсов приложений

О обоих случаях при реализации графического интерфейса использовался вариант отображения приложения в стилистике нативного отображения окон операционной системы (оба фреймворка предоставляют такие возможности).

При этом есть ряд отличий, обусловленных реализацией тех или иных элементов интерфейса непосредственно самими фреймворками (рис. 3, 4). К таким особенностям можно отнести следующие:

- 1) различное расстояние между пунктами меню «Файл», «Поиск», «Вид», «О программе» (меньшее расстояние в варианте «Swing», большее – в варианте «JavaFX»);
- 2) различное качество отображения иконок кнопок (худшее, более «пиксельное» в варианте «Swing»; лучшее, более сглаженное – в варианте «JavaFX»), которое рельефно проявляется при увеличении элементов кнопок (рис. 5, 6); при этом ухудшение качества не связано с разрешением начального файла – фреймворк Swing сжимает («ухудшает») изображение даже при начальном формате изображений для иконок 512 x 512 пикселей;
- 3) различное качество отображения иконок стандартных информационных окон (рис. 7, 8) – худшее, более «пиксельное» в варианте «Swing»; лучшее, более сглаженное – в варианте «JavaFX»;
- 4) различное визуальное представление панели вкладок и способа выделения активной вкладки (рис. 3, 4); в варианте «Swing» заголовок вкладки более компактный с выделением активной вкладки с помощью обрамления заголовка прямоугольником из пунктирной линии; в варианте «JavaFX» заголовок вкладки более широкий, выделение активной вкладки обрамлением трех сторон ярлыка активной вкладки синим цветом;
- 5) различное визуальное представление соответствующих окон «Открытие» и «Сохранение» (рис. 9, 10, 11, 12), касающееся в первую очередь цветового оформления (только светлой темы в варианте «Swing» и текущей темы операционной системы в варианте «JavaFX»).

2.5 Описание времени отклика элементов графический интерфейсов приложений

Субъективная оценка времени открытия самого приложения при старте, а также отклика элементов интерфейса при взаимодействии пользователя с приложением с помощью мышки и клавиатуры показала практически

одинаковое время запуска и отклика обоих вариантов приложения, кроме времени отклика между нажатием кнопок и пунктов меню «Открыть» и «Сохранить» с одной стороны и открытия соответствующих окон «Открытие» и «Сохранение» – с другой.

В варианте «Swing» отклик указанных двух элементов интерфейса существенно более длительный, чем в варианте «JavaFX».

Для объективизации полученных предварительных данных необходимо выполнение измерений по п. 6.

2.6 Выборочное измерение времени отклика элементов интерфейсов на основании результатов п. 5.

Для объективизации времени отклика между воздействием пользователем на той или иной элемент интерфейса с одной стороны и желаемым действием с другой, необходима программная реализация следующих строчек кода.

Перед выполнением запрашиваемого действия добавлялся блок кода

```
long startTime = System.currentTimeMillis(),
```

а после выполнения запрашиваемого действия блок кода

```
long endTime = System.currentTimeMillis();
```

```
System.out.println("Окно открыто через " + (endTime -  
startTime) + " мс").
```

Указанные строки кода были добавлены для оценки времени отклика функций открытия и сохранения файлов. Результаты измерений (рис. 13, 14) свидетельствуют о существенной задержке при исполнении кода в варианте «Swing» (от 726 мс до 240 мс) по сравнению с вариантом «JavaFX» (от 2 мс до 0 мс).

3 ПРЕДСТАВЛЕНИЕ ПОЛУЧЕННЫХ ВЫВОДОВ В НАГЛЯДНОЙ ФОРМЕ

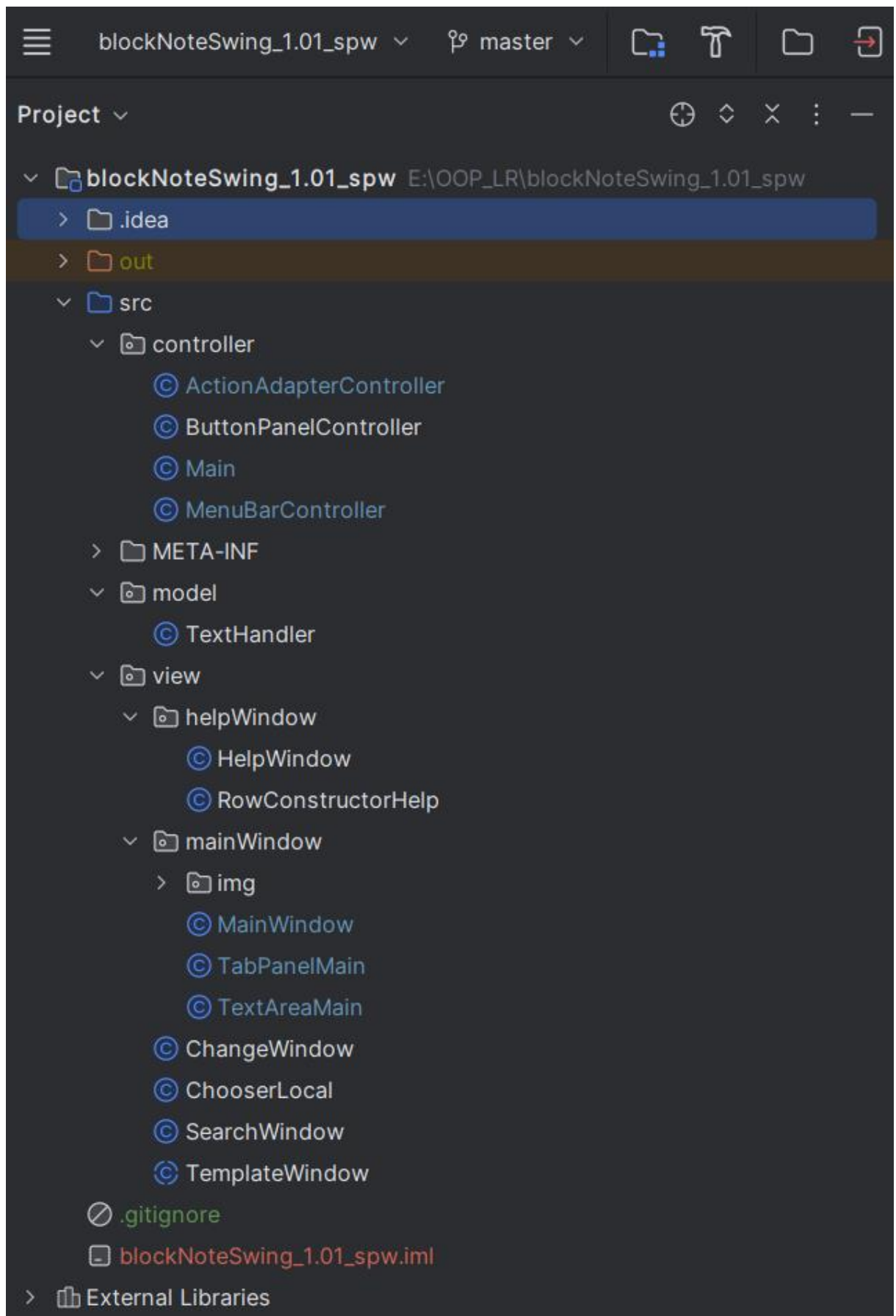


Рисунок 1 – Структура проекта экран варианта «Swing»

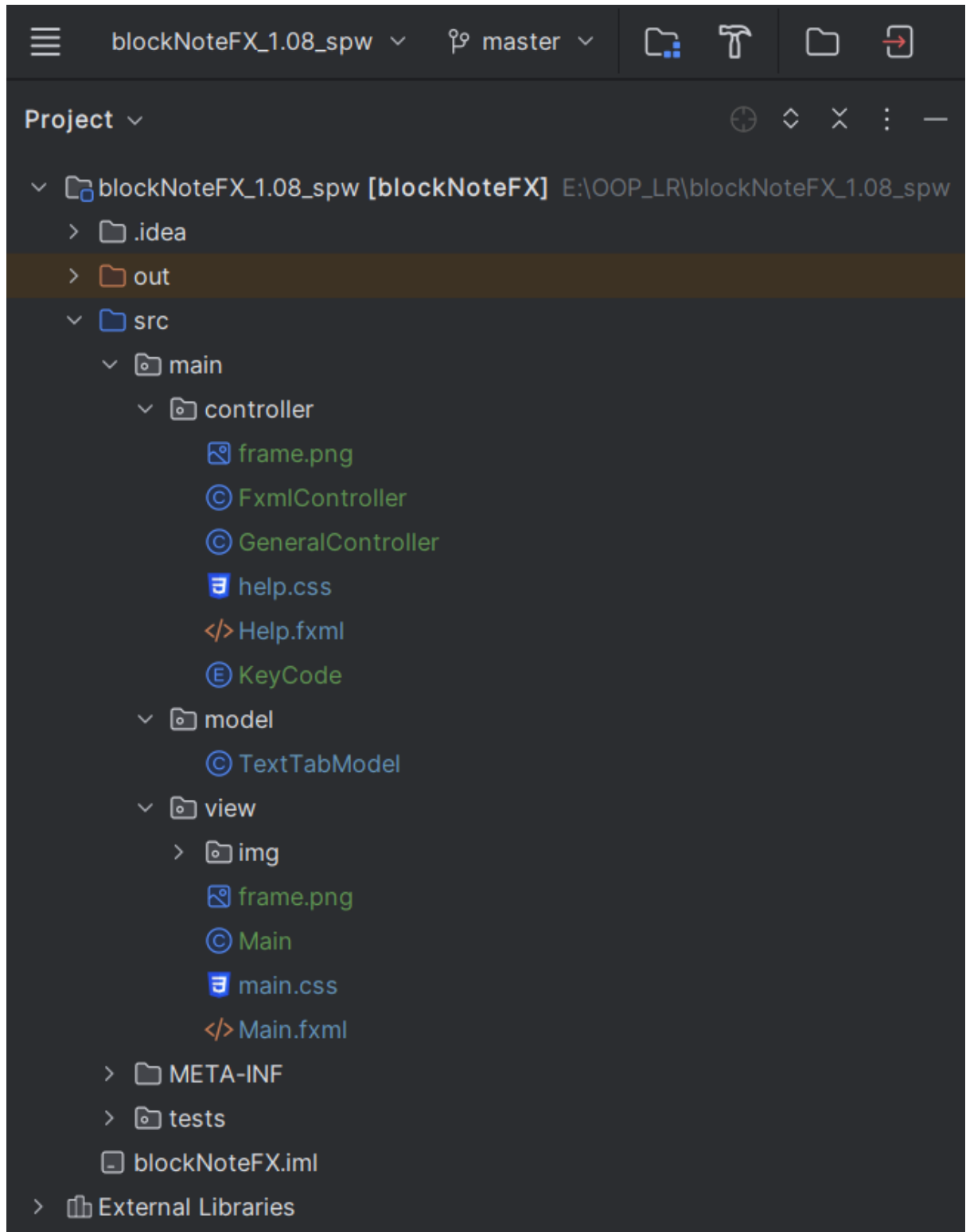


Рисунок 2 – Структура проекта экран варианта «JavaFX»

Таблица 1 – Количество строк кода приложения в варианте «Swing»

	Наименование файла	Количество строк
1	ActionAdapterController.java	156
2	ButtonPanelController.java	80
3	Main.java	16
4	MenuBarController.java	79
5	TextHandler.java	44
6	HelpWindow.java	156
7	RowConstructorHelp.java	25
8	MainWindow.java	17
9	TabPanelMain.java	17
10	TextAreaMain.java	28
11	ChangeWindow.java	76
12	ChooserLocal.java	36
13	SearchWindow.java	65
14	TemplateWindow.java	13
	ИТОГО:	808

Таблица 2 – Количество строк кода приложения в варианте «JavaFX»

	Наименование файла	Количество строк
1	FxmlController.java	51
2	GeneralController.java	240
3	help.css	6
4	Help.fxml	215
5	KeyCode.java	23
6	TextTabModel.java	51
7	Main.java	74
8	main.css	28
9	Main.fxml	45
	ИТОГО:	733

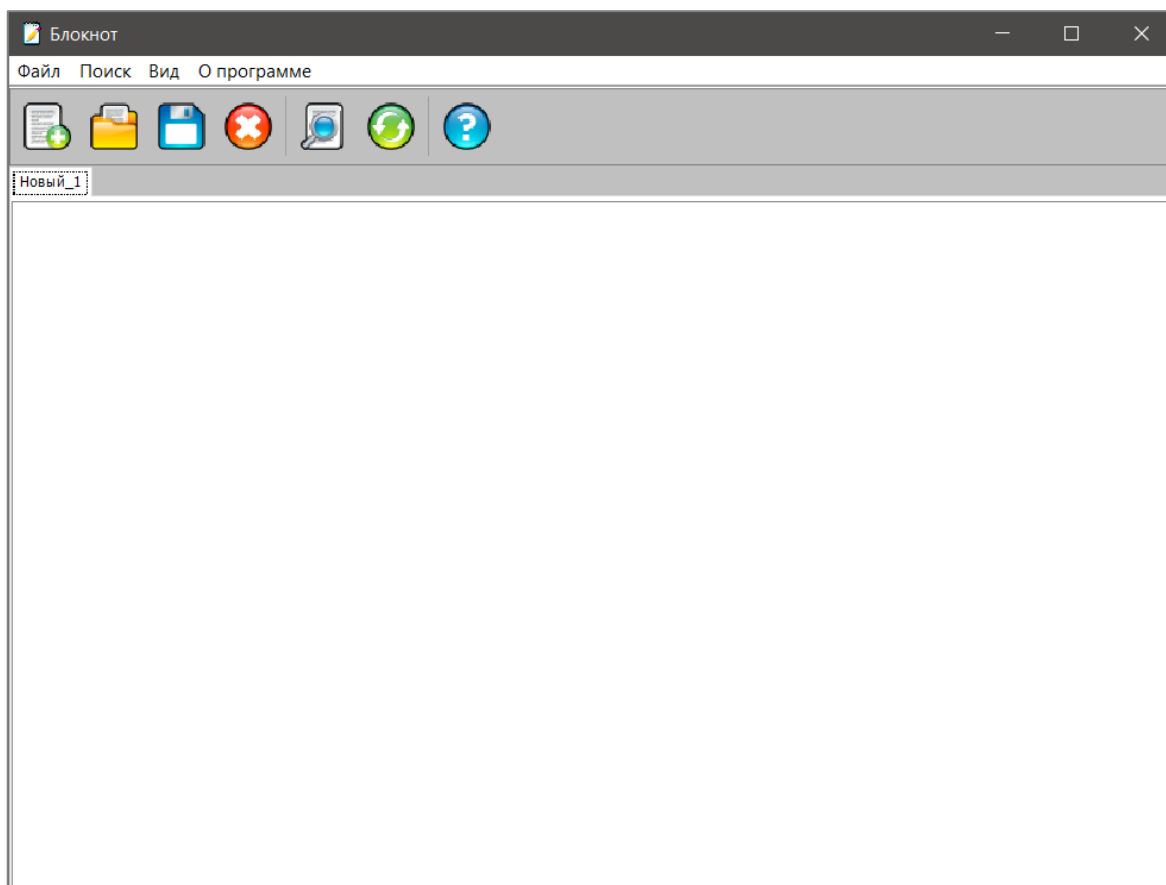


Рисунок 3 – Основное окно варианта «Swing»

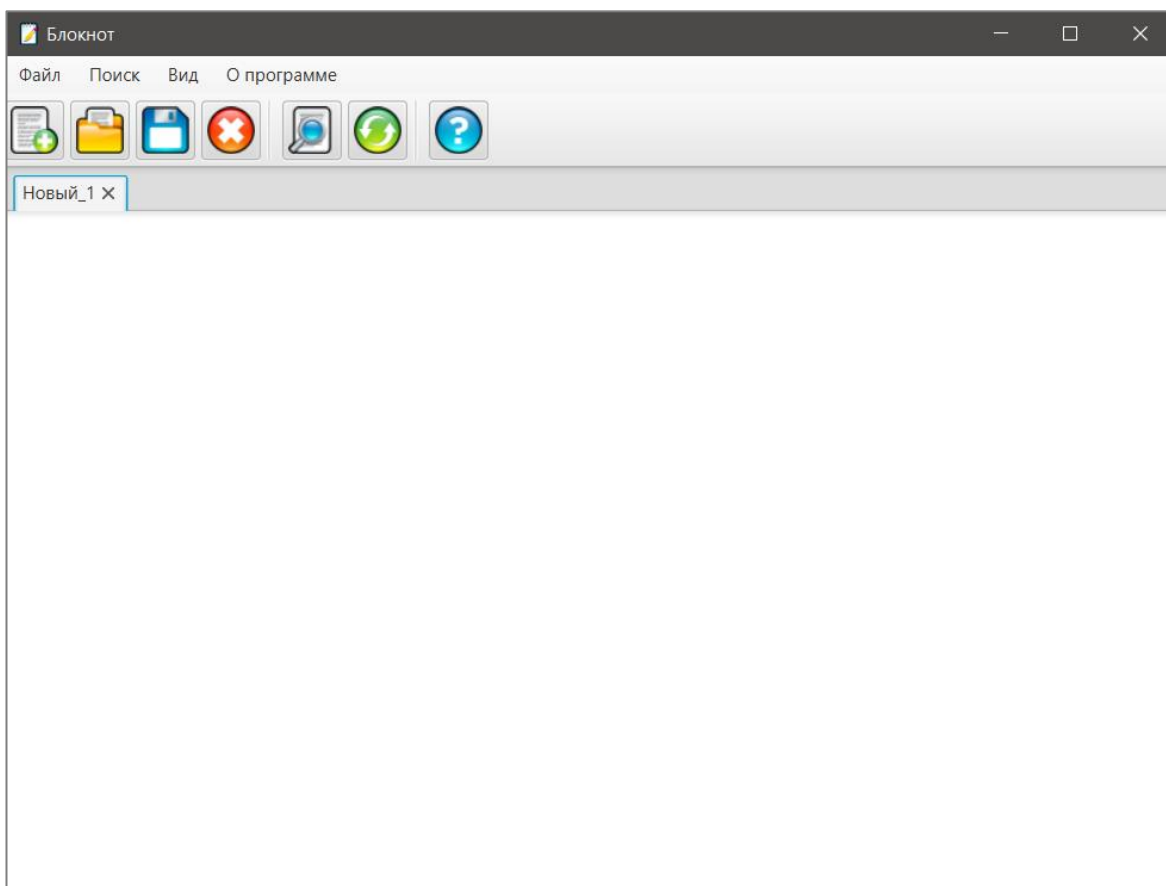


Рисунок 4 – Основное окно варианта «JavaFX»

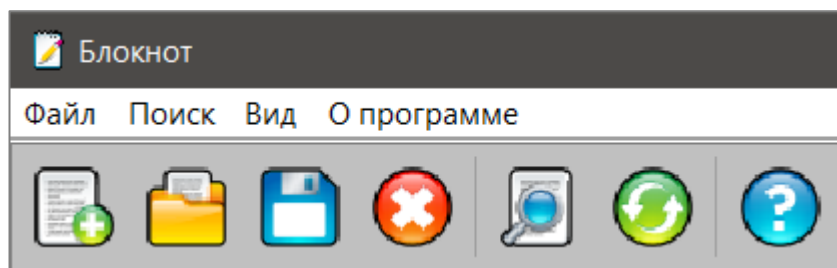


Рисунок 5 – Панель управления варианта «Swing»

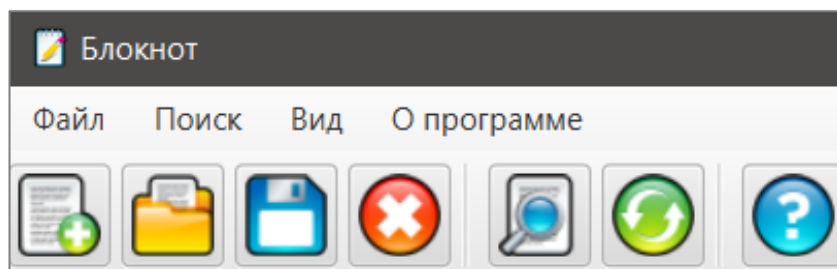


Рисунок 6 – Панель управления варианта «JavaFX»

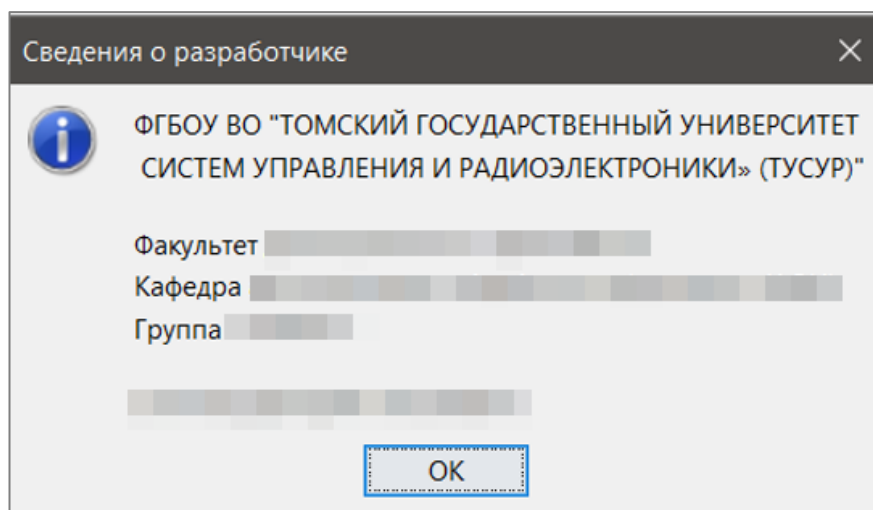


Рисунок 7 – Окно сведений о разработчике приложения варианта «Swing»

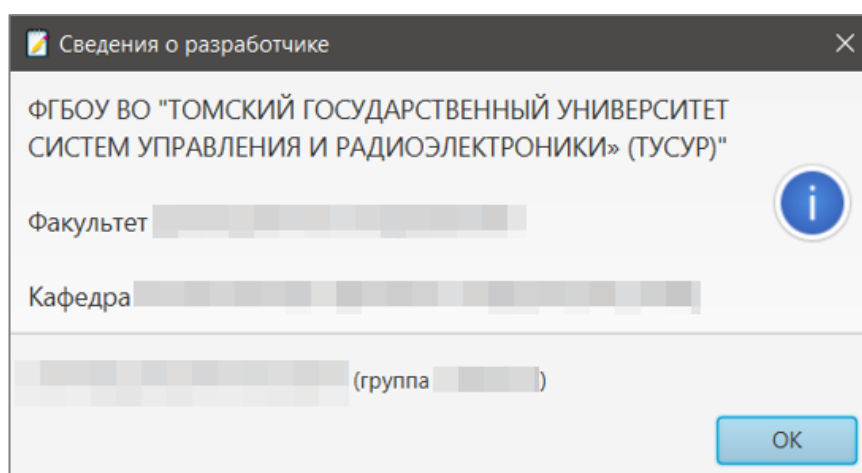


Рисунок 8 – Окно сведений о разработчике приложения варианта «JavaFX»

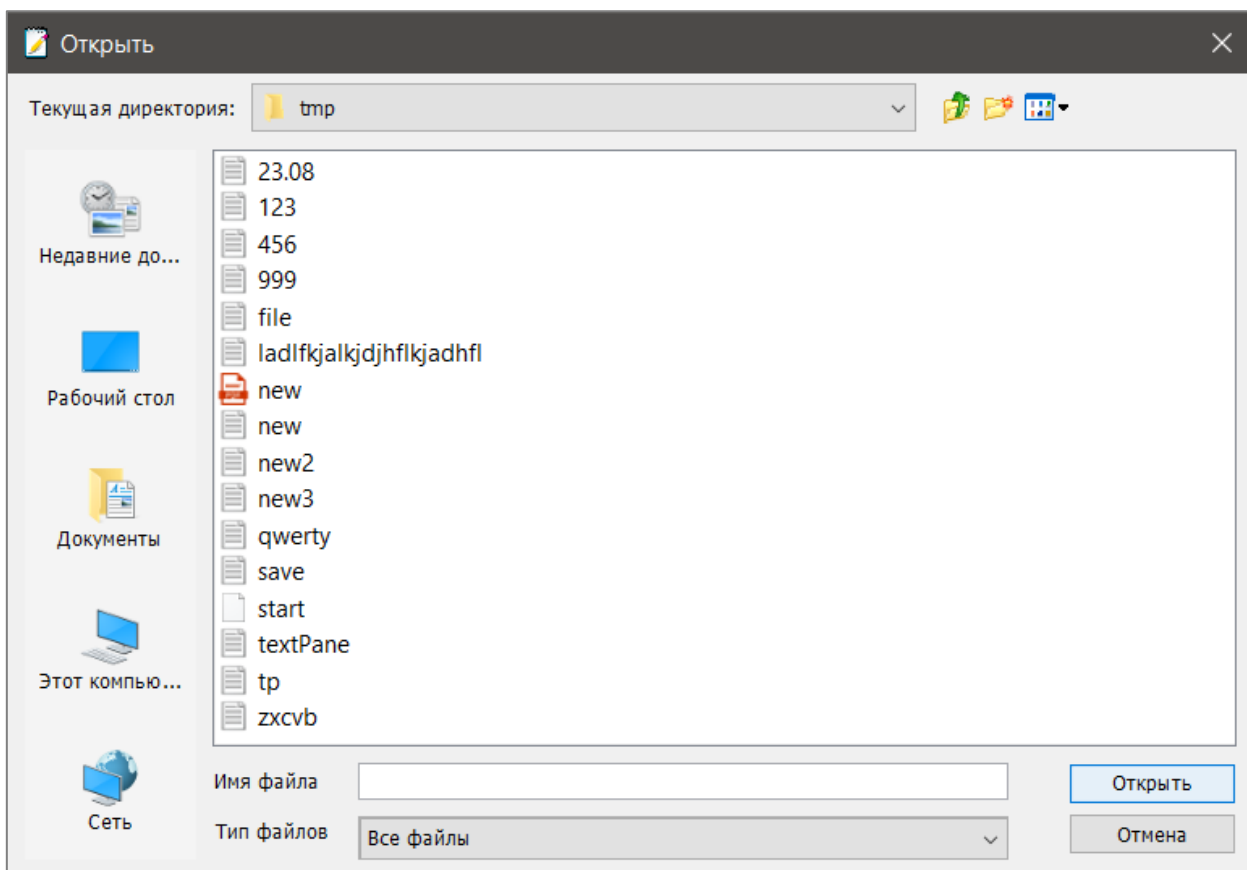


Рисунок 9 – Окно открытия файла варианта «Swing»

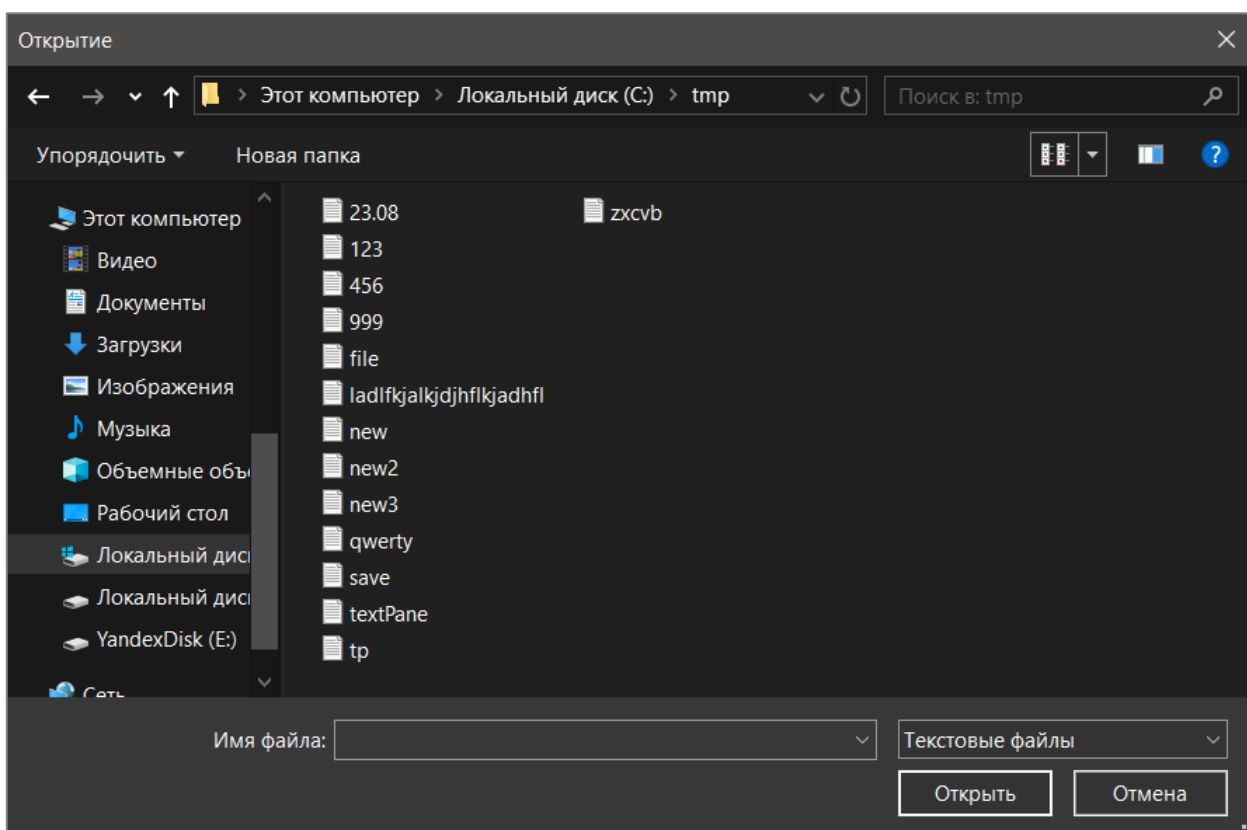


Рисунок 10 – Окно открытия файла варианта «JavaFX»

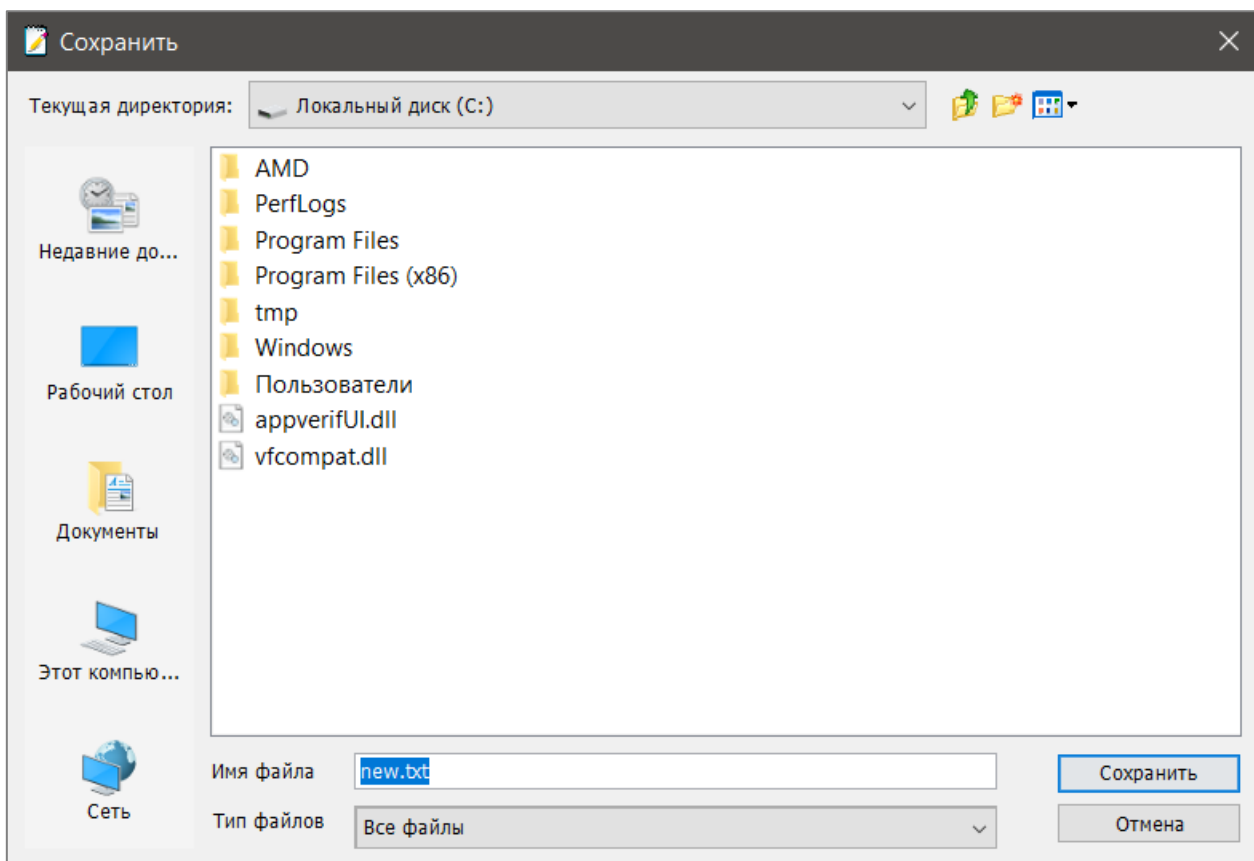


Рисунок 11 – Окно сохранения файла варианта «Swing»

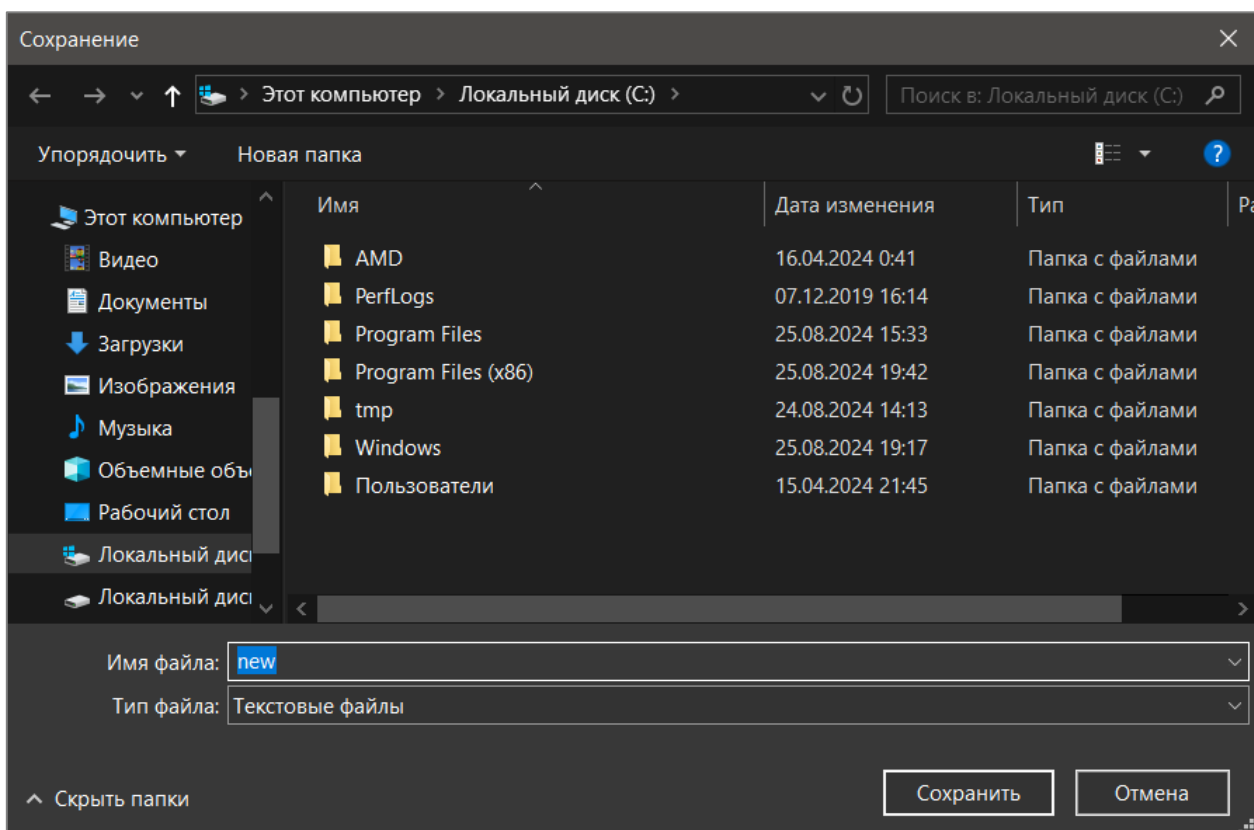


Рисунок 12 – Окно сохранения файла варианта «JavaFX»

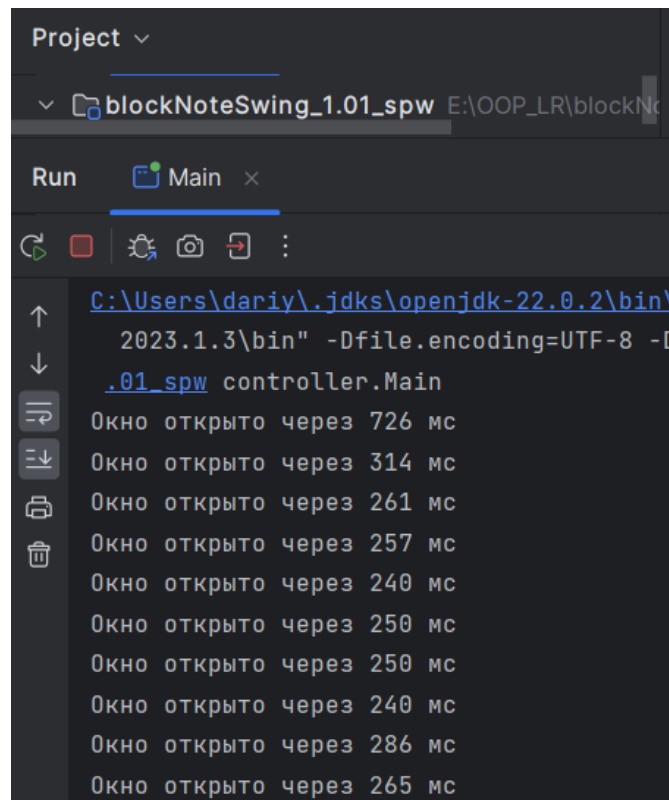


Рисунок 13 – Скорость отклика окон открытия и сохранения файла варианта «Swing»

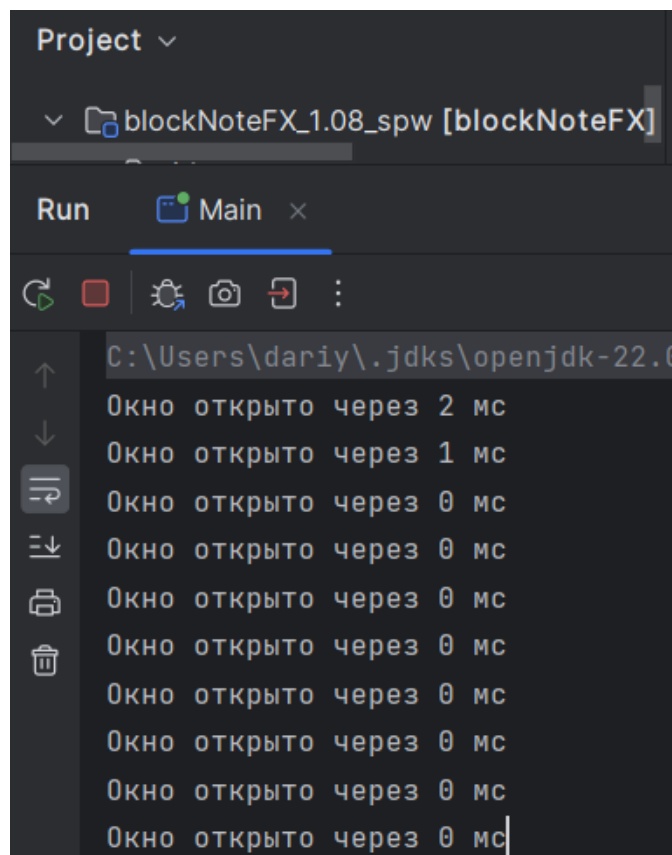


Рисунок 14 – Скорость отклика окон открытия и сохранения файла варианта «JavaFX»

4 АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ ЭКСПЕРИМЕНТАЛЬНЫХ ИССЛЕДОВАНИЙ

После проведения запланированных экспериментов по разработке и использованию и программных средств был проведен анализ полученных результатов.

1. Время, затраченное на разработку обоих вариантов программных продуктов, сопоставимо: 7 дней для варианта «Swing» и 8 дней для варианта «JavaFX». При этом необходимо отметить сложность фиксации времени, затраченного на разработку, связанного с тем фактом, что затрачивалось время не только на собственно написание программного кода, но и на поиск и изучение кодовой базы и новой области знаний.

2. Для разработки в варианте «JavaFX» использовано меньшее количество файлов, чем в варианте «Swing» (9 файлов против 14). В основном данное различие обусловлено возможностью файлов *.fxml (вариант «JavaFX») содержать лаконичную структуру визуально отображаемых окон программного продукта, в то время как для подобного отображения в варианте «Swing» приходится создавать не только дополнительные строки кода, но и дополнительные классы и файлы.

3. Для разработки в варианте «JavaFX» использовано меньшее количество строк кода, чем в варианте «Swing» (733 файлов против 808), что объясняется не только наличием типов файлов *.fxml и *.css в варианте «JavaFX», но и особенностями реализации обоих фреймворков собственно на языке Java.

4. Визуально полученные с использованием применяемых фреймворков приложения достаточно похожи. В ходе более детального изучения отдельных элементов графического интерфейса наблюдаются достаточно существенные различия. При этом необходимо отметить, что если пп. 1, 4 и 5 (расстояние между пунктами меню, визуальное представление панели вкладок, способа выделения активной вкладки, визуальное представление соответствующих окон «Открытие» и «Сохранение») можно

считать хоть и морально устаревшими, но стилистическими особенностями интерфейса Swing, то пп. 2 и 3 (различное качество отображения иконок кнопок и стандартных информационных окон) однозначно являются недостатками фреймворка Swing, ухудшающими визуальное восприятие программного продукта пользователем.

5. Время открытия самого приложения при старте, а также отклика элементов интерфейса при взаимодействии пользователя с приложением с помощью мышки и клавиатуры субъективно сопоставимы (кроме открытия окон «Открытие» и «Сохранение»).

6. При исполнении кода в варианте «Swing» наблюдается существенная задержка (от 726 мс до 240 мс) по сравнению с вариантом «JavaFX» (от 2 мс до 0 мс) при открытии окон «Сохранение» и «Открытие».

Исходя из полученных результатов, можно подтвердить данные других авторов [1, 3] о том, что по сравнению с фреймворком Swing фреймворк Java является более современным. В проведенных исследованиях это подтверждается как субъективными данными (время разработки, визуальное восприятие), так и объективными данными (количество файлов, количество строк кода, время отклика элементов интерфейса).

Список использованных источников

1. Машнин Т.С. JavaFX 2.0: разработка RIA-приложений. СПб., БХВ-Петербург, 2012. 320 с.
2. Портянкин И. Swing: Эффектные пользовательские интерфейсы. Москва, Лори, 2011. 590 с.
3. Прохоренок Н.И. JavaFX. СПб., БХВ-Петербург, 2020. 768 с.
4. IntelliJ IDEA Ultimate. Ведущая IDE для разработки на Java и Kotlin [Электронный ресурс]: официальный сайт компании JetBrains. URL: <https://www.jetbrains.com/ru-ru/idea/download/?section=windows> (дата обращения: 15.09.2024).
5. OpenJDK JDK 22.0.2 General-Availability Release [Электронный ресурс]: официальный сайт производственных сборок OpenJDK и сборок с ранним доступом от Oracle. URL: <https://jdk.java.net/22/> (дата обращения: 15.09.2024).