



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_\_\_\_\_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_

КАФЕДРА \_\_\_\_\_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_\_\_09.03.01 Информатика и вычислительная техника\_\_\_\_\_

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## *К КУРСОВОЙ РАБОТЕ*

*НА ТЕМУ:*  
**«ПРИЛОЖЕНИЕ ДЛЯ РЕДАКТИРОВАНИЯ  
ФОТОГРАФИЙ “ФОТОМАСТЕР”»**

Студент ИУ6-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Р.В. Дорохов  
(И.О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Т.И. Булдакова  
(И.О. Фамилия)

2024 г.

## **РЕФЕРАТ**

Расчетно-пояснительная записка состоит из 42 страниц, включающих в себя 15 рисунков, 7 таблиц, 9 источников и 3 приложения.

**ПРИЛОЖЕНИЕ, ИЗОБРАЖЕНИЕ, ПАРАМЕТР, ОБРАБОТКА, ПАКЕТНАЯ ОБРАБОТКА, RGB**

Объектом разработки является приложение для редактирования изображений.

Цель работы – проектирование и реализация приложения, позволяющего выполнять точечное редактирование параметров фотографий (таких как баланс белого, тон и присутствие) и проводить пакетную обработку изображений.

В результате разработки было спроектировано и реализовано приложение для редактирования фотографий, а также проведено тестирование программного продукта с использованием методов функционального, ручного и оценочного тестирования.

Разрабатываемое приложение предназначено для широкого круга пользователей, включая фотографов, дизайнеров и любителей редактирования фотографий.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	4
1 Анализ требований и уточнение спецификаций .....	5
1.1 Анализ предметной области .....	5
1.2 Анализ существующих приложений для редактирования фотографий. .	7
1.3 Анализ задания и выбор технологии и языка разработки .....	10
2 Проектирование структуры и компонентов программного продукта .....	12
2.1 Проектирование интерфейса пользователя .....	12
2.1.1 Разработка диаграммы вариантов использования .....	12
2.1.2 Разработка диаграммы состояний интерфейса .....	15
2.1.3 Разработка форм интерфейса .....	16
2.2 Разработка диаграммы пакетов .....	17
2.3 Разработка диаграмм классов приложения .....	18
2.3.1 Разработка диаграмм классов интерфейсной части .....	18
2.3.2 Разработка диаграмм классов функциональной части .....	20
2.4 Демонстрация работы функций приложения .....	26
3 Выбор стратегии тестирования и разработка тестов .....	29
3.1 Тестирование ручным контролем .....	29
3.2 Функциональное тестирование .....	35
3.2.1 Метод эквивалентных разбиений .....	36
3.2.2 Метод анализа граничных значений .....	39
3.3 Оценочное тестирование .....	42
ЗАКЛЮЧЕНИЕ .....	43
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	44
ПРИЛОЖЕНИЕ А .....	45
ПРИЛОЖЕНИЕ Б .....	54
ПРИЛОЖЕНИЕ В .....	63

## **ВВЕДЕНИЕ**

Фотография – это искусство и средство коммуникации, доступное каждому. Она сочетает в себе творчество и технологии, позволяя запечатлевать и преобразовывать моменты жизни. Процесс редактирования изображений представляет собой ручной подбор параметров изображения. Однако ручное редактирование может быть утомительным и подверженным ошибкам, особенно если необходимо обрабатывать сразу несколько изображений.

Современные приложения для редактирования фотографий значительно упрощают эту задачу, однако многие из них имеют свои ограничения. Некоторые программы являются платными или требуют подписки, другие рассчитаны на мобильные устройства и имеют урезанный функционал. Также существует множество профессиональных редакторов с перегруженным интерфейсом, который может отпугнуть начинающих пользователей.

Именно поэтому было решено разработать собственное приложение для редактирования фотографий, которое будет обладать необходимым функционалом для корректировки основных параметров (таких как баланс белого, тон и присутствие) и поддерживать пакетную обработку, при этом оставаясь удобным и понятным в использовании.

Основная цель – предоставить пользователю инструмент для простого и эффективного редактирования изображений.

Для решения поставленной задачи необходимо

- проанализировать предметную область;
- исследовать существующие аналоги приложений для редактирования изображений;
- определить инструменты и технологии для разработки;
- спроектировать структуру и интерфейс приложения;
- реализовать приложение на основе выбранных инструментов;
- отладить и протестировать приложение.

# **1 Анализ требований и уточнение спецификаций**

## **1.1 Анализ предметной области**

Перед началом проектирования приложения необходимо провести анализ предметной области, в рамках которой оно будет функционировать. Требования и ожидаемый результат пользователя формализованы в техническом задании, из которого вытекает предметная область.

Предметной областью задачи является обработка и редактирование цифровых изображений. Это включает в себя операции по корректировке таких параметров, как экспозиция, температура, оттенок, а также возможность пакетной обработки фотографий.

Исходя из вышеизложенного, можно выделить такие сущности предметной области, как:

- изображение (пакет изображений);
- параметры обработки;
- предпросмотр результата;
- температура;
- оттенок;
- экспозиция;
- контрастность;
- светлые области;
- тени;
- белые;
- черные;
- зернистость;
- четкость;
- размытие;
- красочность;
- насыщенность;
- пользователь.

Связи между этими сущностями можно выразить так:

- пользователь загружает изображение или пакет изображений;
- пользователь настраивает параметры редактирования;
- параметры применяются к активному изображению;
- параметры применяются ко всем изображениям в пакете;
- предпросмотр отображает изменения в реальном времени;
- измененное изображение сохраняется в файл;
- пакет измененных изображений сохраняется в файл.

Примем за данные: изображение, пакет изображений, параметры.

Конечным результатом анализа предметной области является концептуальная модель предметной области, изображенная на рисунке 1.

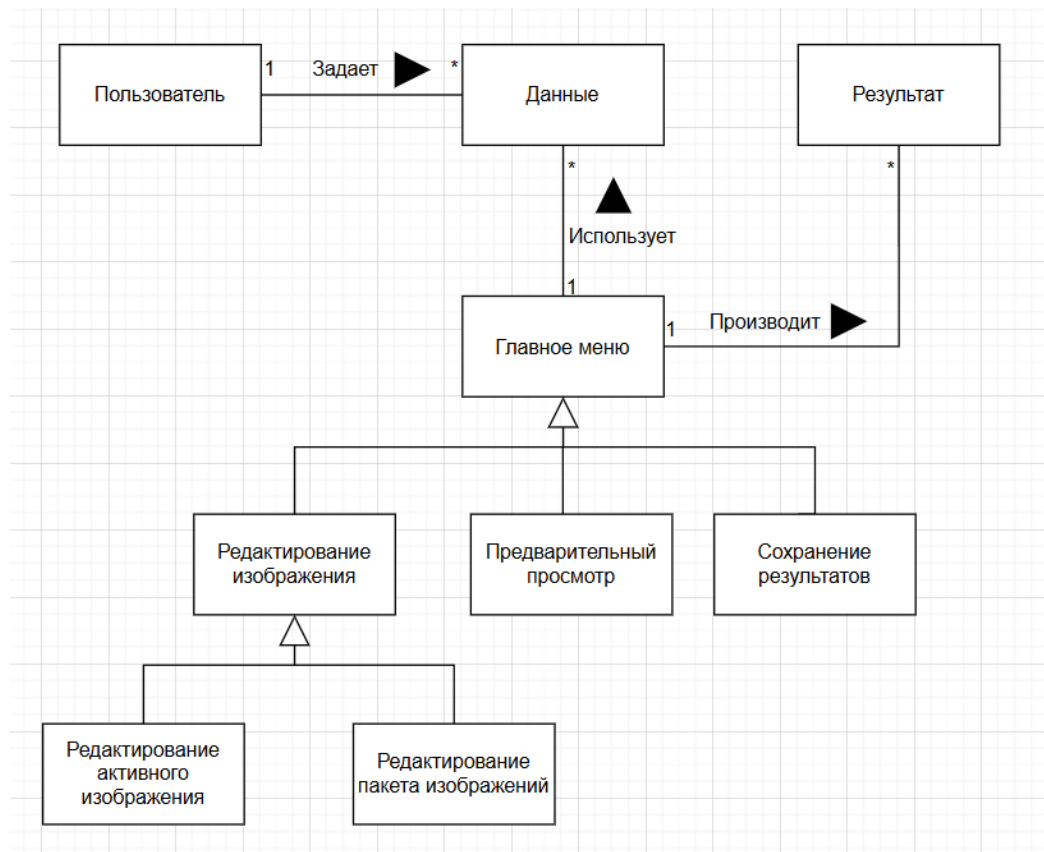


Рисунок 1 - Концептуальная модель предметной области

Далее проведем анализ существующих приложений для редактирования фотографий.

## **1.2 Анализ существующих приложений для редактирования фотографий.**

Для формирования более чёткого представления о проектируемом приложении был проведён сравнительный анализ, заключающийся в рассмотрении существующих приложений-аналогов для редактирования изображений. Такой анализ показал, что одними из самых популярных являются:

- 1) “Adobe Lightroom”: Мощное приложение для профессионального редактирования фотографий, включающее инструменты для точечной коррекции, работы с цветами и пакетной обработки изображений;
- 2) “GIMP”: Бесплатный кроссплатформенный редактор с широким функционалом;
- 3) “Snapseed”: Мобильное приложение для редактирования фотографий, разработанное Google, предоставляющее интуитивный интерфейс и мощные инструменты для точной корректировки;
- 4) “Pxlr Editor”: Облачный графический редактор, предлагающий базовые функции редактирования изображений через веб-браузер.

Аналоги было решено оценивать по следующим критериям:

- точечная коррекция параметров изображений;
- возможность пакетной обработки;
- интуитивно понятный, простой и неперегруженный интерфейс;
- необходимость подключения к интернету;
- кроссплатформенность;
- потребление системных ресурсов.

Точечная коррекция и пакетная обработка являются ключевыми функциями, необходимыми для редактирования изображений. Неперегруженный интерфейс обеспечивает удобство использования для широкого круга пользователей, включая начинающих. Работа без интернета позволяет применять приложение в любой ситуации, независимо от доступности сети.

Последние два критерия – кроссплатформенность и малое потребление ресурсов – оценивались как второстепенные, поскольку приложение предназначено для работы на десктопных системах и не предполагает сложных вычислений или высокой нагрузки.

Результаты сравнительного анализа по выделенным критериям разрабатываемого приложения и аналогов показан в таблице 1.

Таблица 1 – Результаты сравнительного анализа

Приложение \ Критерий	Приложение	Adobe Lightroom	GIMP	Snapseed	Pxlr Editor
Точечная коррекция параметров изображений	+	+	+	+	+
Возможность пакетной обработки	+	+	-	-	-
Неперегруженный интерфейс	+	-	-	+	+



Продолжение таблицы 1

Приложение \ Критерий	Прило- жение	Adobe Lightroom	GIMP	Snapseed	Pxlr Editor
Работа без подключения к интернету	+	+	+	+	-
Кроссплатфо- рменность	+	+	+	+	+
Малое потребление системных ресурсов	+	-	+	+	+

По результатам проведенного анализа можно сделать вывод, что некоторые аналогичные приложения обладают уникальными возможностями, такими как расширенные функции для детальной обработки изображений или наличие различных эффектов, которые могут быть полезны для профессиональных фотографов. Однако, многие из них имеют платные опции, рекламу, или требуют постоянного подключения к интернету. Кроме того, некоторые приложения слишком сложны для новичков, перегружены лишними функциями или потребляют значительные системные ресурсы, что может снизить их производительность, особенно при пакетной обработке изображений.

Изучив существующие аналоги, можно сделать вывод, что на рынке нет простого и понятного десктопного приложения для редактирования фотографий, которое могло бы быть удобно использовать студентам, начинающим фотографам и любителям фотографам. Многие из существующих решений либо слишком сложны, либо предлагают функции, которые не всегда необходимы для базового редактирования.

Таким образом, было принято решение разработать приложение, которое будет включать минимальный, но достаточный набор функций для редактирования фотографий, таких как:

- точечная корректировка параметров изображений (баланса белого, тона и присутствия);
- пакетная обработка фотографий;
- интуитивно понятный интерфейс с подсказками;
- наличие настольного и веб-приложения;
- оптимизация для малого потребления ресурсов.

Далее проведем анализ задания, выбор технологии и языка разработки.

### **1.3 Анализ задания и выбор технологии и языка разработки**

Прежде, чем приступать к разработке приложения, необходимо выбрать средства разработки.

В качестве основной технологии программирования для создания десктопного приложения для редактирования фотографий было выбрано объектно-ориентированное программирование (ООП). Это решение обусловлено тем, что основные сущности предметной области, такие как изображение, параметры редактирования и файлы, могут быть эффективно представлены в виде объектов, с которыми можно удобно работать и манипулировать [1].

Использование ООП позволяет разделить программу на независимые компоненты (объекты), каждый из которых отвечает за свою задачу, что идеально подходит для разработки сложных приложений с графическим интерфейсом. ООП способствует модульности, расширяемости и упрощает поддержку приложения в будущем. Кроме того, ООП является основой для создания большинства современных приложений и предоставляет гибкие и мощные средства для разработки.

В соответствии с техническим заданием, приложение должно работать на операционных системах семейства Windows, Linux и MacOS. Для разработки приложения был выбран язык программирования Java, так как он является высокоуровневым и кроссплатформенным, что позволит в дальнейшем запускать приложение на различных операционных системах. Для создания графического интерфейса и работы с изображениями был выбран фреймворк JavaFX, так как он предлагает широкий набор инструментов для разработки приложений с графическим интерфейсом. В качестве библиотеки для обработки изображений был использован OpenCV, которая идеально подходит для работы с различными операциями на изображениях, такими как изменение экспозиции, температуры и оттенка [4][5][6].

Для разработки интерфейса был использован SceneBuilder — инструмент, позволяющий создавать пользовательские интерфейсы с помощью визуального конструктора, что ускоряет процесс разработки и делает интерфейс более удобным для пользователя [7].

Использование CSS для стилизации обосновано тем, что оно позволяет отделить визуальное представление интерфейса от логики приложения, обеспечивая гибкость и удобство разработки. Стили, заданные в CSS, могут быть легко изменены и переиспользованы, что ускоряет процесс разработки и упрощает поддержку кода. Кроме того, CSS поддерживает наследование и каскадность, что позволяет централизованно управлять дизайном приложения, а также обеспечивает единообразие внешнего вида всех элементов интерфейса, независимо от их расположения в структуре FXML-документа [8][9].

## **2 Проектирование структуры и компонентов программного продукта**

Так как приложение рассчитано на широкую аудиторию, включая начинающих пользователей, удобный и интуитивно понятный интерфейс является ключевым элементом успешной работы программы.

Кроме того, интерфейс играет важную роль в структуре всего приложения. Хорошо продуманный интерфейс способствует улучшению взаимодействия пользователя с программой и повышает её удобство. Согласованный интерфейс позволяет избежать переписывания кода, так как разработки и изменения программного кода выполняются с учётом заранее определённого и утверждённого внешнего вида приложения. Понимание того, как будет выглядеть интерфейс, способствует формированию более чёткого представления о том, какие классы и методы должны быть реализованы в приложении.

В связи с этим было решено начать разработку с проектирования пользовательского интерфейса.

### **2.1 Проектирование интерфейса пользователя**

#### **2.1.1 Разработка диаграммы вариантов использования**

Для уточнения функциональности разрабатываемого приложения и его взаимодействия с пользователем было решено определить основные варианты использования и разработать диаграмму вариантов использования.

Диаграмма вариантов использования помогает выявить и чётко обозначить взаимодействие пользователя с приложением, а также выделить ключевые операции, которые должны быть реализованы для полноценной работы приложения. Этот подход позволяет более чётко понять требования и потребности конечного пользователя.

В результате анализа технического задания было выделен один пользователь и определены основные, вспомогательные и дополнительные варианты использования приложения.

К основным относятся такие варианты использования:

- точечное изменение параметров изображений;
- пакетная обработка изображений;
- предпросмотр внесенных изменений.

К вспомогательным относятся:

- загрузка изображений;
- сохранения изображений.

К дополнительным относится:

- сброс настроенных параметров.

Ниже приведены подробные описания основных вариантов использования приложения, которые приведены в таблицах 2–3.

Таблица 2 – Вариант использования «Редактирование изображения»

Действие пользователя	Отклик системы
1. Пользователь выбирает изображение для редактирования.	1. Приложение отображает выбранное изображение в окне для редактирования и ожидает изменения параметров изображений.
2. Пользователь изменяет параметры изображения	2. Приложение применяет измененные параметры и отображает измененное изображение
3. Пользователь сохраняет измененное изображение	3. Приложение создает новый файл с изображением с измененными параметрами в выбранной пользователем директории

Таблица 3 – Вариант использования «Редактирование пакета изображений»

Действие пользователя	Отклик системы
1. Пользователь выбирает пакет изображений для редактирования.	1. Приложение отображает пакет изображений в списке, отображает одно из изображений пакета в окне для редактирования и ожидает изменения параметров изображений
2. Пользователь изменяет параметры изображения	2. Приложение применяет измененные параметры и отображает измененное изображение
3. Пользователь применяет параметры ко всем изображениям из пакета	3. Приложение применяет измененные параметры к всем изображениям пакета
4. Пользователь сохраняет пакет изображений	4. Приложение сохраняет новые файлы изображений с примененными настройками в выбранной пользователем директории

Разработанная диаграмма вариантов использования представлена на рисунке 2.

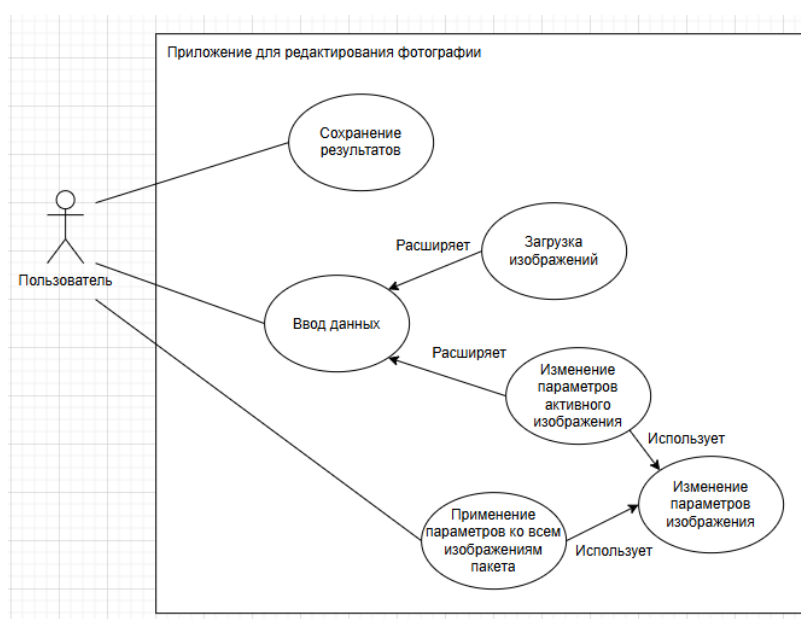


Рисунок 2 – Диаграмма вариантов использования

### 2.1.2 Разработка диаграммы состояний интерфейса

На основе анализа предметной области и требований к удобству использования, было решено разработать приложение с одним основным окном для редактирования изображений и отображением окна проводника для выбора изображений и директорий для сохранения изображений.

Использование одного окна в редакторе оправдано с точки зрения удобства и эффективности работы пользователя. Один экран позволяет сосредоточиться на выполнении основной задачи — редактировании фотографий — без необходимости переключаться между множеством окон. Это также уменьшает вероятность путаницы, так как вся информация и инструменты находятся в одном месте. Современные приложения для редактирования изображений часто используют такой подход, чтобы упростить взаимодействие с пользователем и обеспечить максимальную скорость работы [2].

В результате разработана диаграмма состояний интерфейса, которая представлена на рисунке 3.

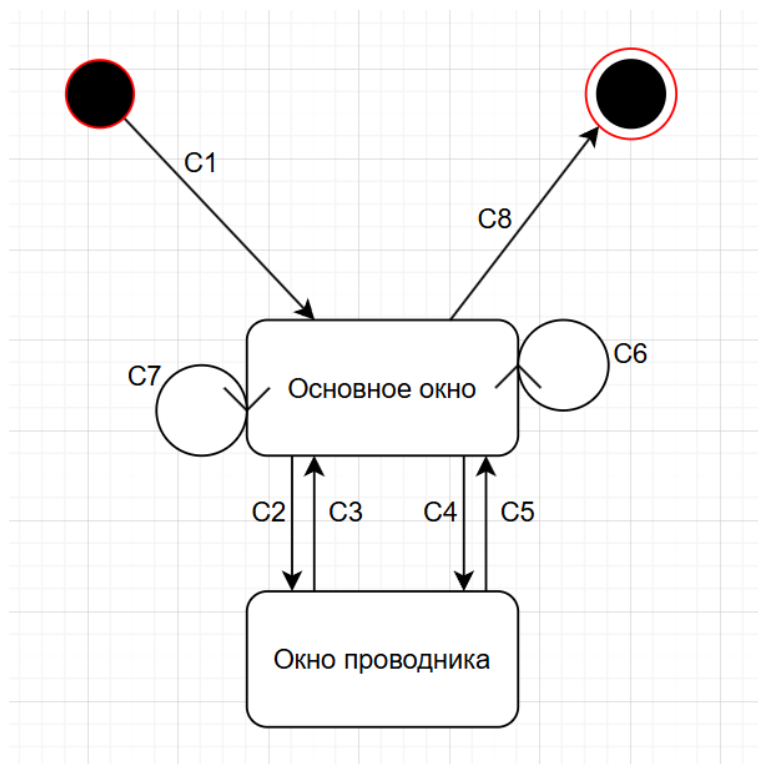


Рисунок 3 – Граф состояний интерфейса

- C1 – Активация приложения;
- C2 – переход в состояние «Выбор изображений» - выбор изображений для редактирования в окне проводника;
- C3 – переход в состояние «Открыть и закрыть» - открытие изображений, закрытие окна и переход в основное окно приложения;
- C4 – переход в состояние «Выбор директории сохранения» - выбор директории для сохранения изображений в окне проводника;
- C5 – переход в состояние «Сохранить и закрыть» - сохранение изображений, закрытие окна и переход в основное окно приложения;
- C6 – переход в состояние «Изменить изображение» - изменить параметры активного изображения;
- C7 – переход в состояние «Изменить пакет» – изменить параметры всех изображений в пакете;
- C8 – переход в состояние «Выход» – корректное завершение приложения.

### **2.1.3 Разработка форм интерфейса**

Приложение должно обеспечивать прямое и удобное взаимодействие с пользователем, поэтому было принято решение разработать понятный и интуитивно понятный интерфейс. Как было сказано выше, выбрано решение с одним окном, в котором сосредоточены все основные функции приложения.

В основном окне расположены все элементы управления для редактирования фотографий. Для загрузки и сохранения изображений были добавлены кнопки. Основные параметры для изменения регулируются с помощью текстовых полей ввода и ползунков, что даёт пользователю точный контроль над значениями и позволяет мгновенно видеть изменения на изображении. Такие элементы управления являются удобными для быстрого и точного редактирования параметров фотографий. Для удобства работы также добавлена кнопка сброса настроек.



Размеры окна зафиксированы, чтобы избежать возможных проблем с адаптивностью и не перегружать пользователя дополнительными настройками. Так как адаптивный интерфейс может быть сложным в реализации, было решено запретить пользователю изменять размеры окна, что обеспечивает стабильную и предсказуемую работу приложения. Разработанный вид главного меню приложения представлен на рисунке 4.

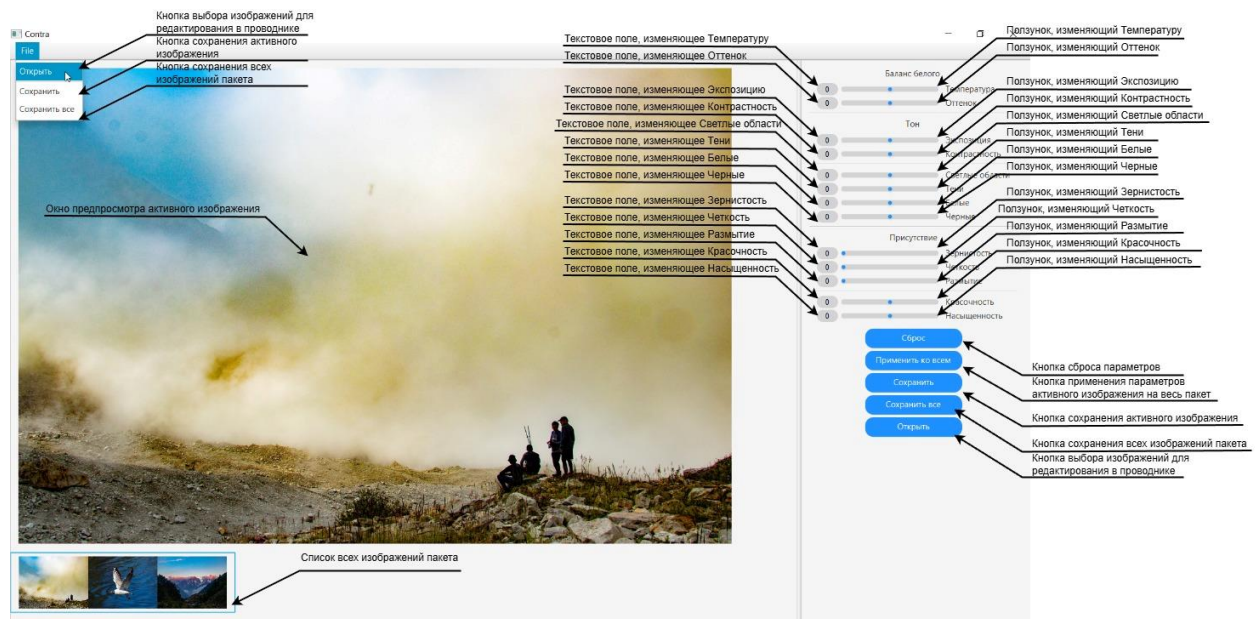


Рисунок 4 – Вид главного окна приложения

## 2.2 Разработка диаграммы пакетов

Проведя анализ концептуальной модели и вариантов, можно выделить следующие пакеты:

- 1) Пользовательский интерфейс - классы, реализующие объекты пользовательского интерфейса;
- 2) Библиотека интерфейсных компонентов — классы JavaFX, реализующие интерфейсные компоненты: окна, кнопки, ползунки, отображение изображения и т. п.;
- 3) Объекты управления — класс контроллера и класс-обработчик изображения;
- 4) Объекты задачи - классы, реализующие объекты предметной области;

- 5) Базовые структуры данных – классы из Java Collections, реализующие внутренние структуры данных: словари, списки и т. п.;
- 6) Обработка ошибок - классы исключений, реализующие обработку ошибок.

В результате получаем диаграмму пакетов, изображенная на рисунке 5.



Рисунок 5 – Диаграмма пакетов

## 2.3 Разработка диаграмм классов приложения

### 2.3.1 Разработка диаграмм классов интерфейсной части

Реализация классов интерфейсной части выполняется в соответствии с разработанным интерфейсом и с использованием встроенных средств платформы JavaFX, которая предоставляет богатый набор графических компонентов. Основные используемые компоненты включают Label (для текстовых подписей), TextField (для ввода значений параметров), Slider (ползунки для ввода значений параметров), Button (кнопки) и ImageView (для отображения изображений). Эти элементы позволяют эффективно реализовать весь необходимый функционал редактирования изображений.

Для реализации интерфейса основного окна приложения был создан класс MainApplication, который отвечает за отображение всех ключевых

компонентов. Основное окно включает кнопку для загрузки изображения (кнопка «Загрузить»), кнопку для сохранения результата (кнопка «Сохранить» и «Сохранить все»), набор текстовых полей и ползунков для редактирования параметров, превью изображения, список всех загруженных изображений, кнопку сброса изменений (кнопка «Сброс») и кнопку применения параметров ко всем изображениям в пакете (кнопка «Применить ко всем»).

Класс `MainWindow` наследуется от базового класса `Application`, который предоставляет все необходимые методы для управления окном и взаимодействия с компонентами. В атрибутах класса `MainApplication` определены поля `FXMLLoader`, `Scene` и `Stage`. `FXMLLoader` загружает интерфейс пользователя из `FXML`-документа, `Scene` представляет содержимое окна, `Stage` представляет само окно приложения. В методах класса `MainApplication` будут два метода `start(Stage stage)` и `main(String... args)`. Метод `start(Stage stage)` представляет собой точку для входа `JavaFX`-приложения и вызывается автоматически, в этом методе задается окно приложения `Stage` и содержимое окна `Scene`. Метод `main(String... args)` будет запускать приложение `JavaFX` с помощью статического метода `launch()` библиотеки `JavaFX`.

`FXML`-документ, описывающий расположение элементов, создан автоматически с помощью `SceneBuilder`. Стилизация описана в `CSS`-документе.

В результате получается диаграмма классов главного окна, изображенная на рисунке 6.

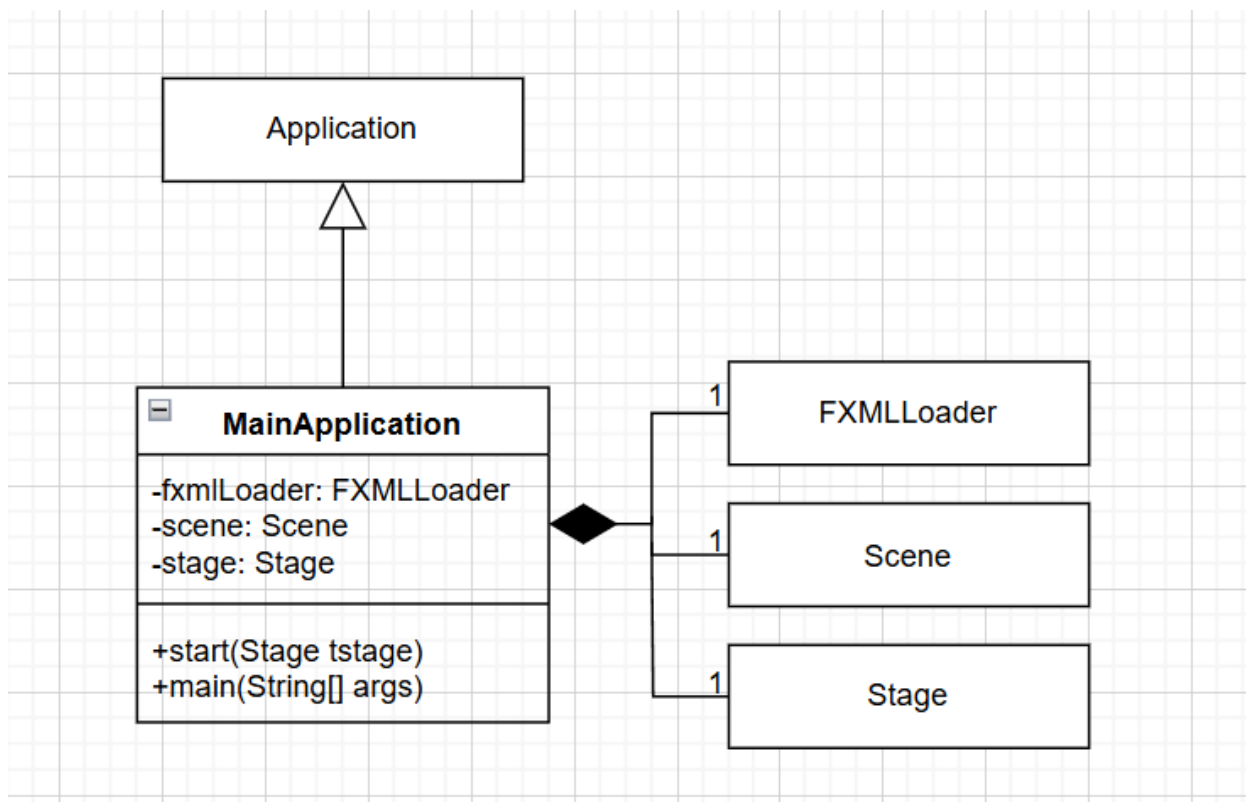


Рисунок 6 – Диаграмма классов главного окна

### 2.3.2 Разработка диаграмм классов функциональной части

В FXML-документе помимо элементов интерфейса указывается контроллер, отвечающий за программную логику, обработку действий пользователя и обновление данных. Контроллер играет ключевую роль в разделении логики пользовательского интерфейса и его визуального представления. Именно контроллер связывает элементы интерфейса с кодом, обрабатывает события, управляет данными и их отображением.

Таким образом, использование контроллера позволяет разделить приложение на часть, описывающую представление в FXML-документе и описанном выше классе MainApplication и часть, описывающую логику и обрабатывающую события.

В атрибутах класса MainController определены поля imageView, которое будет хранить экземпляр класса ImageView, отображающий предпросмотр выбранного изображения, editingImages, которое будет хранить список всех изображений пакета, curImage, которое будет хранить активное изображение.

В методах класса MainController определены методы:

- `onSliderDrag(Event event)` и `onTextFieldEdit(Event event)`, связывающие значения ползунков и текстовых полей с помощью статических методов вспомогательного класса `TextSliderConnct`;
- `changeSetting(Event event)`, `changeImage(EditingImage editingImage, String settingName, double coef)`, `resetSetting()` отвечающие за изменение параметров активного изображения в окне предпросмотра;
- `changeAllImages()`, применяющий параметры активного изображения ко всем изображениям пакета;
- `openFiles()`, `saveFile()`, `saveAllFiles()`, `saveFilesToDirectory(List<String> files)`, `getDirectory()`, обеспечивающие загрузку изображений в приложение и их сохранение;
- `addImagesToList(List<File> files)`, `setCurImage(VouseEvent event)`, `setImageToImageView(EditingImage editingImage)`, `clear()`, `findEditingImage()` отвечающие за отображение изображений в окне приложения.

Вспомогательный класс `TextSliderConnect` обеспечивает связывание значений ползунков и текстовых полей. Он включает в себя статические методы `sliderDrag(Event event, ImageView imageView)` и `textFieldEdit(Event event, ImageView imageView)`, которые достают из `event` идентификатор изменившегося ползунка или текстового поля и изменяют значение парного текстового поля или ползунка соответственно, и приватного метода `getObjectsById(ImageView imageView, String sliderId, String textFieldId)`, который с помощью идентификатора ползунка и текстового поля находит и возвращает объекты `Slider` и `TextField`.

Вспомогательный класс `Helper` обеспечивает вспомогательный функционал разного рода. Он включает в себя статические методы `getId(Event event)`, возвращающий идентификатор ячейки вызвавшей `event`, `fileToString(File file)`, возвращающий адрес файла в корректном строковом представлении, `getImgName(String img)`, извлекающий из адреса файла в строковом представлении имя самого изображения без пути к нему.

Получаем диаграмму классов, изображенную на рисунке 7.

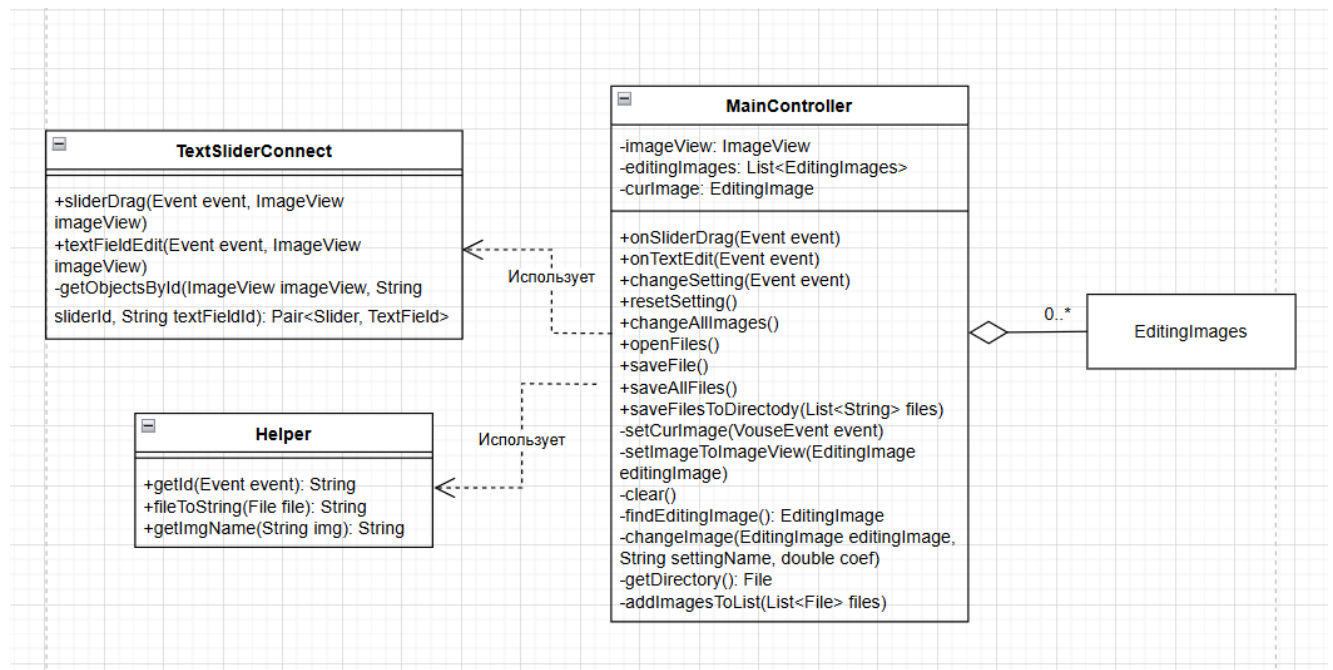


Рисунок 7 – Диаграмма классов контроллера

В основе работы с параметрами изображений лежит три класса – абстрактный класс `Setting`, класс `EditSetting` и класс `EditingImage`.

Класс `Setting` является родителем для всех параметров изображения. Он включает в себя поле `coef` – коэффициент параметра – и методы `convertToCoefficient(double value)`, `getCoef()`, `setCoef(double coef)`, `setCoefWithoutConvertation(double coef)`, `resetCoef()`, `chooseCorrectValue(double value)` для работы с коэффициентом и абстрактный метод `applySetting(Mat image)`, применяющий изменения к изображению.

От `Setting` наследуется абстрактный класс `SettingPixelEdit`, включающий поля `changeB`, `changeR`, `changeG`, которые представляют собой функции, изменяющие по некоторой формуле каждый из компонентов RGB - blue, red и green соответственно. `SettingPixelEdit` определяет методы `applySetting(Mat image)`, `getAverage(double[] rgb)`, возвращающий среднее значение компонентов RGB и методы `changeG(double g)`, `changeB(double b)`, `changeR(double r)`, `changePixel(double[] rgb)`, применяющие описанные в `changeB`, `changeR`, `changeG` функции к пикселю изображения.

От `SettingPixelEdit`, в свою очередь, наследуется класс `SettingWithSameCoefs`, от которого будут унаследованы все классы, описывающие такие параметры изображения, в которых каждый из компонентов RGB изменяется по одной и той же формуле. В `SettingWithSameCoefs` определено поле `changeRGB`, описывающее эту единственную формулу, и переопределен метод `changePixel(double[] rgb)`, приравнивающий каждый из `changeB`, `changeR`, `changeG` к `changeRGB` и вызывающий `changePixel(double[] rgb)` родителя.

Все остальные классы наследуются от этих трех, в большинстве своем не являются абстрактными и представляют собой конкретный параметр изображения. В этих «конечных» классах так или иначе определяются значения `changeB`, `changeR`, `changeG` или `changeRGB` и в случае необходимости незначительно переопределяются методы и добавляются вспомогательные поля, хранящие необходимые для расчета нового значения пикселя величины. В связи с незначительностью отличий от родительских классов эти классы подробно описано не будут.

Получаем диаграмму классов, изображенную на рисунке 8.

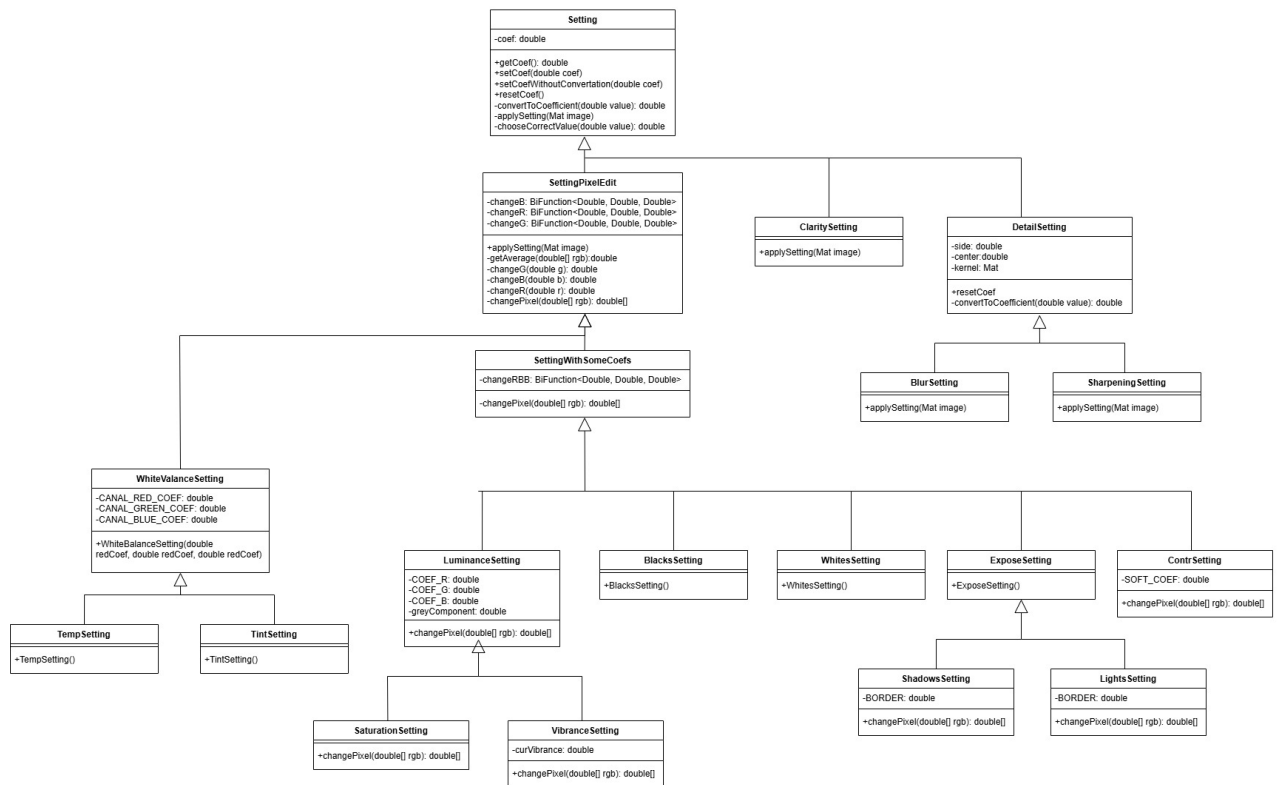


Рисунок 8 – Диаграмма классов параметров изображения

Класс `EditingImage` представляет собой контейнер, хранящий в себе множество реализаций класса `Setting` и методы для работы с ними. `EditingImage` включает «служебные» поля - `original`, `editingCopy`, `editingCopyRel` и `imageView`, которые хранят в себе пути файлов оригинального изображения, рабочей копии изображения (в которую записываются изменения параметров), относительного пути файла рабочей копии и объект `imageView`, отображающий изображение, соответственно – и поля, обеспечивающие корректное изменение параметров – множество реализаций `Setting`, экземпляр класса `EditSetting` и словарь `settingsMap`, связывающий идентификаторы ползунков и текстовых полей основного окна с реализациями `Setting`.



EditingImage описывает:

- методы `changeImage(String settingName, Double coef)` и `changeImage()`, обеспечивающие изменение изображения;
- методы `changeSettings(Map<String, Setting> newSettingsMap)`, `resetSettings()`, обеспечивающие манипуляции с коэффициентами параметров;
- конструктор `EditingImage(String original, ImageView imageView)`, который в том числе создает рабочую копию.

Класс `EditSetting` представляет функционал для применения изменений к изображению в соответствии с конкретными реализациями `Setting`. Он включает в себя поле `calls` – список вызовов, определяющий порядок, в котором будут накладываться изменения параметров на изображение. `EditSetting` определяет методы `changeImage(Mat image)`, изменяющий изображение по хранящимся в `calls` параметрам, `initSettings(Collection<Setting> settings)`, инициализирующий `calls`, и `resetSettings()`, сбрасывающий коэффициенты всех параметров в `calls`.

Получаем диаграмму классов, изображенную на рисунке 9.

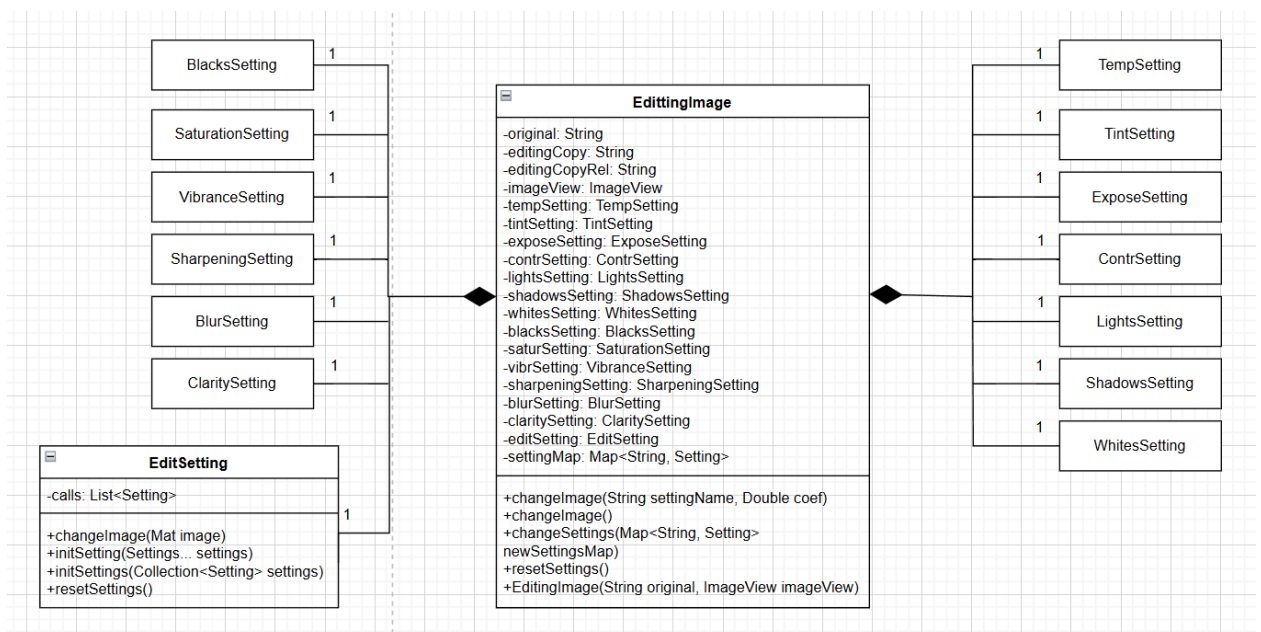


Рисунок 9 – Диаграмма классов работы с изображением

## 2.4 Демонстрация работы функций приложения

Результатом проектирования является разработанное приложение, которое обладает всеми перечисленными в техническом задании функциями.

На рисунках 10-15 показано выполнение функций приложения.

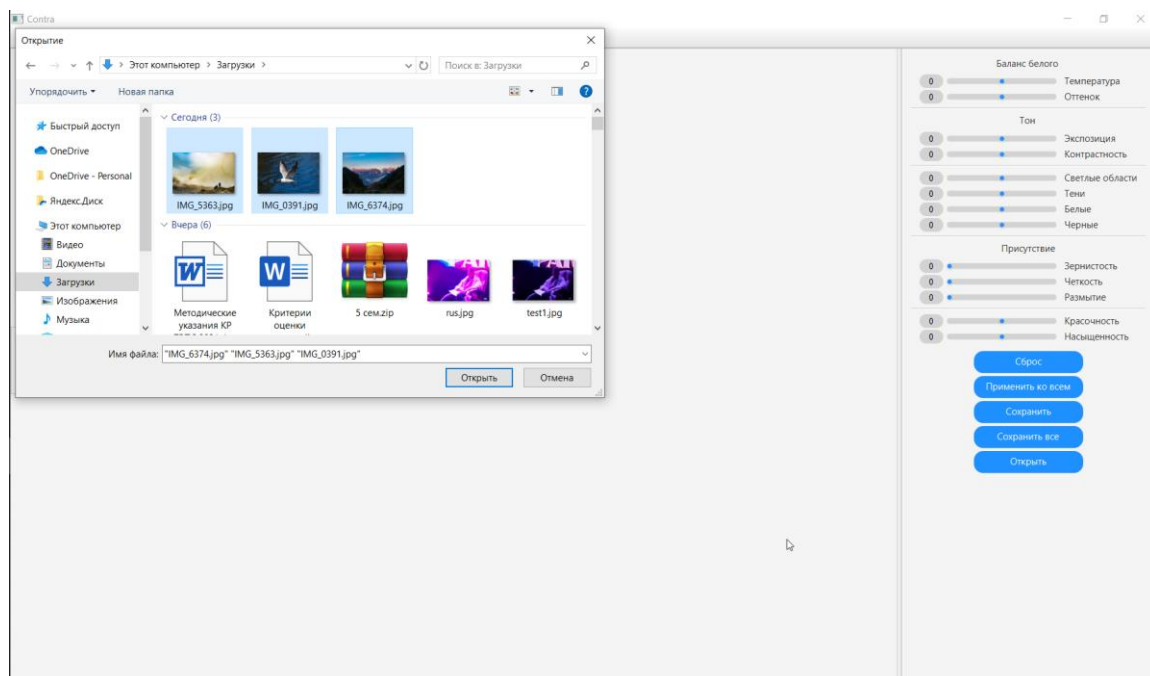


Рисунок 10 – Выбор изображений для редактирования в проводнике

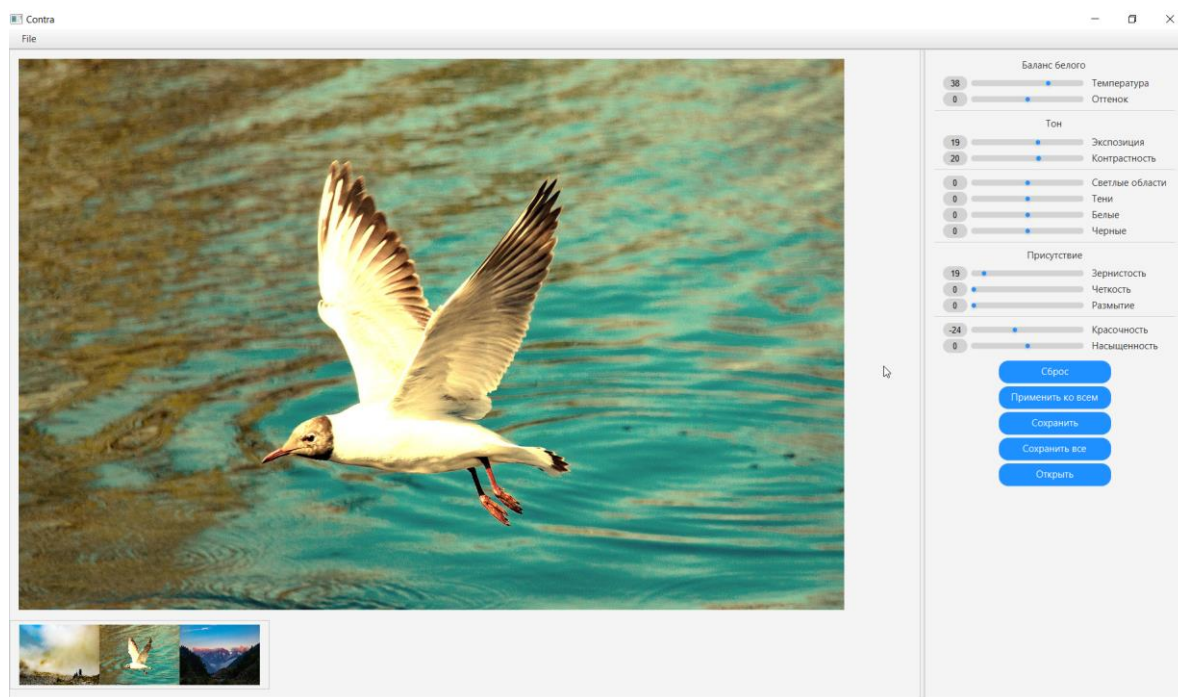


Рисунок 11 – Изменение параметров «Температура», «Экспозиция», «Контрастность», «Зернистость», «Красочность» изображения

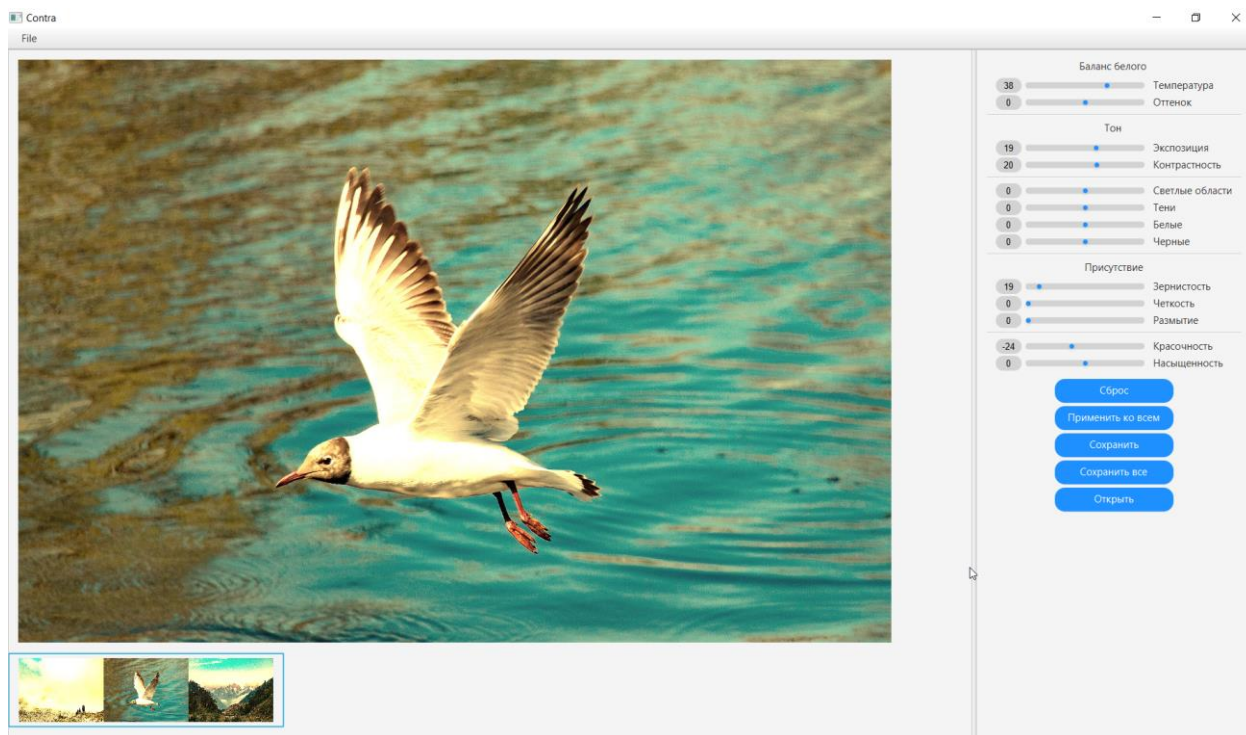


Рисунок 12 – Применение параметров ко всем изображениям в пакете

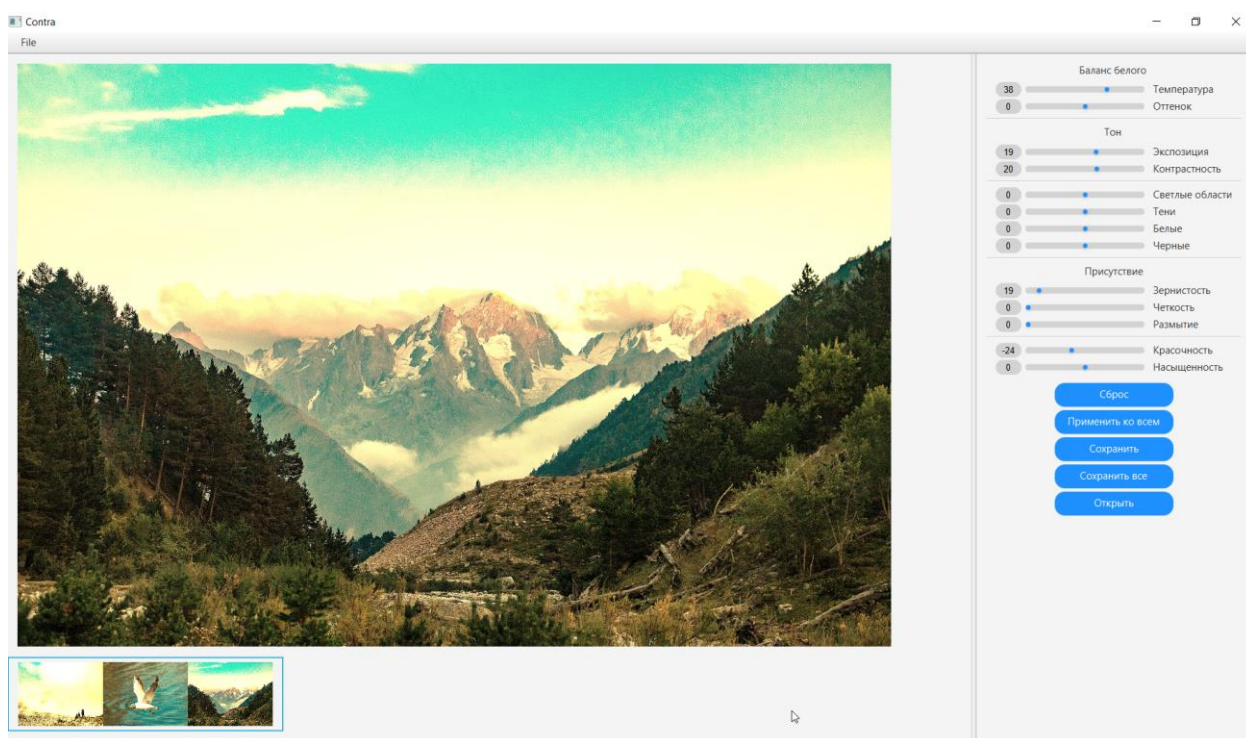


Рисунок 13 – Выбор другого изображения в качестве активного



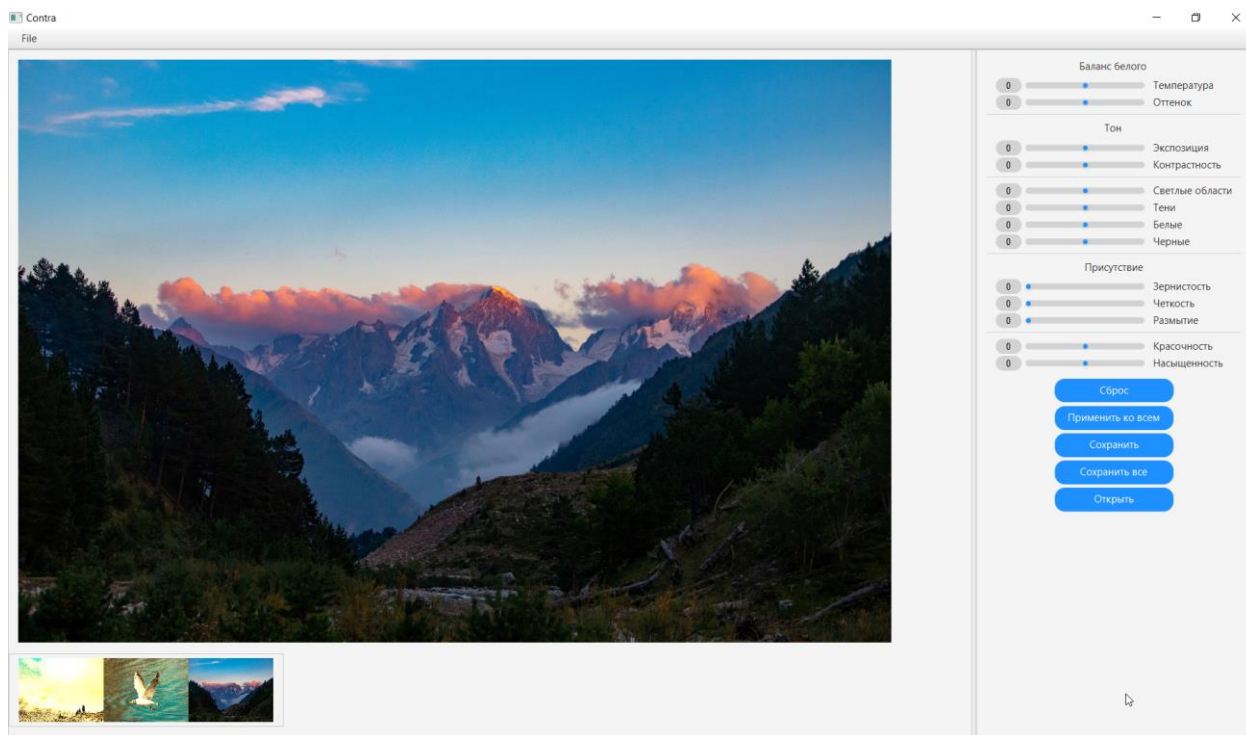


Рисунок 14 – Сброс настроек

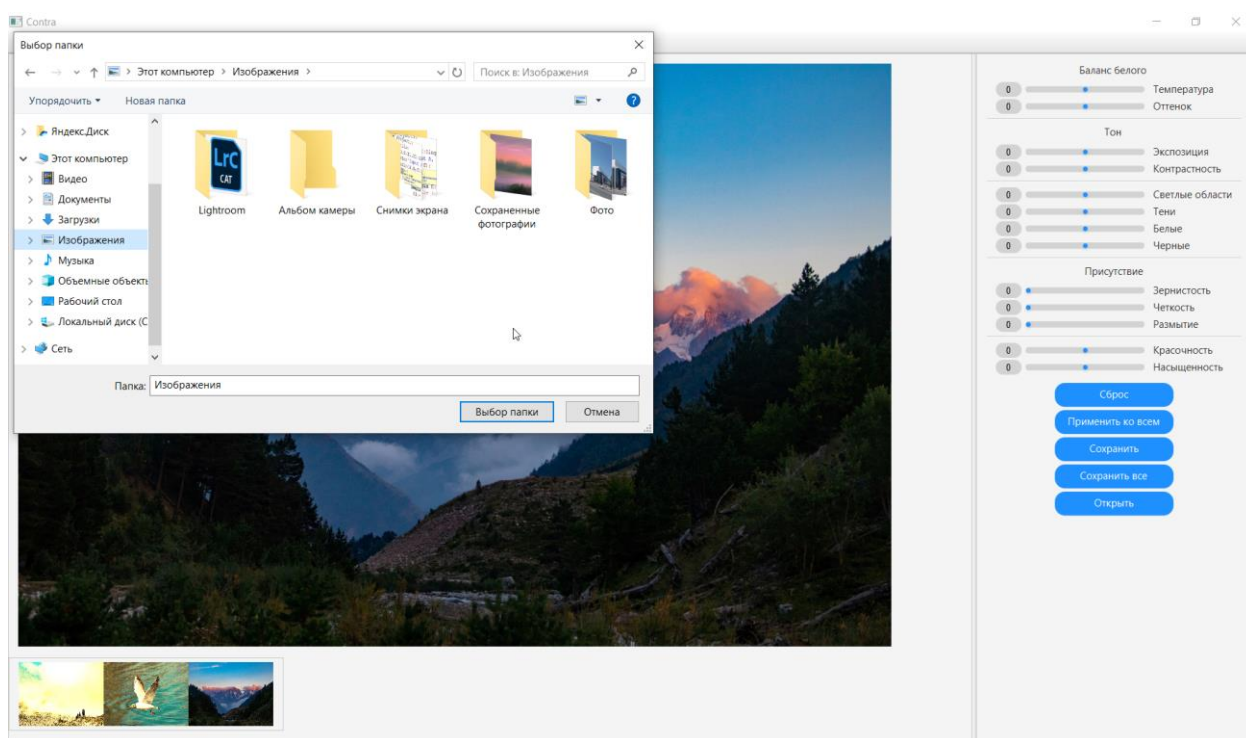


Рисунок 15 – Выбор директории для сохранения

### **3 Выбор стратегии тестирования и разработка тестов**

На этапе разработки модулей приложения было принято решение использовать метод тестирования ручным контролем. Этот подход оказался наиболее экономичным с точки зрения затрат времени и ресурсов, а также позволил выявить общие ошибки на раннем этапе написания кода [3].

В процессе проектирования классов интерфейса приложения применялось функциональное тестирование. Этот вид тестирования был выбран из-за его способности проверить, как программа реагирует на различные действия пользователя, а также предотвратить случаи, когда как корректные, так и некорректные действия приводят к сбоям. Функциональное тестирование проводилось с использованием следующих методов:

- метод эквивалентных разбиений;
- метод анализа граничных условий;
- предположение об ошибке.

Было также выполнено оценочное тестирование, которое позволило глубже понять потребности пользователей и их ожидания благодаря объективной оценке приложения людьми, не участвовавшими в его разработке.

#### **3.1 Тестирование ручным контролем**

Одним из подходов к тестированию является структурный метод, основанный на многолетнем опыте. Со временем разработчики и тестировщики накопили обширный список вопросов и проверок, направленных на обнаружение распространенных ошибок, которые могут оставаться незаметными в коде. Результаты тестирования структурным методом приведены в таблице 4.

Таблица 4 – Результаты тестирования ручным контролем

Контроль обращений к данным		
№ вопроса	Результат проверки	Вывод
Все ли элементы инициализировали?	Поля классов либо инициализируются по умолчанию, либо инициализируются в конструкторе. Локальные переменные инициализируются в тот момент, когда эти переменные требуются.	Все переменные инициализированы.
Не превышены ли максимальные (или реальные) размеры массивов и строк?	Массивы не инициализируются пользователем напрямую и их размеры не превышаются. Однако на вводимые строки нет ограничения, значит их размер может быть превышен.	Размеры массивов не превышены. Размеры строк могут быть превышены.
Не перепутаны ли строки со столбцами при работе с матрицами?	Работа с RGB-матрицей проводится один раз в методе applySetting(Mat image) класса SettingPixelEdit. Строки со столбцами не перепутаны.	Строки со столбцами не перепутаны.

Продолжение таблицы 4

№ вопроса	Результат проверки	Вывод
Присутствуют ли переменные со сходными именами?	Да, присутствуют, однако схожесть их имен обусловлена схожестью заложенного в переменные смысла.	Переменные со сходными именами обусловлены.
Используются ли файлы? Если да, то при вводе из файла проверяется ли завершение файла?	Приложение работает с изображениями, обработка которых происходит посредством методов библиотеки OpenCV.	При вводе из файла ошибок быть не может.
Соответствуют ли типы записываемых и читаемых значений?	Соответствуют.	Типы записываемых и читаемых значений соответствуют.
Использованы ли нетипизированные переменные?	Нетипизированные переменные не используются.	Нетипизированные переменные не используются.
Использованы ли открытые массивы? Не выходят ли индексы за границы массивов?	Используется экземпляр класса ArrayList() из Java Collections, который автоматически выделяет достаточное количество памяти при инициализации, циклы перебирают значения до границы массива.	Открытые массивы используются, но их использование безопасно. Индексы за границы массивов не выходят.

Продолжение таблицы 4

№ вопроса	Результат проверки	Вывод
Использована ли динамическая память?	Динамическая память используется, однако отсутствие утечек обеспечивает Java Garbage Collector.	Утечек памяти нет.
Контроль вычислений		
№ вопроса	Результат проверки	Вывод
Правильно ли записаны выражения (порядок следования операторов)?	Порядок следования не нарушен, иначе ошибка была бы подсвечена IDEA.	Выражения записаны правильно.
Корректно ли выполнены вычисления над неарифметическими переменными?	Используемые неарифметические переменные это строки. Манипуляции со строками проводятся корректно.	Вычисления неарифметических выполнены корректно.
Корректно ли выполнены вычисления с переменными различных типов?	Вычисления с переменными различных типов не выполняются.	Ошибок в вычислении переменных различных типов быть не может



Продолжение таблицы 4

№ вопроса	Результат проверки	Вывод
Возможно ли переполнение разрядной сетки или ситуация машинного нуля?	Переполнение разрядной сетки возможно, поскольку вводимые значения не ограничены.  Ситуация машинного нуля возможна, однако округление до нуля можно расценивать как корректное поведение.	Переполнение разрядной сетки или ситуация машинного нуля возможны.
Соответствуют ли вычисления заданным требованиям точности?	Соответствуют.	Вычисления соответствуют заданным требованиям точности.
Присутствуют ли сравнения переменных различных типов?	Нет.	Сравнения переменных различных типов нет.
Контроль передачи управления		
№ вопроса	Результат проверки	Вывод
Будут ли корректно завершены циклы?	Циклы завершаются корректно.	Циклы будут завершены корректно.

Продолжение таблицы 4

№ вопроса	Результат проверки	Вывод
Будет ли корректно завершена программа?	В случае ошибок все выбрасываемые исключения перехватываются и обрабатываются.	Программа будет завершена корректно.
Существуют ли циклы, которые не будут выполняться из-за нарушения условия входа?	Таких циклов нет.	Таких циклов нет.
Существуют ли поисковые циклы? Корректно ли отрабатываются ситуации «элемент найден» и «элемент не найден»?	Таких циклов нет.	Таких циклов нет.
Соответствуют ли списки параметров и аргументов по порядку, типу, единицам измерения?	Списки параметров соответствуют спискам аргументов.	Списки параметров по порядку, типу, единицам измерения совпадают с списками аргументов.

Продолжение таблицы 4

Контроль межмодульных интерфейсов		
№ вопроса	Результат проверки	Вывод
Не изменяет ли подпрограмма аргументов, которые не должны изменяться?	Аргументы, которые не должны изменяться, не изменяются.	Методы классов не изменяют аргументы, которые изменяться не должны.
Не происходит ли нарушения области действия глобальных и локальных переменных с одинаковыми именами?	Нарушения области действия не происходит.	Нарушения области действия глобальных и локальных переменных с одинаковыми именами не происходит.

Тестирование ручным контролем помогло найти некоторые ошибки в коде. В ходе разработки было введено ограничение на длину вводимой в текстовые поля строки.

### 3.2 Функциональное тестирование

Функциональное тестирование представляет под собой тестирование по принципу “черного ящика”. В этом случае программа рассматривается как «черный ящик», и целью тестирования является выяснение обстоятельств, в которых поведение программы не соответствует спецификации. В тестировании используются четыре вида тестовых наборов:

- эквивалентное разбиение;
- анализ граничных значений;
- анализ причинно-следственных связей;
- предположения об ошибке.

### 3.2.1 Метод эквивалентных разбиений

Метод эквивалентного разбиения работает следующим образом. Все возможные варианты входных данных программы для каждого параметра разделяются на ограниченное количество групп, называемых классами эквивалентности. Данные внутри одного класса объединяются на основании того, что они выявляют одни и те же ошибки: если определенный набор данных из класса обнаруживает ошибку, предполагается, что остальные данные этого класса также укажут на эту же ошибку, и наоборот.

Классы эквивалентности можно разбить на числа и строки/символы. Числа можно разбить на рациональные и целые. Целые числа можно разбить на отрицательные и положительные.

Правильным классом эквивалентности для текстового поля ввода значения параметров «Зернистость», «Четкость» и «Размытие» является целое число в диапазоне от 0 до 200. Неправильным классом эквивалентности строки является все остальное.

Правильным классом эквивалентности для текстового поля ввода значения всех остальных параметров является целое число в диапазоне от -100 до 100. Неправильным классом эквивалентности строки является все остальное.

Тестирование эквивалентным разбиением показано в таблице 5.

Таблица 5 – Тестирование методом эквивалентных разбиений

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
1	Проверка текстового поля ввода параметра «Четкость»	Правильный ввод: Число в диапазоне от 0 до 200	Изменение параметра «Четкость» изображения	Изменение параметра «Четкость» изображения	Программа отреагировала корректно

Продолжение таблицы 5

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
2	Проверка текстового поля ввода параметра «Четкость»	Неправильный ввод: Вещественное число	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно
3	Проверка текстового поля ввода параметра «Четкость»	Неправильный ввод: “aaa”	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно
4	Проверка текстового поля ввода параметра «Четкость»	Неправильный ввод: Число не лежащее в диапазоне от 0 до 200	Открытие всплывающего окна с сообщением о некорректном вводе	Изменение параметра «Четкость» изображения	Программа отреагировала некорректно
5	Проверка текстового поля ввода параметра «Оттенок»	Правильный ввод: Число в диапазоне от -100 до 100	Изменение параметра «Оттенок» изображения	Изменение параметра «Оттенок» изображения	Программа отреагировала корректно

Продолжение таблицы 5

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
6	Проверка текстового поля ввода параметра «Оттенок»	Неправильный ввод: Вещественное число	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно
7	Проверка текстового поля ввода параметра «Оттенок»	Неправильный ввод: “aaa”	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно
8	Проверка текстового поля ввода параметра «Оттенок»	Неправильный ввод: Число не лежащее в диапазоне от -100 до 100	Открытие всплывающего окна с сообщением о некорректном вводе	Изменение параметра «Оттенок» изображения	Программа отреагировала некорректно

Остальные поля ввода были протестированы аналогичным образом. Метод эквивалентных разбиений помог найти ошибку – отсутствие проверки на принадлежность введенного значения к правильному числовому диапазону. Перед проведением тестирования методом анализа граничных значений ошибка была исправлена.

### 3.2.2 Метод анализа граничных значений

Этот метод подразумевает тестирование значениями на границах классов эквивалентности, которые могут привести к ошибке.

Поскольку входные значения являются дискретным рядом значений (целые числа от 0 до 200 или целые числа от -100 до 100), то следует проверить значения ввода -1, 0, 200 и 201 для полей параметров «Зернистость», «Четкость» и «Размытие» и значения ввода -101, -100, 100 и 101 для всех остальных записей.

Тестирование граничными значениями показано на таблице 6.

Таблица 6 – Тестирование граничными значениями

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
1	Проверка текстового поля ввода параметра «Четкость»	-1	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно
2	Проверка текстового поля ввода параметра «Четкость»	0	Изменение параметра «Четкость» изображения	Изменение параметра «Четкость» изображения	Программа отреагировала корректно

Продолжение таблицы 6

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
3	Проверка текстового поля ввода параметра «Четкость»	200	Изменение параметра «Четкость» изображения	Изменение параметра «Четкость» изображения	Программа отреагировала корректно
4	Проверка текстового поля ввода параметра «Четкость»	201	Открытие всплывающего о окна с сообщением о некорректном вводе	Открытие всплывающего о окна с сообщением о некорректном вводе	Программа отреагировала корректно
5	Проверка текстового поля ввода параметра «Экспозиция»	-101	Открытие всплывающего о окна с сообщением о некорректном вводе	Открытие всплывающего о окна с сообщением о некорректном вводе	Программа отреагировала корректно
6	Проверка текстового поля ввода параметра «Экспозиция»	-100	Изменение параметра «Экспозиция» изображения	Изменение параметра «Экспозиция» изображения	Программа отреагировала корректно



Продолжение таблицы 6

№ т.	Назначение теста	Значения исходных данных	Ожидаемый результат	Реакция программы	Вывод
7	Проверка текстового поля ввода параметра «Экспозиция»	100	Изменение параметра «Экспозиция» изображения	Изображение стало полностью белым	Программа отреагировала некорректно
8	Проверка текстового поля ввода параметра «Светлые области»	100	Изменение параметра «Светлые области» изображения	Светлые области стали полностью белыми	Программа отреагировала некорректно
9	Проверка текстового поля ввода параметра «Тени»	100	Изменение параметра «Тени» изображения	Тени стали полностью белыми	Программа отреагировала некорректно
10	Проверка текстового поля ввода параметра «Экспозиция»	101	Открытие всплывающего окна с сообщением о некорректном вводе	Открытие всплывающего окна с сообщением о некорректном вводе	Программа отреагировала корректно

Остальные поля ввода были протестированы аналогичным образом. Метод анализа граничных значений помог найти ошибку – некорректное поведение приложения при значении параметров «Экспозиция», «Тени» и «Светлые области», близким к 100.

### 3.3 Оценочное тестирование

После проверки работоспособности приложения проведено оценочное тестирование с участием конечных пользователей, которые смогут дать свою оценку разработанному продукту. Тестирование проводилось по трем основным критериям: удобство эксплуатации, удобства установки, корректность выполнения. Оценка производилась по 10-ти балльной шкале.

Результаты опроса показаны в таблице 7.

Таблица 7 - Оценка качества программы по 10-балльной шкале.

№ пользователя	Удобство эксплуатации	Удобство установки	Корректность выполнения
1	8	6	7
2	9	5	8
3	9	6	8
4	10	6	9
5	9	5	8
Средняя оценка	9	5,6	8

Результаты тестирования показали, что приложение в целом функционирует корректно, однако пользователи отметили неудобство установки и запуска приложения. Кроме того, пользователям хотелось бы видеть больше функционала. В будущих версиях планируется учесть их пожелания: реализовать дополнительный функционал и написать установочный и запускающий скрипт.

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы по дисциплине "ТРПС" было создано десктопное приложение для редактирования фотографий, которое позволяет изменять параметры изображений и осуществлять пакетную обработку файлов.

В процессе работы были:

- Закреплены навыки работы в JavaFX, SceneBuilder, использования библиотеки OpenCV и CSS, получен опыт написания кода на Java;
- Проанализированы предметная область и существующие аналоги;
- Разработана структура и компоненты приложения;
- Реализованы заявленные функции и проведено тестирование приложения;
- Оформлена расчетно-пояснительная записка.

Разработанное приложение удовлетворяет требованиям, изложенным в техническом задании.

В будущем планируется расширить функционал программы - добавить более тонкую настройку параметров редактирования и встроить в приложение нейросеть, автоматизирующую рутинную часть обработки изображений.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Иванова Г.С. Лекции по Объектно-ориентированному программированию.
- 2 Иванова, Г.С. Технология программирования: Учебник для вузов – М.: Изд-во МГТУ им. Н.Э. Баумана, 2003.
- 3 Иванова Г.С., Ничушкина Т.Н. Тестирование программного обеспечения: Методические указания по выполнению курсовой работы по дисциплине "Технология разработки программных систем". – М.: Электронное учебное издание МГТУ им. Н.Э. Баумана, 2021.
- 4 Документация по Java. URL: <https://docs.oracle.com/en/java/> (дата обращения: 03.10.2024)
- 5 Документация по JavaFX. URL: <https://fxdocs.github.io/docs/html5/> (дата обращения: 06.10.2024)
- 6 Документация по OpenCV для Java. URL: <https://docs.opencv.org/4.x/javadoc/index.html> (дата обращения: 07.10.2024)
- 7 JavaFX Scene Builder User Guide. URL: [https://docs.oracle.com/javafx/scenebuilder/1/user\\_guide/jsbpub-user\\_guide.htm](https://docs.oracle.com/javafx/scenebuilder/1/user_guide/jsbpub-user_guide.htm) (дата обращения: 06.10.2024)
- 8 JavaFX CSS Reference Guide. URL: <https://openjfx.io/javadoc/23/javafx.graphics/javafx/scene/doc-files/cssref.html> (дата обращения: 06.10.2024)
- 9 Документация по FXML. URL: [https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction\\_to\\_fxml.html](https://docs.oracle.com/javase/8/javafx/api/javafx/fxml/doc-files/introduction_to_fxml.html) (дата обращения: 06.10.2024)

**ПРИЛОЖЕНИЕ А**  
**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

Листов 7



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_

КАФЕДРА \_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_09.03.01 Информатика и вычислительная техника

ПРИЛОЖЕНИЕ ДЛЯ РЕДАКТИРОВАНИЯ ФОТОГРАФИЙ  
“ФОТОМАСТЕР”

Техническое задание на курсовую работу  
по дисциплине Технология разработки программных систем

Листов 7

Студент ИУ6-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Дорохов Р.В.  
(И.О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Москва 2024 г.

## 1 ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку компьютерного приложения «ФотоМастер» [ФМ], используемого на этапе постобработки фотографий и предназначенного для улучшения их качества.

Программа актуальна в связи с распространением цифровых камер, ростом их компактности и появляющейся потребностью в инструментах для их отбора. В отличие от аналогов программа будет распространяться с открытым исходным кодом и имеет менее перегруженный интерфейс, что делает ее более привлекательной для начинающих фотографов.

## 2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Приложение ФМ разрабатывается на основе учебного плана кафедры ИУ6 «Компьютерные системы и сети» факультета ИУ «Информатика и системы управления» МГТУ им. Н.Э. Баумана на 5-й семестр, утвержденный учёным советом МГТУ им Н.Э. Баумана.

## 3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основное назначение ФМ заключается в предоставлении пользователям функционала для постобработки фотографий и для отбора удачных фотографий из серии загруженных. ФМ ориентировано на начинающих и любительских фотографов, у которых возникает потребность в простом и интуитивно понятном приложении для обработки фотографий.

## 4 ТРЕБОВАНИЯ К ПРОГРАММЕ

### 4.1 Требования к функциональным характеристикам.

#### 4.1.1 Выполняемые функции

##### 4.1.1.1 Настройка баланса белого

А. Настройка температуры

В. Настройка оттенка

##### 4.1.1.2 Настройка тона

- A. Настройка экспозиции
  - B. Настройка контрастности
  - C. Настройка светлых областей
  - D. Настройка теней
  - E. Настройка белого
  - F. Настройка черного
- 4.1.1.3 Настройка присутствия
- A. Настройка зернистости
  - B. Настройка четкости
  - C. Настройка размытия
  - D. Настройка красочности
  - E. Настройка насыщенности
- 4.1.1.4 Импорт фотографий формате PNG/JPEG/TIFF
- 4.1.1.5 Экспорт фотографий в формате PNG/JPEG/TIFF
- 4.1.2 Исходные данные
- 4.1.2.1 Данные для редактирование фотографии:
- A. Фотография в формате PNG/JPEG/TIFF
  - B. Численные значения соответствующих настроек
- 4.1.2.2 Данные просмотра фотографий:
- A. Загруженная фотография в формате PNG/JPEG/TIFF
- 4.1.3 Результаты:
- 4.1.3.1 Измененные в соответствии с заданными настройками фотографии в формате PNG/JPEG/TIFF
- 4.2 Требования к надежности
- 4.2.1 Защита загруженных фотографий
  - 4.2.2 Защита от некорректного ввода данных
  - 4.2.3 Защита от некорректных действий пользователя
- 4.3 Условия эксплуатации
- 4.3.1 Условия эксплуатации в соответствие с СанПиН 2.2.2/2.4.1340-03.
  - 4.3.2 Обслуживание



- 4.3.2.1 Обновления приложения: Регулярный выпуск обновлений для исправления ошибок, улучшения производительности и добавления новых функций
- 4.3.2.2 Техническая поддержка: Предоставление пользователям возможности обращаться в службу технической поддержки
- 4.3.3 Обслуживающий персонал
  - 4.3.3.1 Разработчики: ответственны за разработку, тестирование и выпуск обновлений приложения
- 4.4 Требования к информационной и программной совместимости
  - 4.4.1 Программное обеспечение должно функционировать на персональных компьютерах
  - 4.4.2 Минимальная конфигурация технических средств:
    - 4.4.2.1 Процессор с тактовой частотой не менее 2 ГГц, с поддержкой SSE 4.2 и 64-разрядных ОС
    - 4.4.2.2 ОЗУ: 8 ГБ
    - 4.4.2.3 Внутренняя память: 10 ГБ
    - 4.4.2.4 2 ГБ VRAM
- 4.5 Требования к информационной и программной совместимости
  - 4.5.1 Приложение должно быть совместимо с персональными компьютерами под управлением операционных систем Windows (версия 10 и выше), Linux и MacOS
  - 4.5.2 Входные данные должны быть представлены в следующих форматах: PNG, JPEG, TIFF, числа
  - 4.5.3 Результаты должны быть представлены в следующем формате: PNG, JPEG, TIFF
  - 4.5.4 Программное обеспечение должно иметь вид: рабочее окно состоит из двух секций. В левой секции отображается редактируемая фотография (изображение), в правой секции находятся ползунки с соответствующими настройками. Под ползунками в правой секции находится кнопка «Сброс», «Сохранить», «Сохранить как», «Открыть»,

«Применить ко всем». В левом верхнем углу находится выпадающее меню с кнопками «Сохранить как», «Сохранить» и «Открыть».

#### 4.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

#### 4.7 Требования к транспортировке и хранению

Требования к транспортировке и хранению не предъявляются.

#### 4.8 Специальные требования

Сгенерировать установочную версию компьютерного приложения.

### 5 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

5.2 Разрабатываемое программное обеспечение должно включать справочную систему.

5.3 В состав сопровождающей документации должны входить:

5.3.1 Расчетно-пояснительная записка на 20-30 листах формата А4 (без приложений 5.3.2, 5.3.3 и 5.3.4).

5.3.2 Техническое задание (Приложение А)

5.3.3 Руководство пользователя (Приложение Б).

5.3.4 Листинг программы (Приложение В).

5.4 Графическая часть должна быть включена в расчетно-пояснительную записку в качестве иллюстраций:

5.4.1 Концептуальная модель предметной области

5.4.2 Диаграмма вариантов использования

5.4.3 Примеры форм интерфейса

5.4.4 Граф состояний интерфейсов

5.4.5 Диаграмма пакетов

5.4.6 Диаграммы классов

5.4.7 Таблицы тестов

## 6 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Этап	Содержание этапа	Сроки и объем	Представляемые результаты	
			Спецификации и программный продукт	Документы
1.	Выбор темы, составление задания, решение организационных вопросов	1..2 недели (10 %)	-	Заполненный бланк задания на курсовую работу – вывешивается на сайт кафедры для получения утверждающей подписи заведующего кафедрой
2.	Анализ предметной области, разработка ТЗ. Исследование методов решения, выбор основных проектных решений	3..4 недели	Результаты декомпозиции предметной области. Эскизный проект: интерфейс, схемы	Фрагмент расчетно-пояснительной записки с обоснованием выбора средств и подходов к разработке
3.	Сдача ТЗ	4 неделя (25 %)	-	Техническое задание – утверждается руководителем
4.	Проектирование и реализация основных компонентов – интерфейса приложения и программ обработки данных.	5..7 недели	Технический проект основной части: структурная схема, схема процессов, схема алгоритмов модулей, граф состояний интерфейсов, формы интерфейса, схемы взаимодействия объектов, схемы взаимодействия модулей. Программный продукт, реализующий основные функции (демонстрируется руководителю)	Фрагмент расчетно-пояснительной записки с обоснованием разработанных спецификаций Тексты части программного продукта, реализующего основные функции.
5.	Сдача прототипа программного продукта	7 неделя (50 %)	Прототип программного продукта – демонстрируется руководителю	
6.	Разработка компонентов, обеспечивающих функциональную полноту	8..10	Рабочий проект программного комплекса. Готовый программный комплекс	Черновик расчетно-пояснительной записки. Тексты программного продукта.

Этап	Содержание этапа	Сроки и объем	Представляемые результаты	
			Спецификации и программный продукт	Документы
7.	Сдача программного продукта: компьютерное приложение и обработка фотографий	11 неделя (75 %)	Готовый программный комплекс – оценивается руководителем в баллах	-
8.	Тестирование программы и подготовка документации	12..14	Тесты и результаты тестирования.	РПЗ и Руководство пользователя.
9.	Оформление и сдача документации	14 неделя (90 %)	–	Расчетно-пояснительная записка и Руководство пользователя – проверяются и подписываются руководителем
10.	Защита курсовой работы	15..16 недели (100%)	–	Доклад (3-5 минут). Защита курсовой работы. Подписанная документация – вывешивается на сайт кафедры

## 7 ПОРЯДКИ КОНТРОЛЯ И ПРИЕМА

### 7.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

### 7.2 Порядок защиты

Защита осуществляется перед комиссией преподавателей.

### 7.3 Срок защиты

Срок защиты – 16-ая неделя.

## 8 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

**ПРИЛОЖЕНИЕ Б**  
**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Листов 7



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_

КАФЕДРА \_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_09.03.01 Информатика и вычислительная техника

ПРИЛОЖЕНИЕ ДЛЯ РЕДАКТИРОВАНИЯ ФОТОГРАФИЙ  
“ФОТОМАСТЕР”

Руководство пользователя

Листов 7

Студент ИУ6-53Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Дорохов Р.В.  
(И.О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Москва 2024 г.

## **1 Общие сведения о программном продукте**

Настоящий документ был сформирован специально для программного продукта редактирования изображений.

Доступные пользователю функции приложения:

- изменение параметров изображения;
- изменение параметров пакета изображений;
- предпросмотр результата изменения параметров изображения;
- загрузка пользовательских изображений для редактирования;
- сохранение результатов изменения параметров изображений.

Пользователями приложения могут быть студентам, начинающим фотографам и любительским фотографам.

## **2 Требования к системе**

Минимальная конфигурация технических средств:

- Процессор с тактовой частотой не менее 2ГГц, с поддержкой SSE 4.2 и 64-разрядных ОС;
- объем ОЗУ: 8 Гб;
- Внутренняя память: 10 Гб.

## **3 Описание установки и запуска**

1. Установите JRE и JDK;
2. Откройте директорию с загруженным .jar-файлом;
3. Запустите приложение с помощью команды `java -jar photoeditor.jar`;



## 4 Инструкция по работе

После запуска приложения пользователя встречает окно, представленное на рисунке Б1.

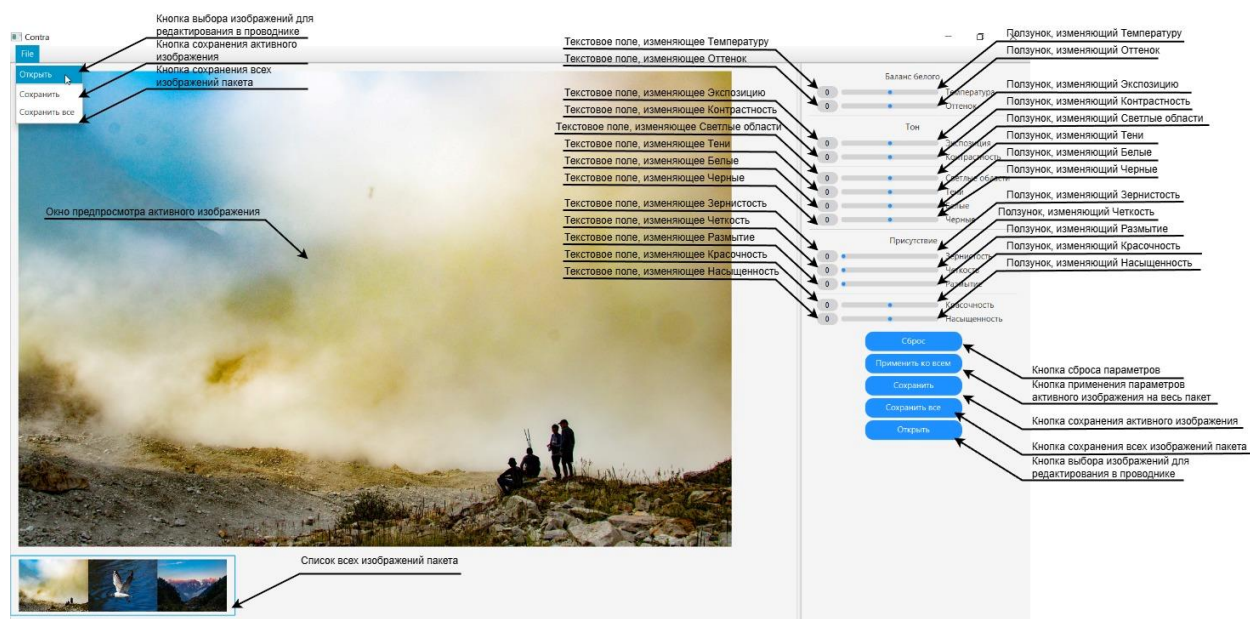


Рисунок Б1 – Начальное меню приложения

Приложение распознает следующие форматы изображений:

- JPEG файлы - \\*.jpeg, \\*.jpg, \\*.jpe;
- JPEG 2000 файлы - \\*.jp2
- Portable Network Graphics файлы - \\*.png
- TIFF файлы - \\*.tiff, \\*.tif

Чтобы изменить активное изображение нажмите левой кнопкой мыши на изображение из списка всех изображений пакета. Пример представлен на рисунке Б2.

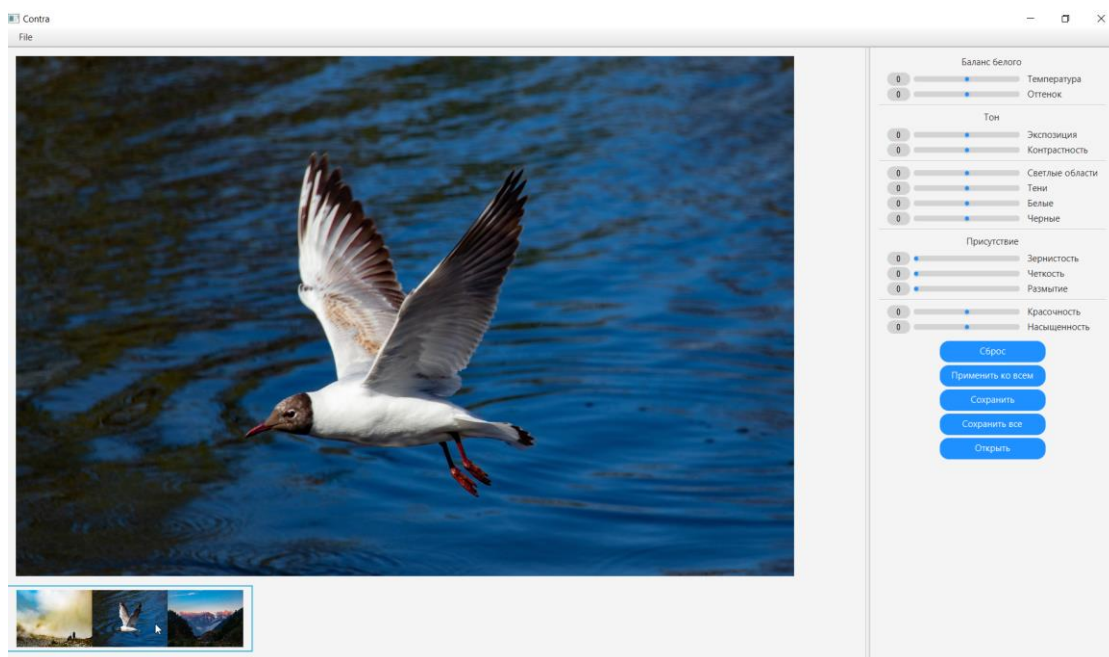


Рисунок Б2 – Результат нажатия левой кнопки мыши на фотографию чайки из списка всех изображений пакета

Чтобы выбрать изображения для редактирования нажмите левой кнопкой мыши на кнопку «Открыть» в основном окне приложения или кнопку «Открыть» в выпадающем меню в верхнем левом углу экрана. Откроется окно системного проводника, в котором вы можете выбрать изображения для редактирования. Результат нажатия на кнопку «Открыть» представлен на рисунке Б3.

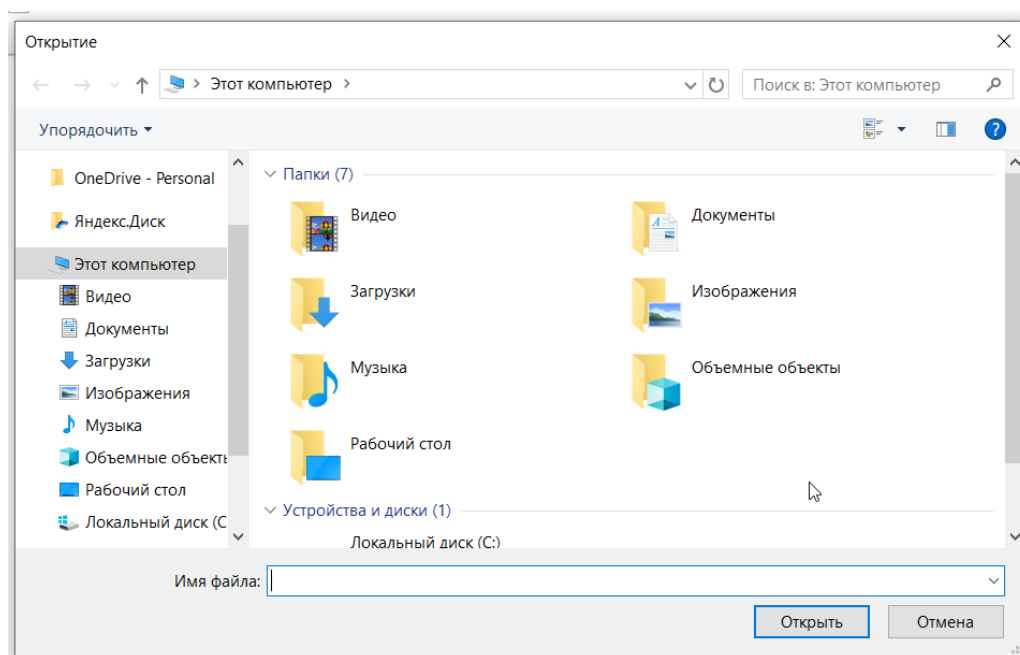


Рисунок Б3 – Окно проводника

Если вы выберете изображение с неподдерживаемым форматом, то приложение сообщит вам об этом во всплывающем окне. Пример всплывающего окна представлен на рисунке Б4.

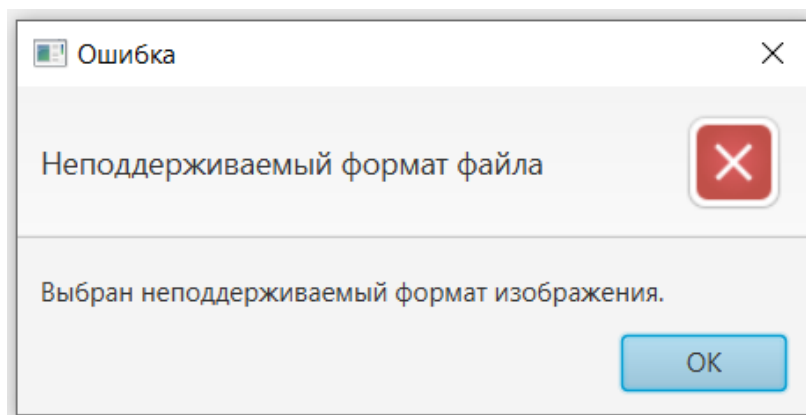


Рисунок Б4 – Пример сообщение о некорректном формате изображения

Если вы хотите изменить параметры активного изображения, введите новое значение необходимого вам параметра в соответствующее текстовое поле или измените значение с помощью ползунка.

Для параметров «Зернистость», «Четкость», «Размытие» диапазон значений лежит в пределах от 0 до 200. Для остальных параметров – от -100 до 100.

Пример представлен на рисунке Б5.

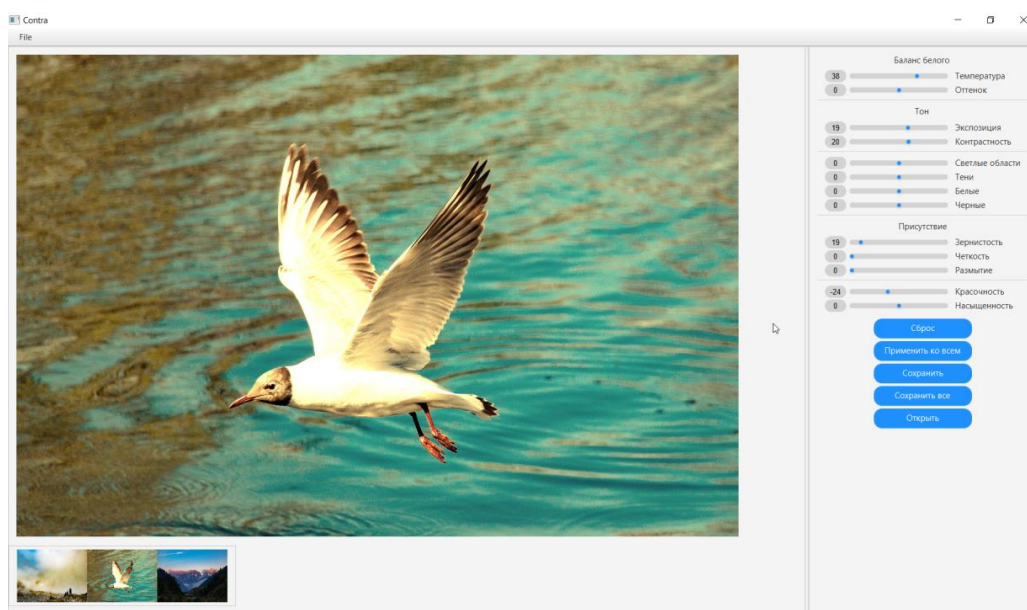


Рисунок Б5 – Изменение параметров «Температура», «Экспозиция», «Контрастность», «Зернистость», «Красочность» изображения

Если вы введете некорректное значение, то программа скажет вам об этом в всплывающем окне и установит изменяемое значение в положение 0. Пример всплывающего окна представлен на рисунке Б6.

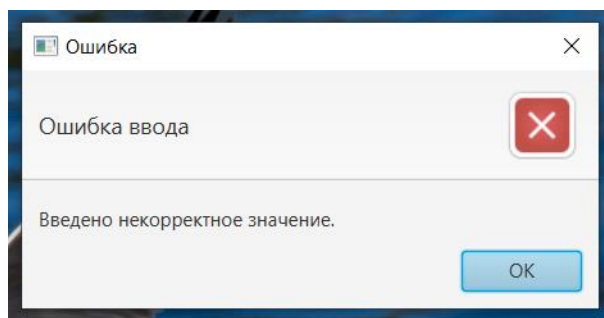


Рисунок Б6 – Пример сообщения о некорректном вводе

Чтобы применить изменения ко всем изображениям в пакете нажмите левой кнопкой мыши на кнопку «Применить ко всем». Пример представлен на рисунке Б7.

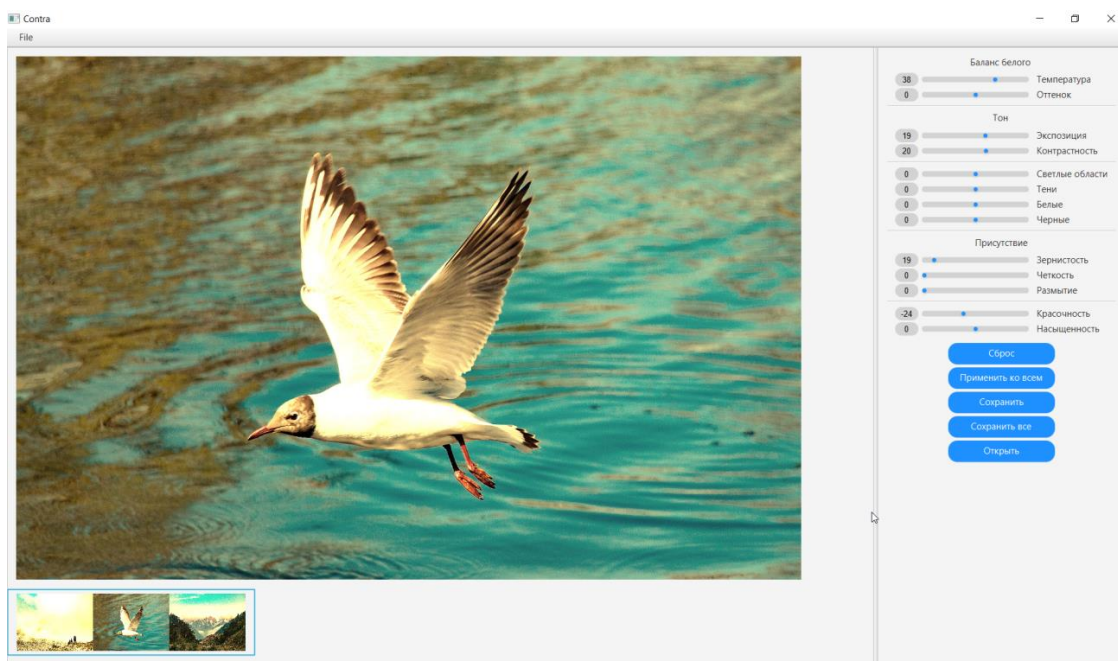


Рисунок Б7 – Применение параметров ко всем изображениям в пакете

Вы можете сбросить примененные параметры к начальным значениям 0, если нажмете на кнопку «Сброс» левой кнопкой мыши. Пример представлен на рисунке Б8.



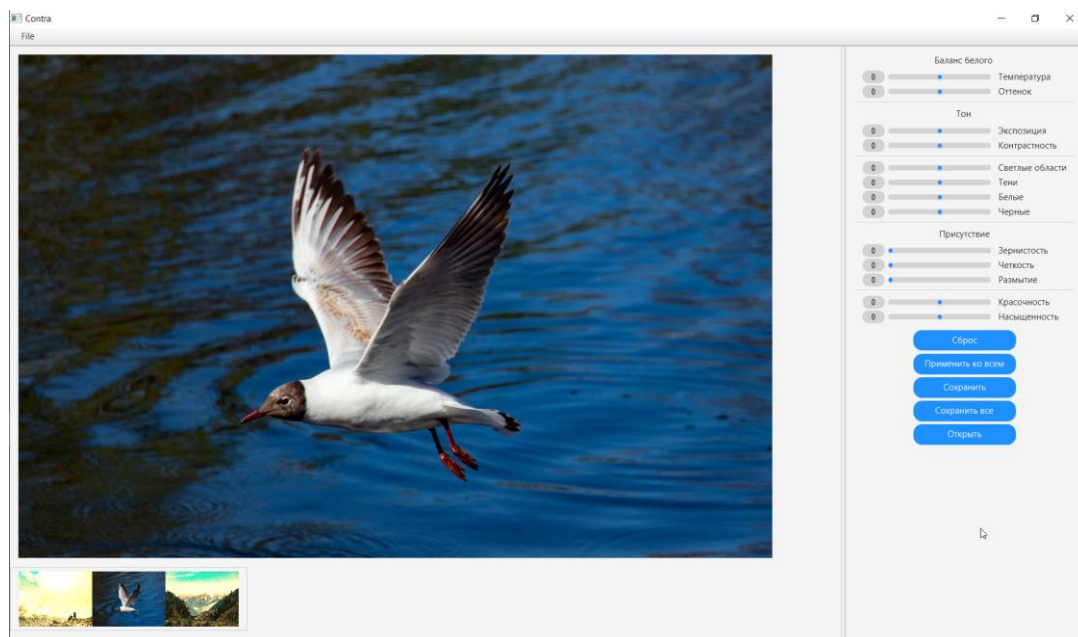


Рисунок Б8 – Результат нажатия кнопки «Сброс»

Вы можете получить краткое словесное описание параметра, если наведетесь на него мышкой. Пример представлен на рисунке Б9.

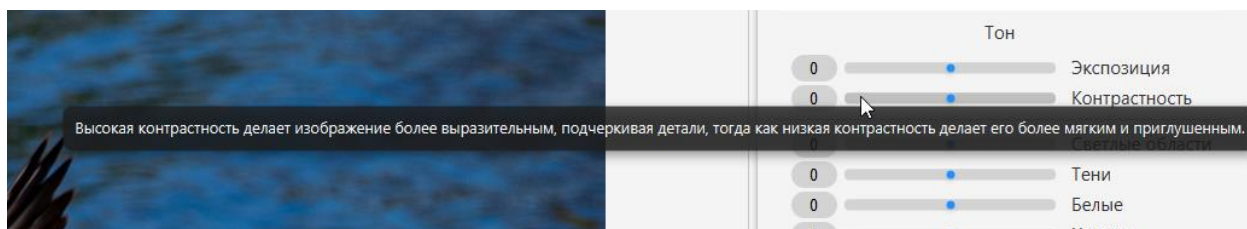


Рисунок Б9 – Описание параметра «Контрастность»

Если вы хотите сохранить активное изображение, то нажмите левой кнопкой мыши на кнопку «Сохранить» в основном окне приложения или на кнопку «Сохранить» в выпадающем меню в левом верхнем углу экрана.

Если вы хотите сохранить все изображения в пакете, то нажмите левой кнопкой мыши на кнопку «Сохранить все» в основном окне приложения или на кнопку «Сохранить все» в выпадающем меню в левом верхнем углу экрана.

Откроется окно проводника, в котором вы можете выбрать директорию для сохранения. Результат нажатия на кнопку «Сохранить» представлен на рисунке Б10.

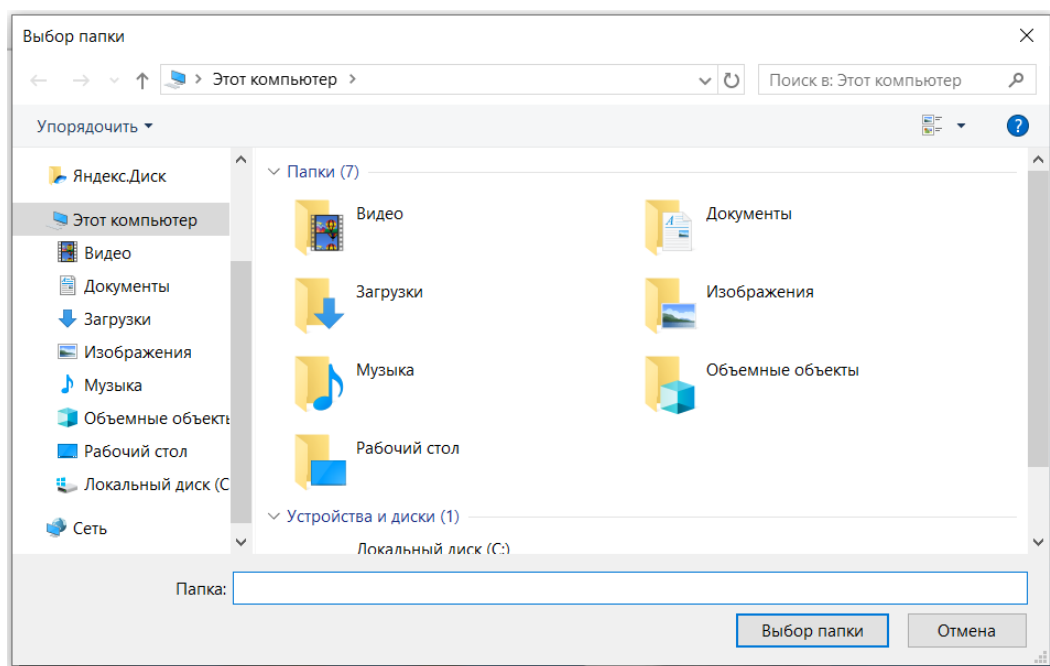


Рисунок Б10 – Выбор директории для сохранения изображений

**ПРИЛОЖЕНИЕ В**  
**ФРАГМЕНТ ИСХОДНОГО КОДА ПРОГРАММЫ**

Листов 5



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ \_ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ\_\_\_\_\_

КАФЕДРА \_КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ\_\_\_\_\_

НАПРАВЛЕНИЕ ПОДГОТОВКИ \_\_09.03.01 Информатика и вычислительная техника

ПРИЛОЖЕНИЕ ДЛЯ РЕДАКТИРОВАНИЯ ФОТОГРАФИЙ  
“ФОТОМАСТЕР”

Фрагмент исходного текста программы

Листов 5

Студент ИУ6И-51Б  
(Группа)

\_\_\_\_\_  
(Подпись, дата)

Дорохов Р.В.  
(И.О. Фамилия)

Руководитель курсовой работы

\_\_\_\_\_  
(Подпись, дата)

Москва 2024 г.



В качестве примера приведен текст класса MainController, описывающего логику приложения.

```
public class MainController {
    @FXML private ImageView imageView;

    private List<EditingImage> editingImages = new ArrayList<>();
    private EditingImage curImage;

    static {
        System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
    }

    private static String SLIDER_STYLE_CLASS = "slider";
    private static String TEXT_FIELD_STYLE_CLASS = "text-input text-field";

    @FXML public void onSliderDrag(Event event) {
        TextSliderConnect.sliderDrag(event, imageView);
    }

    @FXML public void onTextFieldEdit(Event event) {
        TextSliderConnect.textFieldEdit(event, imageView);
    }

    @FXML public void changeSetting(Event event) {
        double coef = 0;

        String id = Helper.getId(event);
        final Node source = (Node) event.getSource();
        String sourceClass = source.getStyleClass().toString();

        if (sourceClass.equals(SLIDER_STYLE_CLASS)) {
            Slider slider = (Slider) event.getSource();
            coef = Math.round(slider.getValue());
        } else if (sourceClass.equals(TEXT_FIELD_STYLE_CLASS)) {
            TextField textField = (TextField) event.getSource();
            coef = Double.parseDouble(textField.getText());
            if (!Helper.isValueCorrect(coef, event)) {
                coef = 0;
            }
        }

        changeImage(curImage, id, coef);
    }
}
```

```

@FXML public void changeAllImages() {
    for (EditingImage image : editingImages) {
        if (image != curImage) {
            image.changeSettings(curImage.settingsMap);
            image.changeImage();
        }
    }
}

private void changeImage(EditingImage editingImage, String settingName,
double coef) {
    editingImage.changeImage(settingName, coef);

    String imageEditCopy = editingImage.editingCopy;
    File file = new File(imageEditCopy);
    Image img = new Image(file.toURI().toString());
    imageView.setImage(img);
}

private void addImagesToList(List<File> files) {
    Scene scene = imageView.getScene();
    HBox imagesList = (HBox) scene.lookup("#imagesList");

    if (files != null) {
        clear();

        for (File file : files) {
            Image img = new Image(file.toURI().toString());
            ImageView imageView = new ImageView(img);
            imageView.setFitWidth(100);
            imageView.setFitHeight(75);
            imageView.setOnMouseClicked(this::setCurImage);
            imagesList.getChildren().add(imageView);

            String curFileName = Helper.fileToString(file);

            EditingImage image = null;
            try {
                image = new EditingImage(curFileName, imageView);
            } catch (Exception e) {
                clear();
                Alert alert = new Alert(Alert.AlertType.ERROR);
                alert.setTitle("Ошибка");
            }
        }
    }
}

```

```

        alert.setHeaderText("Неподдерживаемый формат файла");
        alert.setContentText("Выбран неподдерживаемый формат
изображения.");
        alert.showAndWait();
    }

    editingImages.add(image);
}

Image img = new Image(files.getFirst().toURI().toString());
curImage = editingImages.getFirst();

imageView.setImage(img);
imageView.setFitWidth(1030);
}
}

@FXML public void openFiles() throws IOException {
    FileChooser fileChooser = new FileChooser();
    List<File> files =
fileChooser.showOpenMultipleDialog(imageView.getScene().getWindow());

    addImagesToList(files);
}

@FXML public void saveFile() {
    saveFilesToDirectory(List.of(curImage.editingCopy));
}

@FXML public void saveAllFiles() {
    List<String> editingCopyImages = new ArrayList<>();
    editingImages.forEach(editingImage ->
        editingCopyImages.add(editingImage.editingCopy));
    saveFilesToDirectory(editingCopyImages);
}

@FXML public void saveFilesToDirectory(List<String> files) {
    File directory = getDirectory();

    for (String fileName : files) {
        String saveTo = Helper.fileToString(directory)
            + "/" + Helper.getImgName(fileName);
        Mat curImage = imread(fileName);
        imwrite(saveTo, curImage);
    }
}

```

```

    }
}

@FXML public void resetSettings() {
    curImage.resetSettings();

    changeImage(curImage, "#expSlider", 1); // можно подставить любую
    другую настройку вместо expose

    Scene scene = imageView.getScene();
    for (String id : curImage.settingsMap.keySet()) {
        if (id.endsWith("Slider")) {
            Slider slider = (Slider) scene.lookup(id);
            slider.setValue(0);
        } else if (id.endsWith("TextField")) {
            TextField textField = (TextField) scene.lookup(id);
            textField.setText("0");
        }
    }
}

private void setCurImage(MouseEvent event) {
    ImageView newImageView = (ImageView) event.getSource();
    EditingImage editingImage = findEditingImage(newImageView);

    curImage = editingImage;

    setImageToImageView(editingImage);
}

private void setImageToImageView(EditingImage editingImage) {
    File file = new File(editingImage.editingCopy);
    Image img = new Image(file.toURI().toString());
    imageView.setImage(img);
}

private File getDirectory() {
    DirectoryChooser directoryChooser = new DirectoryChooser();
    Scene scene = imageView.getScene();
    return directoryChooser.showDialog(scene.getWindow());
}

private void clear() {
    editingImages.clear();
}

```

```
Scene scene = imageView.getScene();
HBox imagesList = (HBox) scene.lookup("#imagesList");
imagesList.getChildren().clear();
}

private EditingImage findEditingImage(ImageView imageView) {
    for (EditingImage image : editingImages) {
        if (image.imageView == imageView) {
            return image;
        }
    }

    return null;
}
}
```

Полный текст проекта размещен по ссылке:  
<https://github.com/rvdorokhov/PhotoRedactor>.