|  |
| --- |
| **МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**  **БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**  **МЕХАНИКО МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**  **Кафедра веб-технологий и компьютерного моделирования**  **РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ ДЛЯ ОБРАБОТКИ ИЗОБРАЖЕНИЙ**  Курсовая работа  Евсей Ирины Сергеевны  студентки 3 курса, специальность «математика и информационные технологии (направление математическое и программное обеспечение мобильных устройств)»  Научный руководитель:  А. С. Девицын  Минск, 2018 |

**ОГЛАВЛЕНИЕ**

Введение………………………………………....…………………………………...3

Цели проекта………………………………………………………………………....4

Краткое описание приложения……………………………………………………...5

Описание используемых технологий…………………………………………….....6

Описание архитектуры приложения……………………………...…………….....10

Дизайн……………………………………………………………………………….13

Тестирование приложения…………………………………………………………14

Заключение………………………………………………………………………….21

Приложения……………………………………………………………………........22

Список использованных источников……………………………………………...30

**ВВЕДЕНИЕ**

Сегодняшний день является временем научно-технического прогресса, очень сложно представить себе жизнь современного общества без использования мобильных устройств. Мы пользуемся ими повсюду — дома, в поездке, на работе, на отдыхе. Очевидно, что мобильный телефон уже давно перестал быть просто средством общения. Обыкновенный разговор по телефону постепенно становится второстепенной функцией, пропадая в огромном наборе функций, реализуемых мобильным телефоном. Слушать музыку, фотографировать, играть... Этот список можно продолжать бесконечно. Таким образом, мобильный телефон стал многофункциональным устройством, позволяющим человеку пользоваться практически всеми современными технологиями. Изменилось и отношение людей к мобильным телефонам. Сегодня, забыв сотовый дома, люди ощущают себя не комфортно. Хотя, всего несколько лет назад они спокойно обходились без мобильного телефона и считали его атрибутом роскоши. Их развитие идет колоссальными темпами. Если в начале телефоны старались сделать компактными и дешевыми, то сегодня мобильные телефоны делают универсальным устройством, которое может заменить диктофон, фотоаппарат, и даже компьютер. Эта достигается благодаря новым возможностям операционных систем, которые развиваются вместе с самими аппаратными возможностями. Успех мобильного устройства еще и в большей степени зависит от того, какая на нем установлена операционная система. Но несомненным лидером по популярности, на сегодняшний день выступает, несомненно ОС Android. Она нацелена на массовую аудиторию и является весьма доступной и удобной программной разработкой. Секрет ее успеха очень прост - она открыта и бесплатна, предоставляет программисту удобные средства разработки и возможность реализации своих продуктов, число которых на сегодняшний день насчитывает несколько сотен тысяч.

Цель курсовой работы – разработать Android-приложение для редактирования/обработки изображений.

Входе выполнения работы предстоит выполнить следующие задачи:

•Обзор и изучение операционной системы android;

•Выбор средства разработки и тестирования программного продукта для ОС Android;

•Разработка приложения и его тестирование для ОС Android.

**ЦЕЛИ ПРОЕКТА**

Перед написанием данной работы были поставлены цели:

* изучить и углубить знания в технологиях программирования под ОС Android, разобрать внутреннее состояние и жизненный цикл приложений, запускаемых на Android
* изучить новые паттерны проектирования приложений
* спроектировать и разработать качественный пользовательский опыт и интерфейс, используя современные графические библиотеки и инструменты
* разработать высококачественное и практичное приложение для ОС Android
* изучить работу с библиотеками и применить их в проекте
* проанализировать, изучить и применить алгоритмы для работы с изображениями

**КРАТКОЕ ОПИСАНИЕ ПРИЛОЖЕНИЯ**

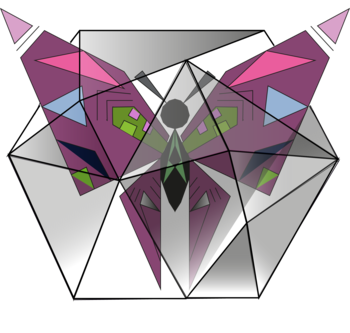


Рисунок 1. Логотип приложения

Результатом работы является мобильное приложение, которое позволяет работать с изображениями. С помощью него можно проделывать следующие манипуляции: для начала, загрузить готовое изображение из галереи телефона или же сделать моментальное фото; пропорциональное уменьшение/увеличение изображения; наложение графических фильтров: Greyscale, Histogram, Negative, Sepia, Sketch; матричные фильтры: Laplacian и Sobel; регулирование яркости и контраста изображения; отмена действий; сохранение изображения в память телефона с нужным именем.

Также приложение имеет современный пользовательский интерфейс.

В дальнейшем, я думаю, будут расширены возможности данного приложения.

**ОПИСАНИЕ ИСПОЛЬЗУЕМЫХ ТЕХНОЛОГИЙ**

В ходе работа мною были использованы:

1. **Android Studio**

Android Studio — это интегрированная среда разработки (IDE) для работы с платформой Android, анонсированная 16 мая 2013 года на конференции Google I/O.

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, - официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, OS X и Linux. Официальнаями языками программирования для платформы Android являются Java, С++ и Kotlin.

Особенности Android Studio:

Новые функции появляются с каждой новой версией Android Studio. На данный момент доступны следующие функции:

* + Расширенный редактор макетов: WYSIWYG, способность работать с UI компонентами при помощи Drag-and-Drop, функция предпросмотра макета на нескольких конфигурациях экрана;
  + Сборка приложений, основанная на Gradle;
  + Различные виды сборок и генерация нескольких .apk файлов;
  + Рефакторинг кода;
  + Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое;
  + Встроенный ProGuard и утилита для подписывания приложений;
  + Шаблоны основных макетов и компонентов Android;
  + Поддержка разработки приложений для Android Wear и Android TV;
  + Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine;

1. **Библиотека OpenCV**

OpenCV – (Open Computer Vision) — библиотека компьютерного зрения с открытым исходным кодом, предоставляющая набор типов данных и численных алгоритмов для обработки изображений алгоритмами компьютерного зрения.

OpenCV написана на языке высокого уровня (C/C++) и содержит алгоритмы для: интерпретации изображений, калибровки камеры по эталону, устранение оптических искажений, определение сходства, анализ перемещения объекта, определение формы объекта и слежение за объектом, 3D-реконструкция, сегментация объекта, распознавание жестов и т.д.

Эта библиотека очень популярна за счёт своей открытости и возможности бесплатно использовать как в учебных, так и коммерческих целях.

Фактически, OpenCV – это набор типов данных, функций и классов для обработки изображений алгоритмами компьютерного зрения.

Основные модули библиотеки:

* cxcore — ядро

Содержит базовые структуры данных и алгоритмы:

— базовые операции над многомерными числовыми массивами,

— матричная алгебра, математические ф-ции, генераторы случайных чисел,

— запись/восстановление структур данных в/из XML,

— базовые функции 2D графики;

* CV — модуль обработки изображений и компьютерного зрения

— базовые операции над изображениями (фильтрация, геометрические преобразования, преобразование цветовых пространств и т. д.),

— анализ изображений (выбор отличительных признаков, морфология, поиск контуров, гистограммы),

— анализ движения, слежение за объектами,

— обнаружение объектов, в частности лиц,

— калибровка камер, элементы восстановления пространственной структуры;

* Highgui — модуль для ввода/вывода изображений и видео, создания пользовательского интерфейса

— захват видео с камер и из видео файлов, чтение/запись статических изображений,

— функции для организации простого UI (все демо приложения используют HighGUI),

* Cvaux — экспериментальные и устаревшие функции

— пространств. зрение: стерео калибрация, само калибрация,

— поиск стерео-соответствия, клики в графах,

— нахождение и описание черт лица;

* CvCam — захват видео

— позволяет осуществлять захват видео с цифровых видео-камер ( поддержка прекращена и в последних версиях этот модуль отсутствует ).

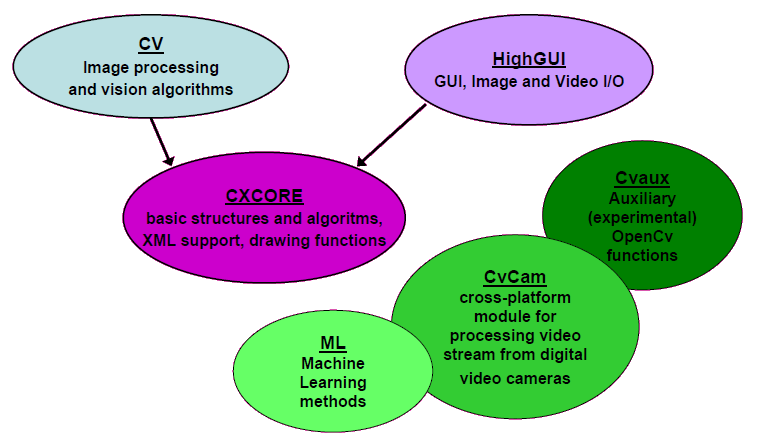


Рисунок 2.

1. **Java**

Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

Язык Java активно используется для создания мобильных приложений под операционную систему Android. При этом программы компилируются в нестандартный байт-код, для использования их виртуальной машиной Dalvik (начиная с Android 5.0 Lollipop виртуальная машина заменена на ART). Для такой компиляции используется дополнительный инструмент, а именно Android SDK (Software Development Kit), разработанный компанией Google.

Разработку приложений можно вести в среде Android Studio, NetBeans, в среде Eclipse, используя при этом плагин Android Development Tools (ADT), или в IntelliJ IDEA. Версия JDK при этом должна быть 5.0 или выше.

8 декабря 2014 года Android Studio признана компанией Google официальной средой разработки под ОС Android.

**ОПИСАНИЕ АРХИТЕКТУРЫ ПРИЛОЖЕНИЯ**

Архитектура приложения:

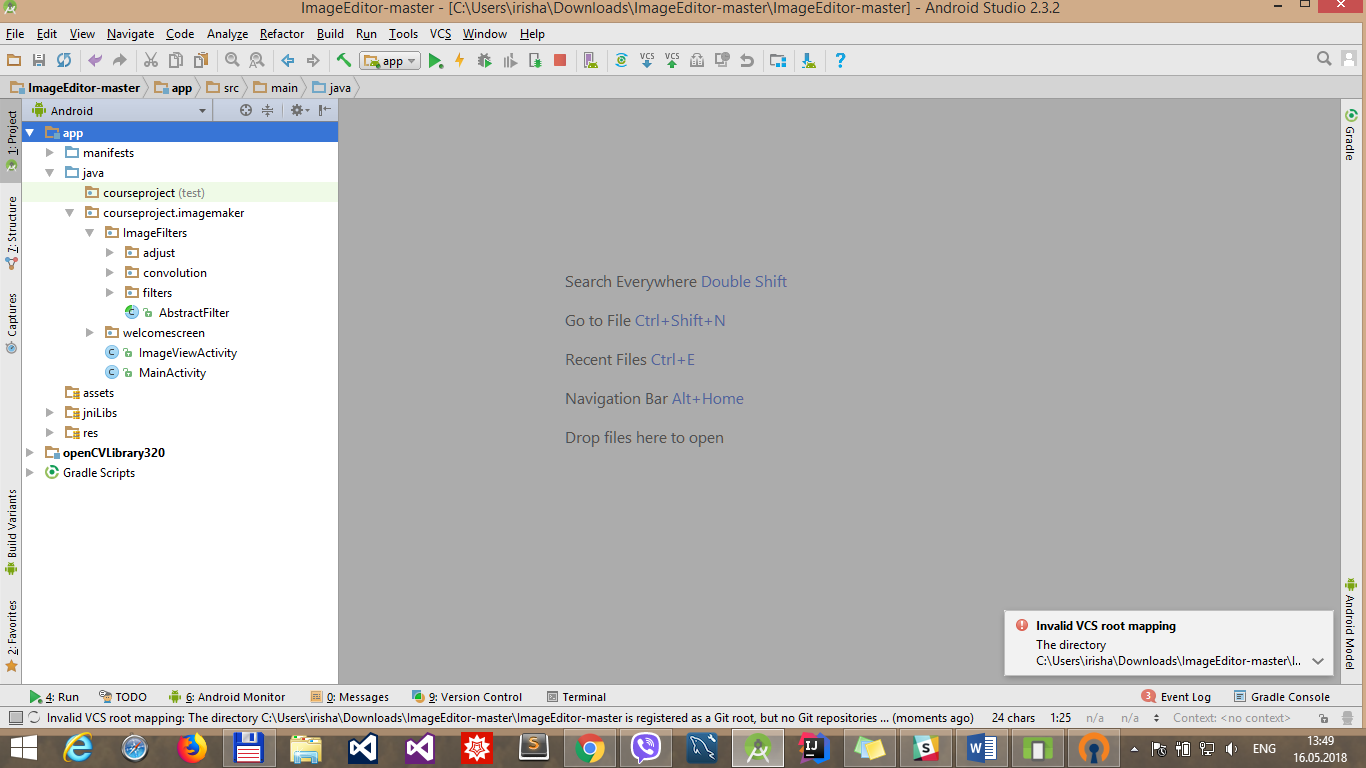


Рисунок 3. Архитектура приложения

Внедрена загрузка изображения из галереи телефона, материализуемое кнопкой в ​​верхнем меню, представляющем галерею. Доступ к снимку непосредственно с камеры возможен с помощью кнопки камеры. После проверки фотография загружается вместо предыдущей фотографии. Чтобы удовлетворить эти потребности, используется Intent. Действительно, при нажатии этих кнопок приложение отправляет сообщения через класс Intent «Я хочу получить доступ к галерее» или «Я хочу сделать снимок», а затем ответ на этот запрос, будет предложен пользователю.

Для работы с внутренней памятью и камерой необходимо в файле AndroidManifest.iml указать следующее:

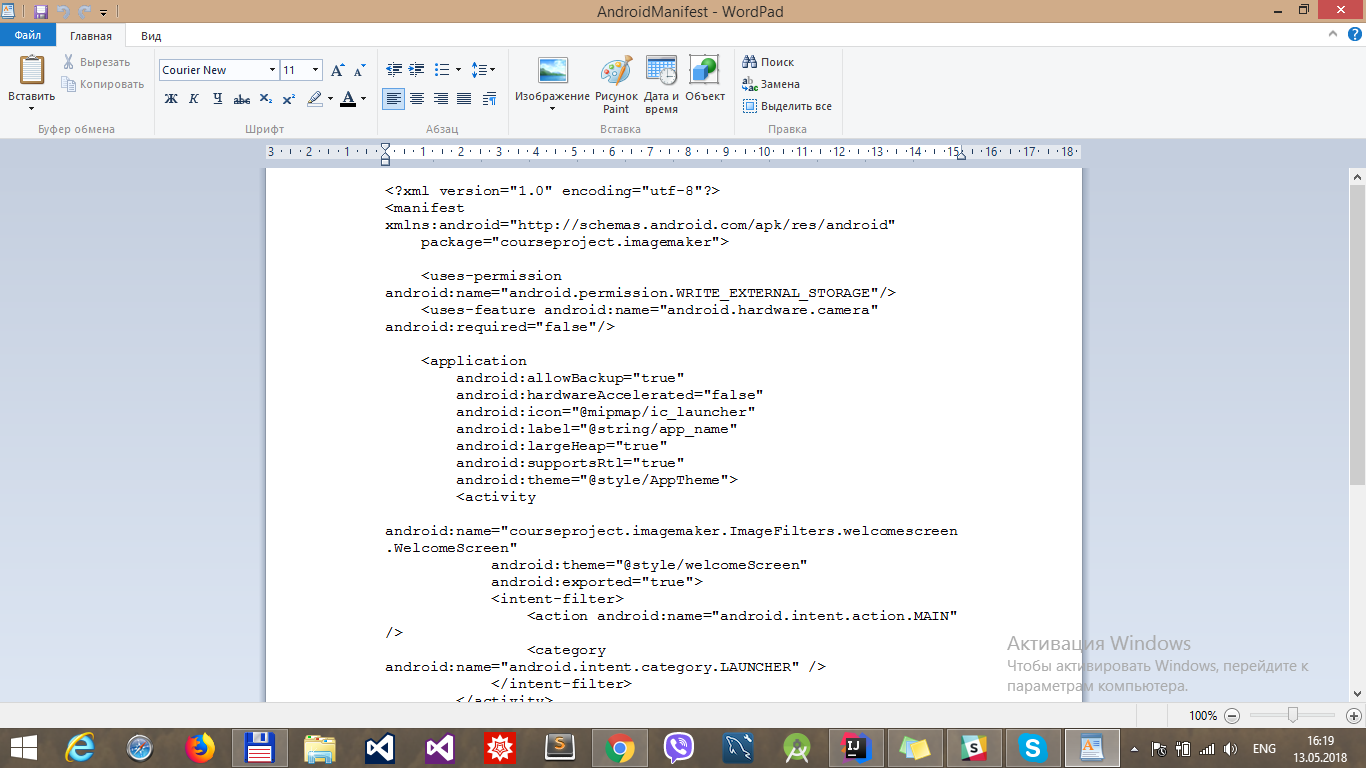


Рисунок 4. Работа с внутренней памятью и камерой

Когда пользователь применяет изменения к выбранному им изображению, предоставляется возможность отменить их для восстановления предыдущего изображения или непосредственно исходного изображения. Эта функция сброса фильтра реализована с помощью кнопки, расположенной в верхнем левом углу (она должна быть нажата пользователем преднамеренно), в виде стрелки «возврат». При простом клике отменяется последнее изменение и так далее. Во время долгого клика, которое намеренно делается, пользователь получает с нуля свое исходное изображение без каких-либо изменений.

Когда пользователь удовлетворен своим измененным изображением, чтобы сохранить его на своем устройстве он должен нажать значок в правом верхнем углу. После указания имени изображение сохраняется во внутренней памяти устройства в папке «ХРАНЕНИЕ> Внутреннее хранилище> ImageMaker».

Пользователь имеет возможность масштабирования и скроллера на своем изображении, чтобы лучше визуализировать изменения. Для скроллера, на изображение должен быть помещен только один палец, а затем перемещен. Чтобы увеличить масштаб необходимо два пальца, для выполнения «растяжение» или «уменьшение». Отмечу, что на данный момент пользователь может скроллить за пределы изображения. Планируется улучшить прокрутку.

Обработка изображений:

● В папке [ImageFilters -> adjust] находятся классы-модификаторы яркости Brightness и контрастности Contrast (можно варьировать эти параметры в соответствии с выбором  интенсивности пользователя, используя полосу прокрутки).

● В [ImageFilters -> filters] находятся классы-фильтры для работы с изменением цвета изображения : переход на черно-белый Greyscale, сепия Sepia, негатив Negative, изменение оттенка изображения на основе выбранного пользователем цвета ColorPicker, фильтр “Эскиз” Sketch.

● Также, в [ImageFilters -> adjust] присутствует класс-фильтр эквалайзера с гистограммой для оптимальной контрастности изображения с низким контрастом.

● В [ImageFilters -> convolution] классы-фильтры для обнаружения и выделения краев изображения с использованием таких алгоритмов, как Sobel или Laplacien.

В приложении используется OpenCV — библиотека алгоритмов обработки изображений, использование не требует подключения к Интернету.

Обработка изображений - это процесс, который требует высокого использования RAM, поскольку выполняются тяжелые операции.

Чтобы впоследствии сохранить эволюционирующий код и избежать избыточности, я реализовала абстрактный класс AbstractFilter, который служит основой для его производных классов. Его реализация не является полной и методы должны быть реализованы в дочерних классах.

Упрощенная диаграмма классов ниже объясняет архитектуру.

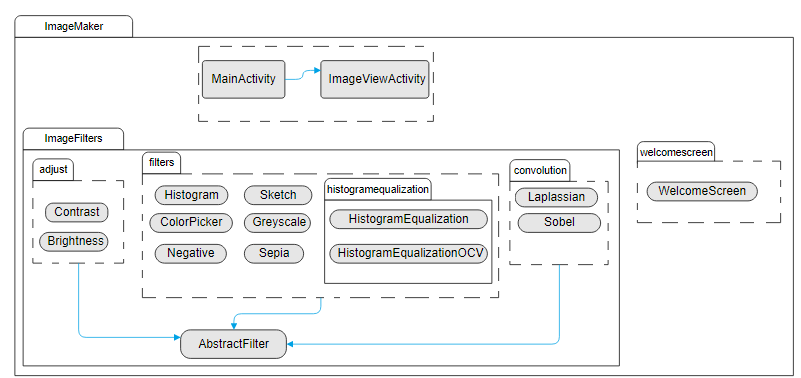


Рисунок 5. Диаграмма классов

Таким образом, для каждой модификации изображения создается новый класс. Маска абстрактного класса содержит исходное растровое изображение и общие методы, которые служат для нескольких модификаций. Если в будущем необходимо будет добавить новый вид редактирования, то просто создается новый класс, который реализует абстрактный класс.

**ДИЗАЙН**

Пользовательский интерфейс мобильного приложения должен быть интуитивно понятным, легко используемым для всех.

Я хотела, чтобы меню и значки были более минималистичными и оставила много места для изображения. Таким образом, были реализованы два прозрачных меню.

Также, я различаю функции, связанные с изменениями изображения в меню. В нижней части экрана расположены те, которые применяются для редактирования, а в верхней части экрана - сохранение или сброс изменений.

Различные изменения сортируются по трем категориям в нижнем меню: «Регулировка», «Фильтр» и «’Свертывание’». Можно заметить, что я выбрала интерфейс на английском языке для охвата более широкой аудитории.

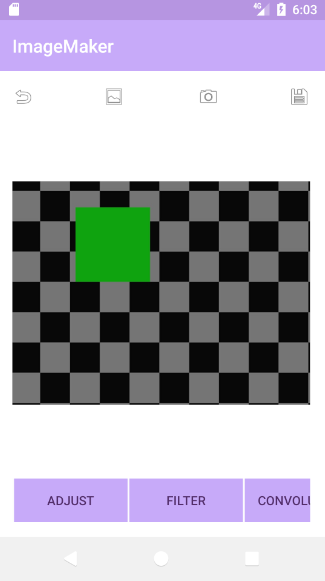


Рисунок 6. Дизайн приложения

**ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ**

Основные различия между мобильными и настольными приложениями:

* Мобильное устройство — это система, которая не обладает мощной начинкой. Таким образом, он не может работать как персональный компьютер.
* Тестирование мобильных приложений проводится на мобильных телефонах (Apple, Samsung, Nokia), в то время как настольное приложение тестируется на центральном процессоре.
* У мобильных устройств бывают разные разрешения. Размер экрана мобильного телефона меньше, чем у настольных.
* Выполнение и прием вызовов является основной задачей телефона, поэтому приложение не должно вмешиваться в эту важную функцию.
* Широкий спектр конкретных операционных систем и компонентных конфигураций: Android, iOS, BlackBerry.
* Операционная система мобильного телефона быстро устаревает.
* Мобильные устройства используют сетевые подключения (3G, 4G, Wi-Fi), широкополосное подключение к настольному ПК или Wi-Fi.
* Мобильные устройства постоянно осуществляют поиск сети. Вот почему нужно протестировать приложение с разной скоростью передачи данных.
* Инструменты, которые хорошо подходят для тестирования настольных приложений, не полностью подходят для тестирования мобильных приложений.
* Мобильные приложения должны поддерживать несколько входных каналов (клавиатура, голос, жесты и т. д.), мультимедийные технологии и другие функции, повышающие их удобство использования.

Для запуска и тестирования приложения можно использовать эмуляторы или реальные устройства. Но в идеале лучше тестировать на реальных устройствах.

Для использования мобильного устройства для тестирования на рабочую машину необходимо установить драйвер. Если смартфон от Google - Nexus 5/6/5x/6P или Google Pixel, то для его поддержки необходимо через SDK Manager установить пакет Google Usb Driver. Если же производитель аппарата - другой вендор, то надо установить то USB-драйвер, который поставляется данным вендором. Если ОС - Windows 10, то там, как правило, система сама может найти через центр обновлений драйвер и установить его.

По умолчанию опции разработчика на смартфонах скрыты. Чтобы сделать их доступными, надо зайти в Settings > About phone ( Настройки > О телефоне ) и семь раз нажать Build Number ( Номер сборки ).

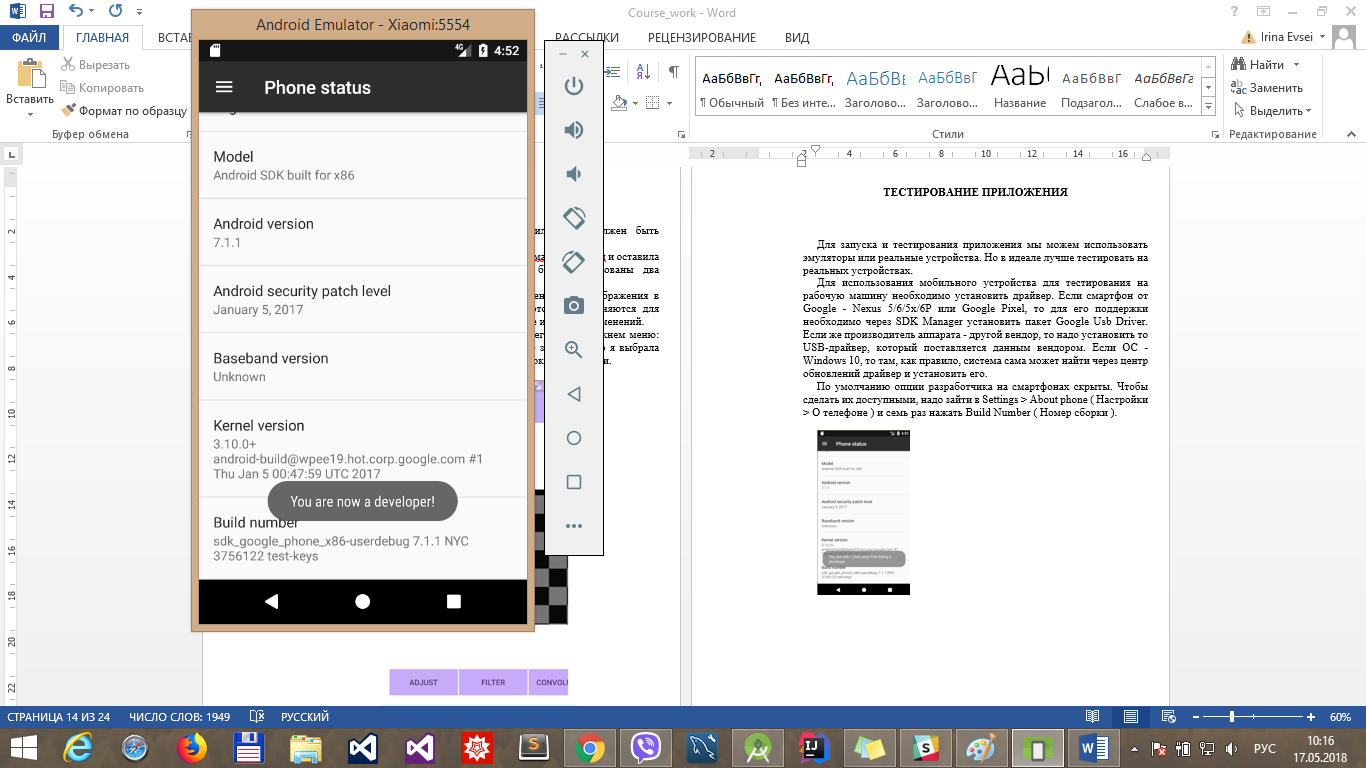


Рисунок 7.

Вернитесь к предыдущему экрану и там вы увидите доступный пункт Developer options (Для разработчика).

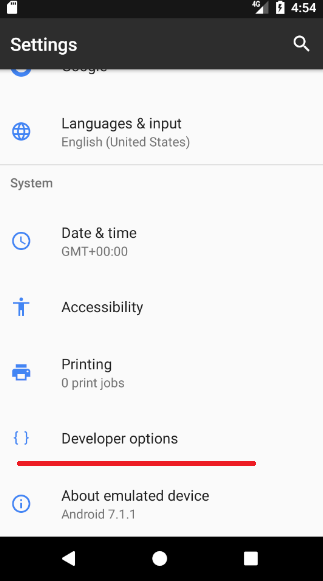
****

Рисунок 8.

Перейдем к пункту Для разработчиков и включим возможность отладки по USB:

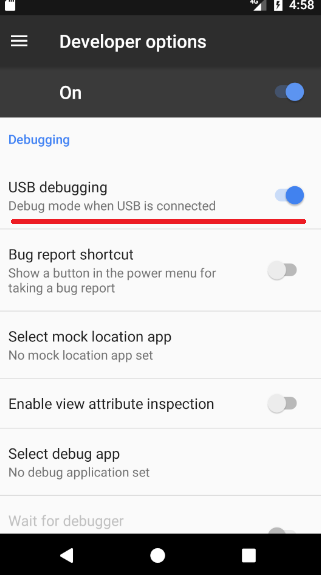


Рисунок 9.

Далее следует запуск приложения.

Подключим устройство с ОС Android (если мы тестируем на реальном устройстве) и запустим проект, нажав на зеленую стрелочку на панели инструментов.

Затем начнется построение проекта. Данный процесс может занять некоторое время, после чего отобразится диалоговое окно для выбора устройства для запуска. Здесь мы можем выбрать подключенный к компьютеру гаджет, либо эмулятор:

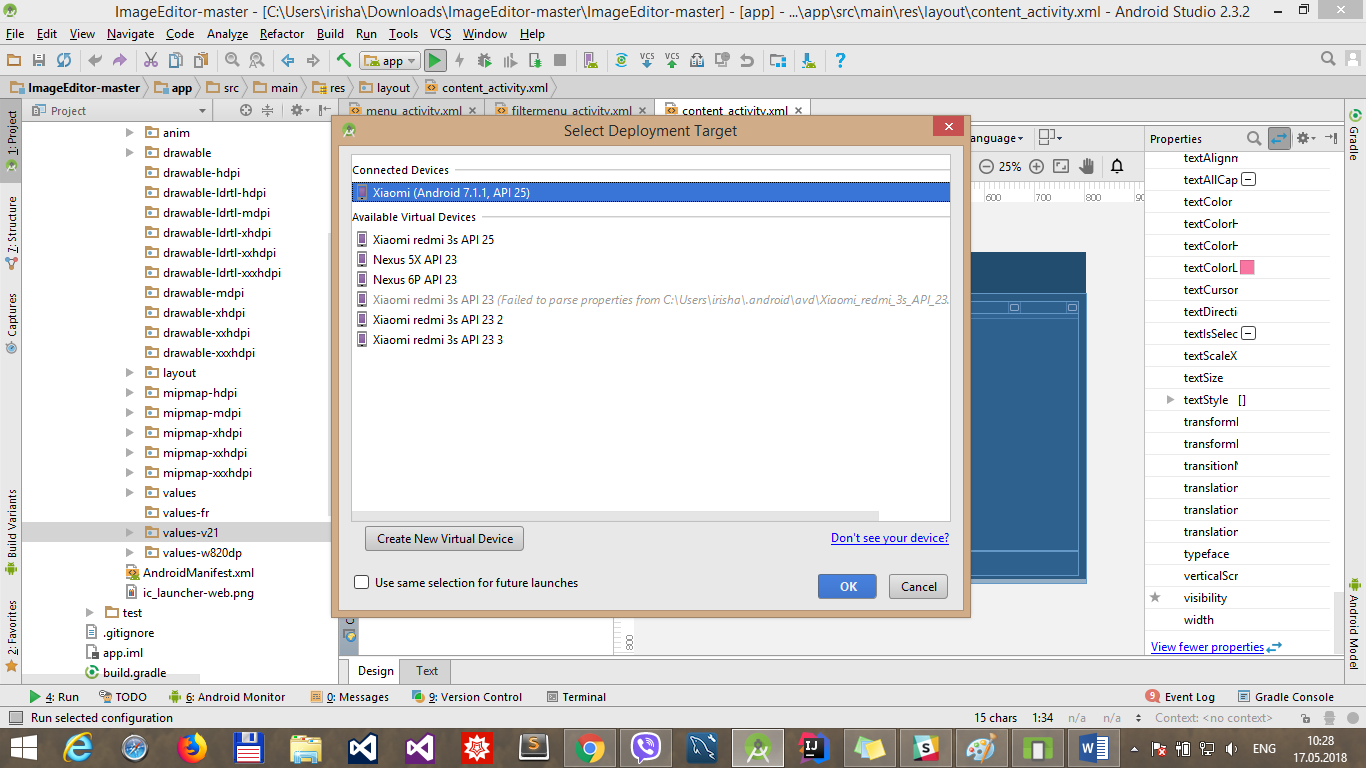


Рисунок 10.

Выберем устройство и нажмем на кнопку OK. И после запуска мы увидим наше приложение на экране устройства:



Рисунок 11.

Приложение прошло тестирование.

Основные моменты, на которые мне необходимо было обратить особое внимание именно при тестировании приложения на мобильных устройствах.

* Размер экрана и touch-интерфейс:

- Размеры всех элементов графического интерфейса пользователя.

- Проверка возможности использования всех активных элементов (кнопок, ссылок и т.д.).

- Скорость отклика активных элементов

- Проверка того, что многократное быстрое нажатие на кнопку не вызовет экстренное завершение приложения.

- Поддержка мультитача – одновременное нажатие нескольких кнопок.

- Проверка использования в приложении специальных жестов (double tap, swipe, pinch in/out и т.д.).

* Ресурсы телефона:

- Необходимо было проконтролировать возможные утечки памяти. Часто это случается в приложениях с окнами, содержащими большое количество информации. Также утечки памяти могут присутствовать во время длительной работы приложения.

- Проверка обработки ситуаций нехватки памяти для функционирования операционной системы, во время работы приложения в активном и фоновом режимах.

- Недостаток места для установки или работы приложения.

- Установка на SD-карту.

- Проверка работы батареи (аккумулятора) устройства при запущенном приложении, работе в фоновом режиме, при включенном Wi-Fi, 3G Интернете, без подключения к сети и т.д.

* Различные разрешения экрана и версии ОС:

- Необходимо было проверить работу приложения на устройствах с различными разрешениями экрана. На экранах с высоким разрешением (например, ретина-экран) элементы интерфейса и текст отображаются мельче, при работе приложения на устройстве с экраном более низкого разрешения элементы интерфейса могут стать слишком большими.

* Реакция приложения на внешние прерывания:

- Входящие и исходящие SMS и MMS.

- Входящие и исходящие звонки.

- Напоминание, будильник и т.д.

- Отключение и подключение Wi-Fi. Например, приложение может экстренно завершать свою работу при запуске в авиа режиме, при отключенном Wi-Fi. При потере сигнала и одновременном выполнении операции может отображаться бесконечная загрузка приложения, вместо корректного сообщения про отсутствии интернет-подключения;

- Отключение и подключение SD-карты.

- Отключение/подключение мобильного устройства к зарядному устройству.

- Работа в фоновом (background) режиме.

- Работа приложения после выхода из спящего режима.

- Совместимость с другими приложениями.

Пользователь мобильного устройства ожидает, что устанавливаемые ими приложения просты, интуитивно понятны, работают всегда и везде без сбоев. Если ожидания не оправдываются, то пользователь просто-напросто устанавливает аналогичное приложение от другого разработчика, которых в сфере мобильных разработок всегда достаточно. Поэтому качество приложения является одним из главных факторов его популярности.

**ЗАКЛЮЧЕНИЕ**

Целью данного проекта являлась разработка мобильного приложения для обработки изображений. Подводя итоги, можно сказать, что поставленная цель была успешно достигнута.

ImageMaker представляет собой мобильное приложение, позволяющее открыть или сделать моментальное изображение, выбрать из меню некоторые эффекты обработки и сохранить новое изображение в памяти устройства.

**ПРИЛОЖЕНИЯ**

**ПРИЛОЖЕНИЕ А**.

Приветственное окно приложения выглядит следующим образом:

Рисунок 12. Приветственное окно приложения

**ПРИЛОЖЕНИЕ B**.

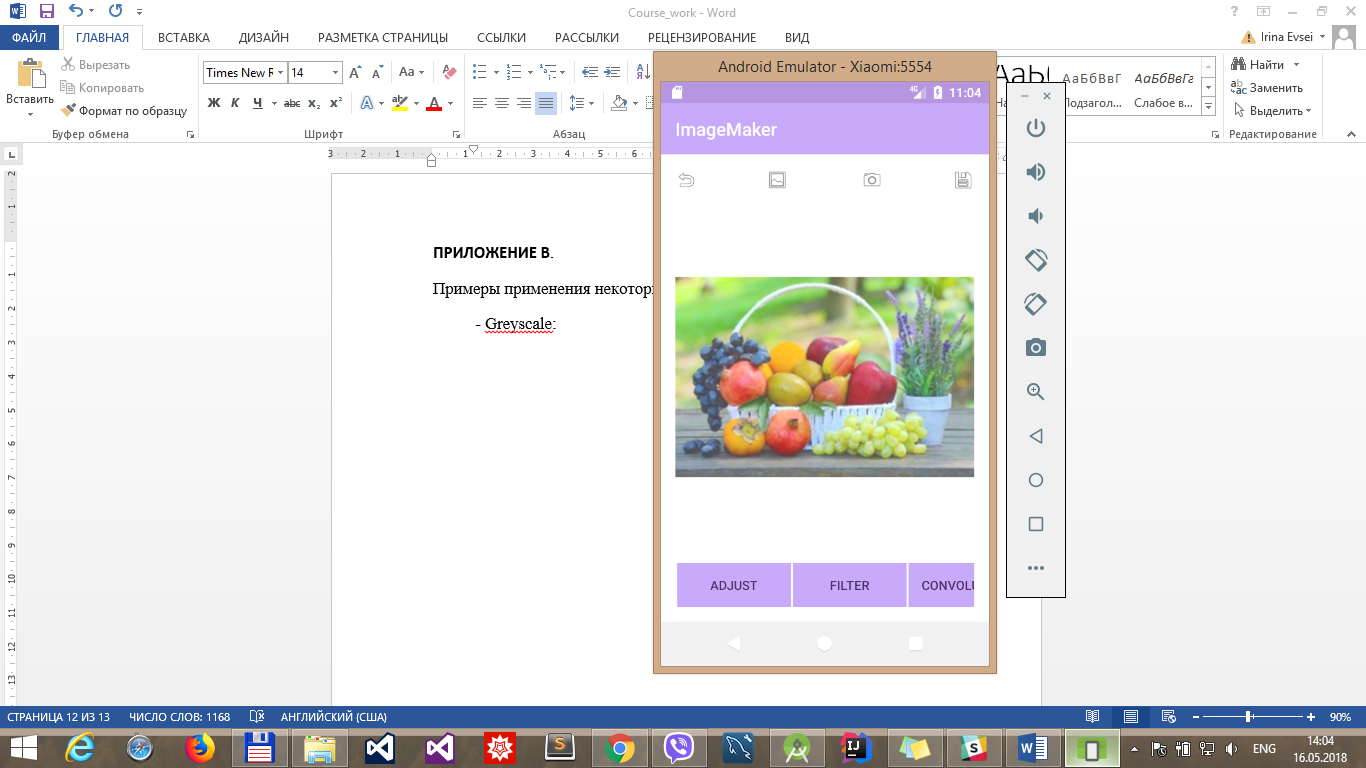
Примеры применения некоторых изменений к изображению (до/после):

Рисунок 13. Исходное изображение

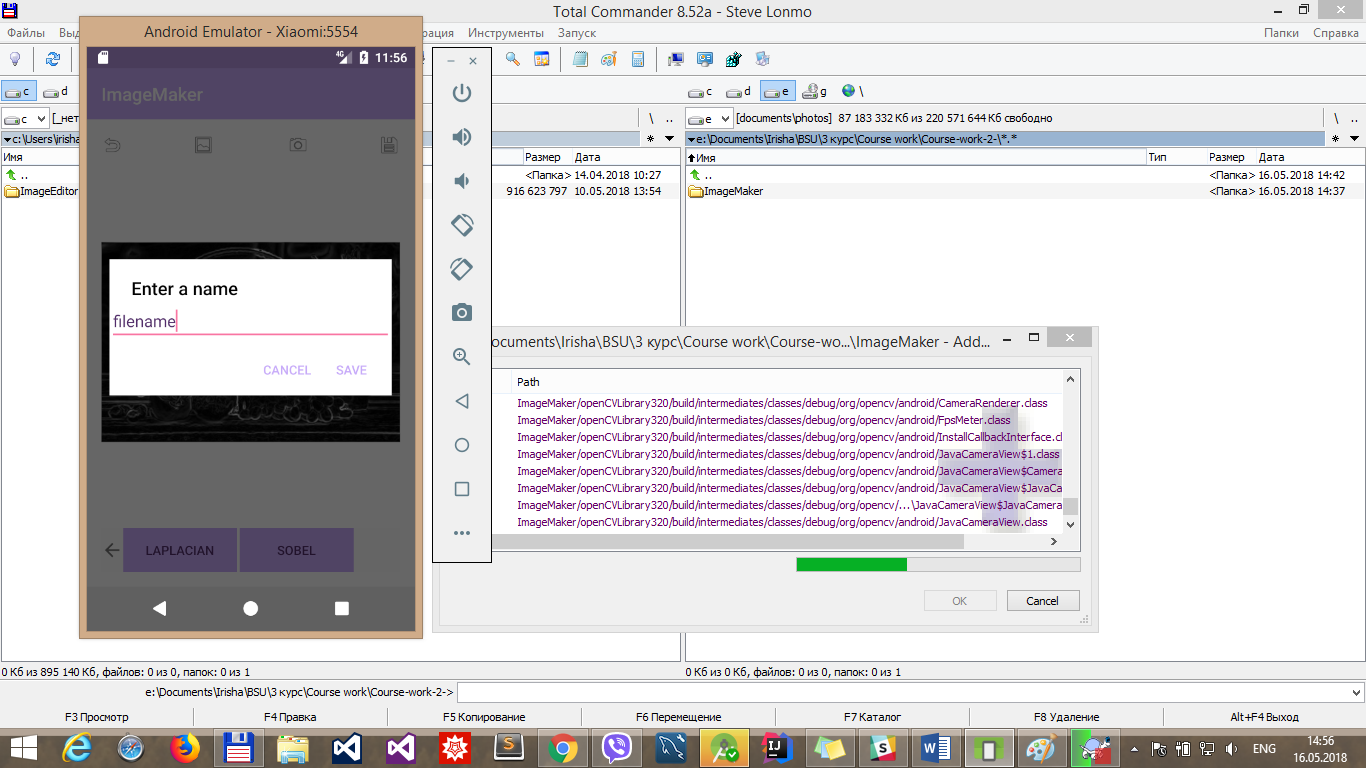
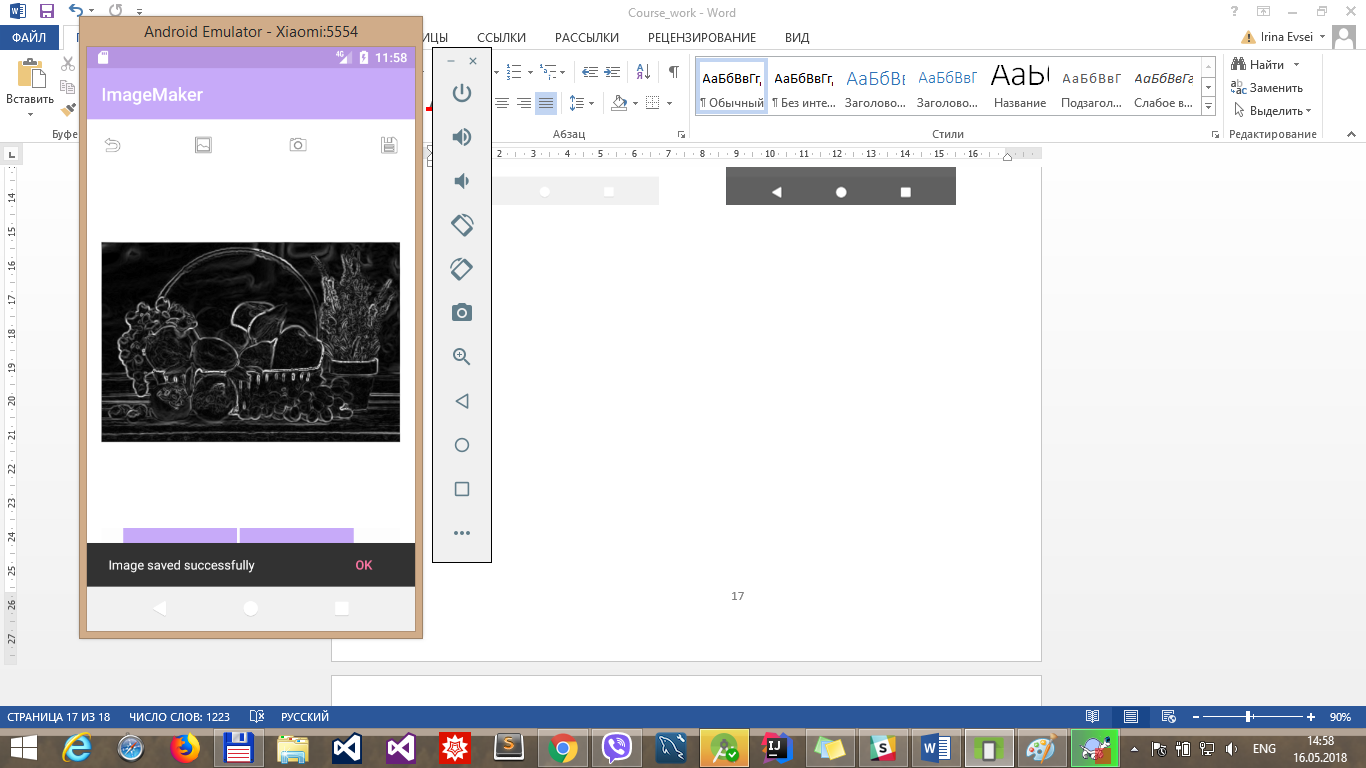
|  |  |
| --- | --- |
| Grayscale filter | Рисунок 13.1 Применение фильтра Grayscale |
| Sepia | Рисунок 13.2 Применение фильтра Sepia |
| Negative | Рисунок 13.3 Применение фильтра Negative |
| Histogram | Рисунок 13.4 Применение фильтра Histogram |
| Brightness -> increase | Рисунок 13.5 Увеличение яркости изображения |
| Contrast -> increase | Рисунок 13.6 Увеличение контраста изображения |
| Sobel | Рисунок 13.7 Применение Sobel к изображения |

**ПРИЛОЖЕНИЕ C**.

Процесс сохранения избражения:



Рисунок 14.1 Клик на иконку для сохранения

Рисунок 14.2 Ввод имени картинки Рисунок 14.3 Клик «SAVE» и появление сообщения о сохранении

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Официальное руководство по разработке на Android: <https://developer.android.com/guide/index.html>
2. Официальная документация для библиотеки OpenCV: <https://opencv.org/>
3. Stackoverflow documentation: <https://stackoverflow.com/documentation>
4. Habrahabr’s article: <https://habrahabr.ru/>
5. Wikipedia: <https://ru.wikipedia.org/>
6. ‘Thinking in Java, 4nd edition’, Revision by Bruce Eckel
7. Инструмент управления растровым изображением:

<https://www.tutorialspoint.com/android/android_image_effects.htm>