1. **ОБЗОР ИСТОЧНИКОВ**
   1. **Обоснование разработки**

Для того, чтобы приступить к разработке, было необходимо ознакомится с основынми подходами к реализации рисовния на компьетере, изучить основные концепты существующих графических интерфейсов, для создания собственного аналога, ничем не уступаещим вышеупомянутым. Требовалось изучить основные принципы работы с изображениями : способы хранения значений цветов, методы выявления и удаления шумов. Необходимо иметь четкое представление о том, как хранится то или иное изображения, для того чтобы любая его модификация была корректной. Так же было необходимо хорошо владеть языком программирования С++, для реализации сложных видов взаимосвязи модулей программы и, в то же время, для максимально эффективного использования ресурсов машины. Для написания грамотного кода требовалось ознакомиться с так называемыми “соглашениями” по написанию кода на С++, чтобы возможные расширения программы или ее рефакторинг не вызывали затруднений при реализации.

* 1. **Основные термины**

**Пиксель** – наименьший логический элемент двумерного цифрового изображения в растровой графике, или элемент матрицы дисплеев, формирующих изображение. Пиксель представляет собой неделимый объект прямоугольной или круглой формы, характеризуемый определённым цветом.

**Растр** – сетка пикселей, обладающих следующими важными свойствами: размер изображения в пикселях (его ширина и высота), количеством используемых цветов или глубиной цвета, цветовой моделью (например RGB) и разрешение.

**RGB** (аббревиатура от английского red, green, blue) –  [аддитивная](https://ru.wikipedia.org/wiki/%D0%90%D0%B4%D0%B4%D0%B8%D1%82%D0%B8%D0%B2%D0%BD%D0%BE%D0%B5_%D1%81%D0%BC%D0%B5%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%D1%86%D0%B2%D0%B5%D1%82%D0%BE%D0%B2) [цветовая модель](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C), описывающая способ кодирования [цвета](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82) для [цветовоспроизведения](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82%D0%BE%D0%B2%D0%BE%D1%81%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5). Выбор основных цветов обусловлен особенностями физиологии восприятия цвета [сетчаткой](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BB%D0%B1%D0%BE%D1%87%D0%BA%D0%B8_(%D1%81%D0%B5%D1%82%D1%87%D0%B0%D1%82%D0%BA%D0%B0)) человеческого глаза.

**Медианный фильтр** – один из видов [цифровых фильтров](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D1%84%D0%B8%D0%BB%D1%8C%D1%82%D1%80), широко используемый в [цифровой обработке сигналов](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D1%81%D0%B8%D0%B3%D0%BD%D0%B0%D0%BB%D0%BE%D0%B2) и [изображений](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BE%D0%B1%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%BA%D0%B0_%D0%B8%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B9) для уменьшения уровня [шума](https://ru.wikipedia.org/wiki/%D0%A8%D1%83%D0%BC). Значения отсчётов внутри окна фильтра сортируются в порядке возрастания (убывания); и значение, находящееся в середине упорядоченного списка, поступает на выход фильтра. В случае чётного числа отсчётов в окне выходное значение фильтра равно среднему значению двух отсчётов в середине упорядоченного списка. Окно перемещается вдоль фильтруемого сигнала и вычисления повторяются. Медианная фильтрация – эффективная процедура обработки сигналов, подверженных воздействию импульсных помех.

**Гамма-коррекция**  – предыскажения [яркости](https://ru.wikipedia.org/wiki/%D0%AF%D1%80%D0%BA%D0%BE%D1%81%D1%82%D1%8C) чёрно-белого или [цветоделённых](https://ru.wikipedia.org/wiki/%D0%A6%D0%B2%D0%B5%D1%82%D0%BE%D0%B4%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5) составляющих цветного [изображения](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%BE%D0%B1%D1%80%D0%B0%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5) при его записи в  [цифровой фотографии](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%84%D0%BE%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F).

**Апертура** – прямоугольная область вокруг текущего [пикселя](https://ru.wikipedia.org/wiki/%D0%9F%D0%B8%D0%BA%D1%81%D0%B5%D0%BB), участвующая в преобразовании.

* 1. **Обзор аналогов**

В сфере работы с графикой существует множество приложений, обеспечивающих редактирование изображений : MicroSoft Paint, PaintBrush for Mac, gPaint for Linux и многие другие. Все эти приложения – серьезные конкуренты, и поэтому, для того, чтобы сделать данный проект конкурентно-способным, необходимо в первую очередь реализовать функционал, объединяющий уже существующие аналоги. Для этого выделим основные возможности конкурентов:

1. Поддержка многих форматов изображений.
2. Поддержка отмены последних действий.
3. Возможность изменения размеров изображений.
4. Рисование простейших фигур.

Крайне сложно перечислить все существующие на данный момент графические редакторы : число их растет постоянно, а отличаются они тем, что стараются обогнать конкурентов, углубляясь в более узкие отрасли работы с графикой. Но все они, при этом, имеют и общий функционал, от которого и будет отталкивать данный проект.

* 1. **Основные технологии**

Для написания курсового проекта был выбран язык программирования С++. Это компилируемый, мультипарадигмальный язык общего назначения с поддержкой процедурного программирования (возможность реализации на чистом С), обобщенного программирования (шаблоны) и объектно-ориентированного (классы и их объекты), последнее из которых и лежит в основе данного курсового проекта. С++ широко используется для разработки программного обеспечения, являясь при этом одним из самых популярных я зыков программирования в мире. На этом языке можно разработать как простейшие консольные приложения, так и целые операционные системы и драйверы.­­­­­­­

Разработка курсового проекта строилась на основе парадигмы объектно-ориентированного программирования, которая подразумевает представление всей программы в виде совокупности взаимодействующих между собой объектов, которые являющихся экземплярами некоторых абстрактных сущностей – классов. Сами классы в свою очередь обладают собственными функциями, называемыми методами, которые их реализую.

Широко использовались возможности языка С++, такие как работа с указателями и динамической памятью, хранение объектов в стандартных контейнерах библиотеки STL, использование абстрактных классов для доступа к классам потомкам с помощью полиморфизма.

Создание проекта выполнялось в кроссплатформенном фреймфорке Qt версии 5.9.2. Данный фреймворк позволяет упростить работу с реализацией графического интерфейсом за счет наличия метаобъектного компилятора, обеспечивающего механизм сигналов/слотов, когда одни классы способны испускать сигналы при определенных условиях(например при взаимодействии с пользователем), а другие перехватывать их и выполнять некоторые предопределенные операции (слоты). Для сборки проекта был использован компилятор MinGW версии 5.3.0

* 1. **Функциональные требования**

После анализа аналогов можно сформулировать основные требования, которым должен удовлетворять проект :

1. Поддержка многих форматов изображений.
2. Рисование фигур.
3. Сохранение изображений.
4. Изменение размеров изображений
5. Возможность отката последних действий
6. Возможность поворота и отражения изображения

Теперь определим дополнительный функционал, который будет являться отличительной чертой данного проекта:

1. Медианная фильтрация изображения
2. Коррекция гаммы изображения
3. Реализация собственной версии заливки области изображения