Государственное образовательное учреждение высшего

профессионального образования

«Московский Государственный Технический Университет имени

Н.Э. Баумана» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»

КАФЕДРА «ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ»

КУРСОВОЙ ПРОЕКТ

Программа для распознания рукописных данных с последующей их обработкой в виде графиков

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Мацак И. В.

(подпись, дата)

Разработчик проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Григорьев А. С.

(подпись, дата)

Москва 2020

Оглавление

[Цель 3](#_Toc58873420)

[Основные определения 3](#_Toc58873421)

[Введение 5](#_Toc58873422)

[Требования к проекту 5](#_Toc58873423)

[Проектирование системы 6](#_Toc58873424)

[Выбор технологий 9](#_Toc58873425)

[Выбор используемых библиотек 14](#_Toc58873426)

# Цель

Создание программы для распознания с фотографии цифровых данных с последующей их обработкой в виде различных графиков с графическим интерфейсом на алгоритмическом языке программирования Си++ при помощи знаний и навыков, полученных на 1 и 2 курсе по специальности «Информационная безопасность».

# Основные определения

**Алгоритмический язык программирования** - формальный язык, используемый для записи, реализации и изучения алгоритмов. В отличие от большинства языков программирования, алгоритмический язык не привязан к архитектуре компьютера, не содержит деталей, связанных с устройством машины.

**Объектно-ориентированное программирование** (ООП) - методология программирования, основанная на представлении программы в виде совокупности [объектов,](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) каждый из которых является экземпляром определенного [класса, а классы образуют иерархию наследования.](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5))

**SFML** - свободная кроссплатформенная  [библиотека,](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) содержащая ряд модулей для простого программирования мультимедиа приложений.

**Qt -** свободная кроссплатформенная библиотека для построения графического интерфейса.

**Tesseract** - это движок оптического распознавания символов (OCR) с открытым исходным кодом.

**CI** (англ. Continuous Integration**)** - это практика разработки программного обеспечения, которая заключается в выполнении частых автоматизированных сборок проекта для скорейшего выявления и решения интеграционных проблем.

**Интерфейс** [(англ. interface)](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) - общая граница между двумя функциональными объектами, требования к которой определяются стандартом. Это совокупность средств, методов и правил взаимодействиямежду элементами [системы.](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)

# Введение

Зачастую в различных предметных областях, например в физике, требуется по имеющимся дискретным экспериментальным или теоретическим данным о значении тех или иных величин установить, как приблизительно будет вести себя одна величина при изменении другой.

Для этой цели строят графики функции по данным точкам. Существуют несколько видов графиков: линейный, логарифмический и полулогарифмический. На шкале в [линейном масштабе](https://ru.wikipedia.org/w/index.php?title=%D0%9B%D0%B8%D0%BD%D0%B5%D0%B9%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%88%D1%82%D0%B0%D0%B1&action=edit&redlink=1) длина отрезка пропорциональна разности величин на его концах, а в логарифмическом -  пропорциональна [логарифму](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC) отношения величин. В полулогарифмическом масштабе логарифмической является только одна из осей X или Y. Линейный масштаб следует выбирать при сопоставимых значениях величин на небольших промежутках, а логарифмическом – при их изменении в больших пределах.

Реализация данной программы позволит быстро строить необходимые графики при наличии фотографии с двумя векторами значений величин.

# Требования к проекту

Данная программа должна удовлетворять следующим требованиям:

* отделение данных от логики (основной принцип ООП)
* автономность, т. е. работа программы без вмешательства человека
* способность распознать цифры на фото
* корректное построение графиков
* наличие пользовательского интерфейса
* поддержка системы Continuous Integration (CI)

# Проектирование системы

Система данной модели дорожного движения состоит из 3 классов:

* Класс построения графика Drawer (Drawer.cpp/.hpp)
* Класс пользовательского интерфейса для выбора фотографии Win win.cpp/.hpp)
* Класс пользовательского интерфейса для выбора типа графика AnotherWin (anotherwin.cpp/.hpp)

**Drawer –** класс для построения графиков по точкам. Его основным полем является объект из библиотеки SFML под названием window, который служит для отображения координатной плоскости, точек и линий, их соединяющих. Другие поля служат для задания параметров координатной плоскости и способов построения точек, в зависимости от выбранного метода, например:

* unsigned int x\_size - размер изображения по оси X в пикселях
* unsigned int new\_center\_x = x\_size / 2 – переопределение центра координатной плоскости по оси X
* double limit\_x0 - минимальное значение по оси X
* double limit\_x1 - максимальное значение по оси X
* double step\_x\_lin – размер шага по оси X в линейном масштабе
* std::array<double, 4> step\_x\_log – размер шага по оси X в логарифмическом масштабе
* double count\_marker\_x – количество делений по оси X
* double scaling\_factor\_x – коэффициент масштабирования по оси X

Рассмотрим некоторые методы данного класса:

* void cicle(const std::vector<double>& x\_vec\_, const std::vector<double>& y\_vec\_) - главный цикл приложения для запуска окна с помощью SFML
* void set\_limit\_x\_lin() – определение крайних значений по оси X в линейном масштабе
* void set\_limit\_x\_log() – определение крайних значений по оси X в логарифмическом масштабе
* void draw\_plate(sf::RenderWindow& window) – отрисовка координатной плоскости
* void draw\_curve(sf::RenderWindow& window) – отрисовка точек и линий, их соединяющих
* void calculate\_scaling\_lin\_x() – расчет коэффициента масштабирования по оси X в линейном масштабе
* void calculate\_scaling\_log\_x(const double& max, const double& min) -расчет коэффициента масштабирования по оси X в логарифмическом масштабе
* double calculate\_point\_x\_lin(const std::vector<std::pair<double, double>>& coordinates, const int& i) – расчет положения точки по оси X в линейном масштабе
* double calculate\_point\_x\_log(const std::vector<std::pair<double, double>>& coordinates, const int& i) – расчет положения точки по оси X в логарифмическом масштабе
* void set\_step\_lin\_x() – расчет шага по оси X в линейном масштабе
* void set\_count\_marker\_x\_linear() – определение количества делений по оси X в линейном масштабе
* void set\_linear\_x() – функция, для задания значения булевой переменной, служащей для информирования программы о выборе масштаба оси X

**UI –** файл пользовательского интерфейса: открытие файла в проводнике, создание окон для выбора типа графика и сообщения об ошибках. Содержит в себе два класса: первый нужен для открытия файла в проводнике и обработки фотографий с помощью Tesseract, второй служит для создания окна выбора типа графика и вызова методов их отрисовки. Оба класса созданы на основе Qt

Рассмотрим основные методы Win:

* Win(QWidget\* parent = 0) – метод, наследуемый от QWidget; создает окно для выбора пути или выхода из программы
* void photo\_processing() – метод для считывания данных с помощью Tesseract и их проверка на корректность
* void error\_occured() – метод для проверки булевой переменной на правильность полученных данных, постройка графика начнется, только если предусмотренных ошибок не возникло

Рассмотрим основные методы AnotherWin:

* AnotherWin (QWidget\* parent) - метод, наследуемый от QWidget; создает окно для выбора типа графика: линейный, логарифмический, линейный по X и логарифмический по Y или наоборот.
* void clicked\_linear() – SLOT-функция, вызываемая при нажатия кнопки построения линейного графика и вызывающая соответствующие методы класса drawer
* void clicked\_log() - SLOT-функция, вызываемая при нажатия кнопки построения логарифмического графика и вызывающая соответствующие методы класса drawer
* и т. д.

# Выбор технологий

**Выбор языка программирования**

Выбор языка программирования очень важная часть в разработке любого проекта. Нужно подобрать именно тот язык, который будет самым оптимальным для конкретной разработки. Именно поэтому чем больше проект, тем больше стек технологий, который в нем используется. Глобальные проекты состоят из огромного количества подпроектов. Для каждого из подпроектов выбирается тот язык, который будет наилучшем образом справляться с задачами, поставленными в нем.

Важными критериями при выборе технологий являются:

• Размер и тип проекта

• Сложность проекта

• Скорость разработки

• Доступные инструменты разработки

• Наличие готовых решений

• Гибкость решения

• Наличие подробной документации

• Требования к нагрузкам

• Требования к безопасности

• Кроссплатформенность

• Возможность интеграции с другими решениями

Так как данная работа планировалась быть не очень объёмной, а время на разработку было ограничено, то стоял вопрос выбора одного языка, а не нескольких.

Всем известно, что языки программирования разделяются по сфере применения. Основными сферами являются веб-разработка, мобильная и игровая разработка. Самыми популярными языками в веб-разработке на данный момент являются: HTML, CSS, JavaScript, Java, Python и PHP . В разработке мобильных приложений на Android: Java, на iOS: Swift и Objective-C.

Я остановился на выборе С++. Ниже приведено подробное доказательство принятого решения.

С++ - язык общего назначения и задуман для того, чтобы настоящие программисты получили удовольствие от самого процесса программирования. За исключением второстепенных деталей он содержит язык С как подмножество. Язык С расширяется введением гибких и эффективных средств, предназначенных для построения новых типов. Программист структурирует свою задачу, определив новые типы, которые точно соответствуют понятиям предметной области задачи. Такой метод построения программы обычно называют абстракцией данных. Информация о типах содержится в некоторых объектах типов, определенных пользователем. С такими объектами можно работать надежно и просто даже в тех случаях, когда их тип нельзя установить на стадии трансляции. Программирование с использованием таких объектов обычно называют объектно-ориентированным. Если этот метод применяется правильно, то программы становятся короче и понятнее, а сопровождение их упрощается.

Достоинства С++:

* Чрезвычайно мощный язык, содержащий средства создания эффективных программ практически любого назначения.
* Компилируемость со статической типизацией.
* Сочетание высокоуровневых и низкоуровневых средств.
* Реализация ООП.
* Работает максимально быстро.
* Предсказуемое выполнение программ, что является важным для построения систем реального времени.
* Автоматический вызов деструкторов объектов при их уничтожении, причём в порядке, обратном вызову конструкторов. Это упрощает (достаточно объявить переменную) и делает более надёжным освобождение ресурсов (память, файлы, семафоры и т. п.), а также позволяет гарантированно выполнять переходы состояний программы, не обязательно связанные с освобождением ресурсов (например, запись в журнал).
* Пользовательские функции-операторы позволяют кратко и ёмко записывать выражения над пользовательскими типами в естественной алгебраической форме.
* Язык поддерживает понятия физической (const) и логической (mutable) константности. Это делает программу надёжнее, так как позволяет компилятору, например, диагностировать ошибочные попытки изменения значения переменной. Объявление константности даёт программисту, читающему текст программы дополнительное представление о правильном использовании классов и функций, а также может являться подсказкой для оптимизации. Перегрузка функций-членов по признаку константности позволяет определять изнутри объекта цели вызова метода (константный для чтения, неконстантный для изменения). Объявление mutable позволяет сохранять логическую константность при использовании кэшей и ленивых вычислений.
* Поддерживаются различные стили и технологии программирования, включая традиционное директивное программирование, ООП, обобщённое программирование, метапрограммирование (шаблоны, макросы).
* Используя шаблоны, возможно создавать обобщённые контейнеры и алгоритмы для разных типов данных, а также специализировать и вычислять на этапе компиляции.
* Возможность имитации расширения языка для поддержки парадигм, которые не поддерживаются компиляторами напрямую. Например, библиотека Boost.Bind позволяет связывать аргументы функций.
* Возможность создания встроенных предметно-ориентированных языков программирования. Такой подход использует, например библиотека
* Кроссплатформенность: стандарт языка накладывает минимальные требования на ЭВМ для запуска скомпилированных программ. Для определения реальных свойств системы выполнения в стандартной библиотеке присутствуют соответствующие возможности (например, std::numeric\_limits ). Доступны компиляторы для большого количества платформ, на языке C++ разрабатывают программы для самых различных платформ и систем.
* Эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы. Ни одна из языковых возможностей, приводящая к дополнительным накладным расходам, не является обязательной для использования — при необходимости язык позволяет обеспечить максимальную эффективность программы.
* Имеется возможность работы на низком уровне с памятью, адресами.
* Высокая совместимость с языком Си, позволяющая использовать весь существующий Си-код (код на Си может быть с минимальными переделками скомпилирован компилятором C++; библиотеки, написанные на Си, обычно могут быть вызваны из C++ непосредственно без каких-либо дополнительных затрат, в том числе и на уровне функций обратного вызова, позволяя библиотекам, написанным на Си, вызывать код, написанный на С++).

В совокупности с вышеописанным, можно сделать вывод, что в данной разработки программного обеспечения C++ - оптимальный выбор.

# Выбор используемых библиотек

В данном проекте требуется использование графической библиотеки для отрисовки графиков по точкам.

SFML – являлся самым оптимальным вариантом графической библиотеки в разработке данного проекта.

Преимущества SFML:

* Наличие понятной документации с примерами.
* Полностью открытый исходный код.
* Содержание ряда модулей (Audio, Window, Graphics, Main, System) для простого программирования мультимедиа приложений.
* Все библиотеки независимы друг от друга.
* Поддержка концепции объективно-ориентированного программирования
* Компиляция и запуск в самых распространенных операционных системах: Windows, Linux, Mac OS X.
* Имеет официальную привязку к языкам С.

Также для данного проекта необходимо было задействовать библиотеку для распознания изображений. В целом, таких библиотек с открытым исходным кодом немного, и лучшей и самой поддерживаемой из них является Tesseract.

Преимущества Tesseract:

* Возможность распознавания текста с изображений.
* Наличие подробной документации.
* Открытый исходный код.
* Возможность переобучения OCR под собственные нужды.

Третьей библиотекой, в которой нуждался проект, была библиотека для создания графических интерфейсов. Необходимо было создать окно вызова проводника и окно выбора графов. Для этой цели была выбрана библиотека Qt.

Преимущества Qt:

* Быстрая многоуровневая разработка.
* Кроссплатформенность.
* Наличие слотов и сигналов.
* Удобное межпроцессное взаимодействие.
* Переносимость на уровне исходного кода ( [Microsoft Windows](https://ru.bmstu.wiki/Microsoft_Windows" \o "Microsoft Windows), [Linux](https://ru.bmstu.wiki/Linux" \o "Linux), [Mac OS X](https://ru.bmstu.wiki/Mac_OS_X" \o "Mac OS X)).
* Хорошая документация.

О**писание технических решений**

В ходе написания курсового проекта возникли следующие проблемы:

1. При построении графиков в линейном масштабе в связи с недостаточной проработанностью методов график при значениях данных больших 50 становился нечитаемым из-за большого размера координатной плоскости. В связи с этим было решено сделать количество интервалов на оси константным и изменить методику расчета граничных значений таким образом, чтобы они немного превышали наибольшую по модулю координату из вектора точек.
2. Точки в логарифмическом масштабе строились некорректно. Проблема была в том, что в логарифмическом масштабе коэффициент масштабирования не является постоянной величиной, а изменяется в зависимости от нахождения точки. Т. е. каждый интервал на логарифмической оси обладает своим коэффициентом, в связи с чем была переделана процедура построения таким образом, чтоб коэффициент рассчитывался для каждой точки отдельно.

Заключение

В процессе выполнения курсового проекта были получены навыки работы с графической библиотекой SFML, Tesseract и Qt, были освоены принципы объектно-ориентированного программирования. Была реализована программа для построения графиков по точкам, чьи координаты даны на фотографии. Построение графиков происходит без ошибок, в читаемом виде.

Список использованных источников

1. Документация: Стандартные библиотеки С++ [Электронный курс]. URL: [www.cplusplus.com/reference/](http://www.cplusplus.com/reference/)
2. Документация: Официальная документация по языку программирования С++ [Электронный курс]. URL: <http://ru.cppreference.com/w/>
3. Документация: Официальная документация по фреймворку SFML [Электронный курс]. URL: <https://www.sfml-dev.org/>
4. Официальный репозиторий графической библиотеки SFML [Электронный курс]. URL: [https://github.com/SFML/SFML/](https://github.com/SFML/SFML)
5. Документация: Документация по библиотеке Qt [Электронный курс]. URL: <http://qt-doc.ru/>
6. Официальный репозиторий библиотеки Tesseract [Электронный курс]. URL: <https://github.com/tesseract-ocr/tesseract>