Министерство образования Республики Беларусь

Учреждение образования БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

К ЗАЩИТЕ ДОПУСТИТЬ

Ассистент кафедры ЭВМ

А. В. Марзалюк

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе тему

ИНФОРМАЦИОННАЯ СИСТЕМА КОМПЬЮТЕРНОГО МАГАЗИНА С ВОЗМОЖНОСТЬЮ ПОДБОРА КОНФИГУРАЦИИ

на

БГУИР КР 1-40 02 01 002 ПЗ

Студент И. Р. Лагодич

Руководитель А.В.Марзалюк

МИНСК 2023

Учреждение образования

«Белорусский государственный университет информатики   
и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

––"–––––––––––––––––20 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту Лагодичу Илье Романовичу

1. Тема проекта Менеджер фотографий

2. Срок сдачи студентом законченного проекта 11 *декабря* 2023 *г.*

3. Исходные данные к проекту: нет.

4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

1. Лист задания.

2. Введение.

3. Постановка задачи

4. Обзор литературы.

4.1. Обзор методов и алгоритмов решения поставленной задачи.

5. Функциональное проектирование.

5.1. Структура входных и выходных данных.

5.2. Разработка диаграммы классов.

5.3. Описание классов.

6. Разработка программных модулей.

6.1. Разработка блок-схем алгоритмов (два наиболее важных метода).

6.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух

методов).

7. Результаты работы программы.

8. Заключение

9. Литература

10. Приложения

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

6. Консультант по проекту(с обозначением разделов проекта)

*Марзалюк А. В*.

7. Дата выдачи задания 15.09.*20*23 *г.*

8. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

1. Выбор задания. Разработка содержания пояснительной

Перечень графического материала - к 01.10.2023 15 %

разделы 2, 3, 4 - к 01.10.2023 10 %:

раздел 5 - к 01.11.2023 20 %:

раздел б - к 01.12.2023 35 %:

разделы 7.89 - к 01.12.2023 5 %:

раздел 10 - к 01.12.2023 5 %:

оформление пояснительной записки и графического материала к 11.12.2023

-10 %:

*Защита курсового проекта с 2*1.12 *по 28*.12.2023 *г.*

РУКОВОДИТЕЛЬ– ––––––––– *Марзалюк А. В.*

Задание принял к исполнению –Х Лагодич *И. Р.*

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc153381038)

[1 ПОСТАНОВКА ЗАДАЧИ 6](#_Toc153381039)

[2 ОБЗОР ЛИТЕРАТУРЫ 7](#_Toc153381040)

[2.1. Обзор методов и алгоритмов решения поставленной задач**и** 7](#_Toc153381041)

[3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 10](#_Toc153381042)

[3.1. Структура входных и выходных данных 10](#_Toc153381043)

[3.2. Разработка диаграммы классов 10](#_Toc153381044)

[3.3 Описание классов 11](#_Toc153381045)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 17](#_Toc153381046)

[4.1. Разработка блок-схем алгоритмов (два наиболее важных метода). 17](#_Toc153381047)

[4.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух 17](#_Toc153381048)

[4.2.1 Метод `drawPlot` класса `GraphDrawer`. 17](#_Toc153381049)

[4.2.2 Метод calculateFunction в классе Function. 18](#_Toc153381050)

[5. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ. 19](#_Toc153381051)

[ЗАКЛЮЧЕНИЕ 22](#_Toc153381052)

[ЛИТЕРАТУРА 23](#_Toc153381053)

[ПРИЛОЖЕНИЕ А 24](#_Toc153381054)

[ПРИЛОЖЕНИЕ Б 25](#_Toc153381055)

[ПРИЛОЖЕНИЕ В 26](#_Toc153381056)

[ПРИЛОЖЕНИЕ Г 27](#_Toc153381057)

# ВВЕДЕНИЕ

Эта работа посвящена разработке программного средства для менеджмента фотографий.

Целью работы является приобретение навыков разработки программных средств для визуализации, изменений и перемещений изображений на компьютере. В данной работе ставится задача создать удобное приложение, которое позволит пользователям перемещать и изменять изображения.

Программное средство будет разработано в виде графического приложения с использованием библиотеки Qt для создания современного пользовательского интерфейса, при этом используется язык программирования C++. Пользовательский интерфейс будет включать в себя следующие основные элементы:

1. Окно для просмотра выбранного изображения.

2. Окно со списком изображений в выбранной директории.

3. Кнопки для выбора и изменений изображений.

В ходе работы применяются различные парадигмы программирования, включая объектно-ориентированное программирование и работу с данными, чтобы обеспечить функциональность приложения.

Таким образом, целью этой работы является создание удобного и мощного программного средства для менеджмента фотографий.

# 1 ПОСТАНОВКА ЗАДАЧИ

Разработать программное средство для менеджмента фотографий. Программа должна предоставлять удобный и интерактивный интерфейс для открытия, изменения и визуализации изображений.

Задачи работы:

1. Изучить основные принципы открытия изображений и их просмотра для дальнейшего их изменения.

2. Разработать пользовательский интерфейс программы, включающий в себя:

- Окно для просмотра выбранного изображения.

- Окно со списком изображений в выбранной директории.

- Кнопки для выбора и изменений изображений.

3. Реализовать алгоритмы переименования и перемещения изображений на основе введенных пользователем данных.

4. Провести тестирование программы на различных изображениях и сценариях использования.

5. Ожидаемые результаты:

# 2 ОБЗОР ЛИТЕРАТУРЫ

В данном разделе проведен обзор существующих методов и алгоритмов, используемых для менеджмента фотографий. Этот обзор поможет определить подходы, которые можно применить при разработке программного средства "Менеджер фотографий" на основе библиотеки Qt и языка программирования C++.

Фотография - технология записи изображения путём регистрации оптических излучений с помощью светочувствительного фотоматериала или полупроводникового преобразователя.

**Изображение, цифровое изображение** (англ. image - образ) - это один графический объект или картинка на экране монитора. Изображения создаются с помощью фотоаппаратов, сканеров, скриншотов, программ для создания и редактирования изображений, например, Paint и Photoshop.

Менеджер фотографий – это программа с широким функционалом, которая позволяет структурировать практически все форматы изображений, просматривать, редактировать, а также распределять их в любые, удобные для вас, категории и альбомы.

Примеры различных форматов изображений:

1. .bmp: Файлы BMP хранятся в виде 2D-изображений различных размеров, цветов и значений глубины цвета без сжатия данных, цветовых профилей или альфа-каналов. Они содержат информацию о пикселях изображения.

2. .png: PNG это растровый формат хранения графической информации, использующий сжатие без потерь по алгоритму Deflate.

3. .jpeg: JPEG это графический формат, применяемый для хранения изображений. Формат JPEG поддерживает, RGB (миллионы цветов) и шкалу серого. JPEG сохраняет всю информацию о цвете в изображении, однако сжимает размер файла путем выборочного удаления данных. Файлы, содержащие данные JPEG, обычно имеют расширения .jpg, .jfif, .jpe или .jpeg. Однако .jpg является самым популярным из них на всех платформах.

**2.1. Обзор методов и алгоритмов решения поставленной задачи**

1. Управление масштабом и интерактивность:

- Масштабирование: Для удобства пользователей важно предоставить возможность изменения масштаба. Это может быть реализовано путем изменения изображения по ширине и высоте.

- Интерактивные элементы: Добавление элементов управления, таких как кнопки для масштабирования, кнопки для переключения между изображениями, выбора папки для их открытия, перемещения, а также переименования.

2. Библиотека Qt и C++:

-Использование библиотеки Qt: Библиотека Qt предоставляет богатые средства для создания графических приложений с интерфейсами. Она включает в себя классы для рисования, обработки событий и создания пользовательских виджетов.

-Язык программирования C++: Программирование на C++ обеспечивает высокую производительность и контроль над памятью, что важно при разработке графических приложений.

3. Исследование аналогичных приложений:

-Приложения для менеджмента фотографий: Исследование существующих приложений, таких как IrfanView[1] (рисунок 2.1.1), XnView, FastStone и Google Photo, может дать представление о стандартах и ожиданиях пользователей.

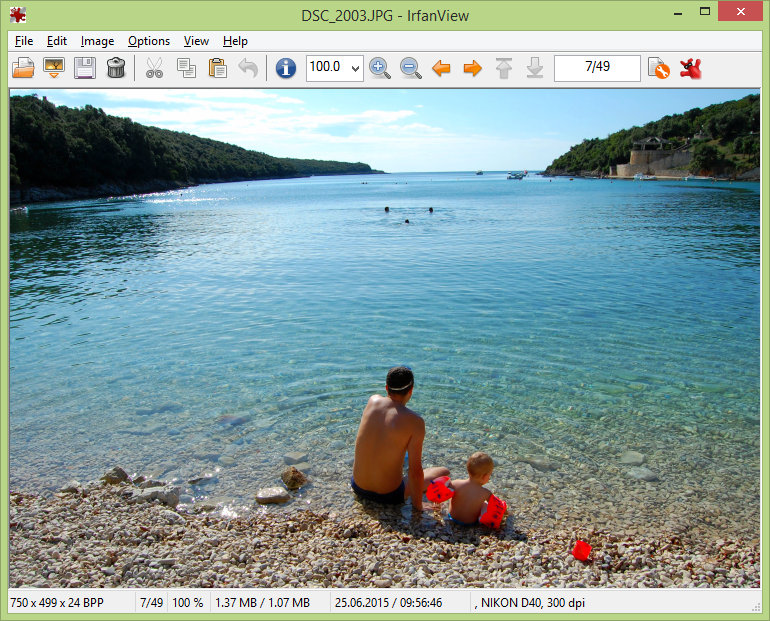


Рисунок 2.1.1 – приложение для построений графиков IrfanView

В заключение, обзор литературы показывает, что существуют разнообразные подходы для постройки пользовательского интерфейса, а также его функционала. Интерактивность и удобство использования играют важную роль в разработке подобных приложений. Применение библиотеки Qt и языка C++ предоставляет мощные инструменты для создания высококачественных приложений.

# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В рамках данной главы рассмотренны основные аспекты функционального проектирования для менеджмента фотографий. Этот этап играет ключевую роль в создании программного продукта, так как определяет структуру входных и выходных данных, а также включает в себя разработку диаграммы классов и описание этих классов.

**3.1. Структура входных и выходных данных**

В качестве входных данных используется любой выбранный файл формата: .bmp ; .png ; .jpeg . Структура которого представлена в таблице 3.1.1.

Таблица 3.1.1 – структура файла изображения

|  |  |
| --- | --- |
| File Height | Размер файла изображения по высоте |
| File Width | Размер файла изображения по ширине |
| File Name | Название файла изображения |
| File Size | Размер файла изображения занимаемый на диске в килобайтах |
| File Directory | Название директории, в которой находится файл |

Эта структура данных позволяет пользователям увидеть название, разрешение(высоту и ширину) и размер файла изображения.

**3.2. Разработка диаграммы классов**

Диаграммы классов являются важным инструментом в объектно-ориентированном программировании и разработке ПО. Вот почему они необходимы:

1. Помогают визуализировать структуру системы, отображая классы, их атрибуты и методы.
2. Облегчают понимание взаимосвязей между классами и пакетами.
3. Содействуют проектированию четких и эффективных архитектурных решений.
4. Позволяют представить базовые отношения и взаимодействия между классами.
5. Создают основу для более глубокого анализа и проектирования системы.

Диаграмма классов представлена в приложении А.

**3.3 Описание классов**

Класс Imageitem – класс для представления базовых параметров изображения и изменения его размера.

Конструктор:

Imageitem(QWidget \*parent = nullptr) – Выделяет память для переменных класса и компонует их в виджете.

Методы:

- setData(QPixmap Image, QString TitleText, QString InfoText, QString Path) – устанавливает изображение, его название, информацию о размере файла и его путь.

- getPath() - возвращает путь изображения.

Приватные поля:

- img - изображение.

- title – название изображения.

- info – информация о размере изображения.

- path – путь к изображению.

Класс Mainwindow – класс для представления изображений и взаимодействия с ними.

Конструктор:

MainWindow(QWidget \*parent = nullptr):

- Создает интерфейс и выделяет память для ярлыка изображения.

Приватные слоты.

- on\_action\_Open\_File\_triggered() – Выбор изображения для открытия.

- setImagesList(QString FileName) – Добавляет изображения в список для их просмотра из указанной директории.

- setOrigin(QString path) – Сохраняет название и указанный путь к изображению.

- on\_listWidget\_itemDoubleClicked(QListWdigetItem \* item) – Устанавливает выбранное изображение из списка в активное окно для просмотра и взаимодействия с ним.

- on\_action\_Zoom\_in\_triggered() – Приближает выбранное изображение умножив текущий размер на 1.25.

- on\_action\_Zoom\_out\_triggered() – Отдаляет выбранное изображение умножив текущий размер на 0.75.

- on\_action\_Original\_size\_triggered() – Устанавливает оригинальный размер изображения в активном окне.

- on\_action\_Adjust\_the\_size\_triggered() – Подгоняет размер выбранного изображения по вертикали/горизонтали(в зависимости от изображения) под размер активного окна.

- on\_actionSave\_File\_triggered() – Сохраняет выбранное изображение.

- on\_actionChoose\_directory\_to\_move\_this\_photo\_triggered() – Выбор директории для перемещения изображений.

- on\_Move\_Photo\_Button\_clicked() – Перемещает изображение в выбранную директорию

- on\_Rename\_Photo\_Button\_clicked() – Переименовывает изображение в написанное пользователем название.

Приватные поля:

- imageLabel – Ярлык изображения.

- fileDir – Название директории с выбранным изображением.

- path – Путь к изображению.

- fileName – Название файла изображения.

- destName – Выбранная директория для перемещения изображений.

# 4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

В данном разделе описана разработка методов приложения которые являются ключевыми для функционирования данное приложения по построению графиков. Более детально были рассмотрены два наиболее значимых метода, а именно drawPlot и calculateFunction, а также представлена шаг за шагом их реализация.

Передо мной стояла задача создать эффективные алгоритмы, способные преобразовывать математические выражения в подходящий для вычислений формат и проводить точные вычисления значений функций. Эти программные модули стали ключевыми строительными блоками моего приложения, определяя его точность и функциональность.

**4.1 Разработка блок-схем алгоритмов (два наиболее важных метода)**

Метод numberAxes в классе GraphDrawer предназначен для размещения чисел на осях графика. Блок-схема данного алгортима предстаавлена в приложении Б.

Метод replaceAllOccurrences в классе Parser предназначен для замены всех вхождений подстроки в строке. Блок-схема данного алгортима предстаавлена в приложении В.

**4.2. Разработка алгоритмов (описание алгоритмов по шагам, для двух методов)**

**4.2.1 Метод `drawPlot` класса `GraphDrawer` отвечает за отрисовку графика функции на виджете `QGraphicsView`**

Входные данные: нет.

Выходные данные: нет.

1. Начало

2. Проверить состояние чекбокса если установлен то очистить график, если нет, ничего не делать.

3. Инициализация переменных:

- QPen pen(Qt::blue): создание пера для рисования с синим цветом.

- error: строка для хранения сообщения об ошибке.

- yVector: вектор для хранения значений функции.

4. Вычисление параметров масштабирования для осей X и Y.

5. Расчет шага и отступов для осей X и Y:

6. Вычисление значений y.

6. Обработка ошибки когда график по оси Y может отображаться некорректно.

7. Обработка ситуации когда график является линий.

8. Вычисление минимальной и максимальной границы для оси Y.

9. Настройка параметров пера для отрисовки осей c последующей отрисовкой

10. Вычисляем значения ‘x’ и отрисовываем график по точкам.

11. Вызов функции “numberAxes” для отрисовки числовых меток на осях.

12. Конец

Этот метод обеспечивает отображение графика функции на виджете с учетом масштабирования и параметров отрисовки.

4.2.2 Метод calculateFunction в классе Function.

Входные данные: Выражение в виде польской записи, значение неизвестной переменной(x).

Выходные данные: Результат вычисления выражения.

1. Начало

2. Инициализация переменных:

- mathExpr: логическая переменная, определяющая наличие математического выражения в строке.

- result: переменная, в которую будет записан результат вычислений.

- error: строка для хранения сообщения об ошибке.

- num: строка для временного хранения числа.

- `variablesStack`: стек для переменных и промежуточных результатов.

3. Разделение выражения на токены используя регулярное выражение с параметром “\\s\* \\s\*”:

4. Проход по токенам.

5. Если токен является математической функцией то производится расчет выражения с помощью функции “calcualteMath” и записывается в “variablesStack”.

6. Если токен является опертором то происходит вычисление выражения в зависимости от оператора с последующей записью в “variablesStack”. Иначе если токен является ‘x’ записываем в “variablesStack” значение ‘x’ переданное пользователем. В любом друом случае записываем данный токен в “variablesStack”.

7. Если токены кончились проверяем не находится ли на вершине “variablesStack” математических функций и если находятся вычисляем иначе переходим к шагу 4.

8. Возвращаем результат

9. Конец

Этот метод эффективно обрабатывает математические выражения, включая функции и переменные, в постфиксной форме, обеспечивая точные результаты.

# 5. РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ.

В рамках данной работы был разработан итерфейс программы представленный на рисунке 5.1.

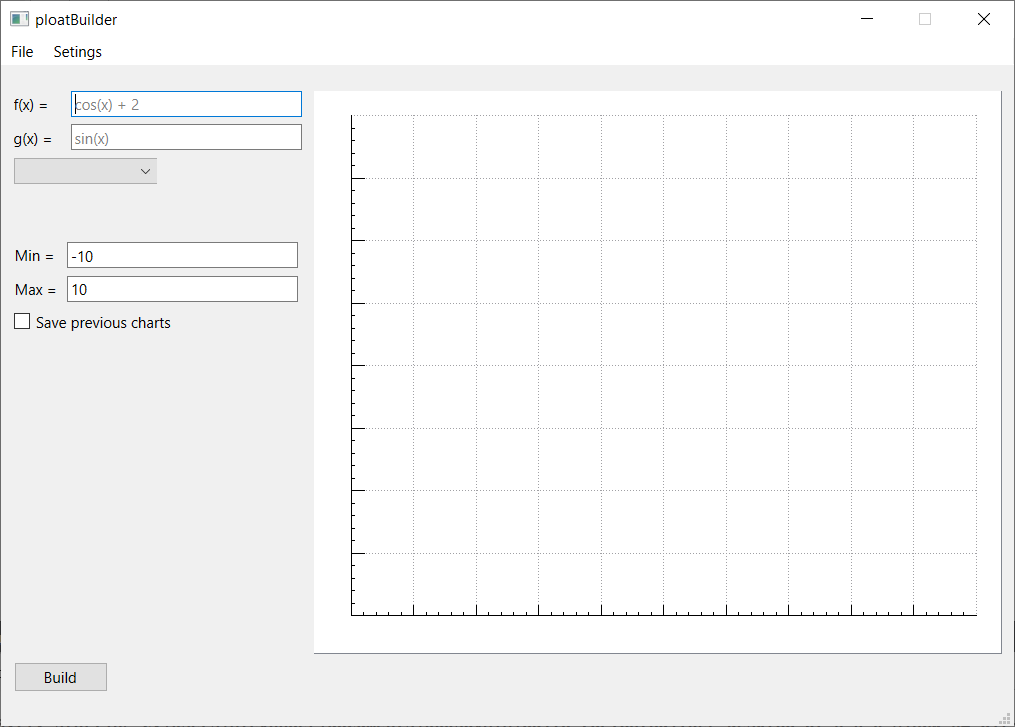


Рисунок 5.1 – Интрефейс программы

На интерфейсе представленно:

* Окно для отрисовки графика.
* Поля для ввода функций:
* f(x) – функций для рассчета y (обязательная).
* g(x) – функций для рассчета x (опциональная).
* Выпадающее окно с различными примерами функций
* Поля для установки минимального(min) и максимального(max) значения по x.
* Кнопка “Build” по нажатию которой происходит отрисовка графика. (рис. 5.2).

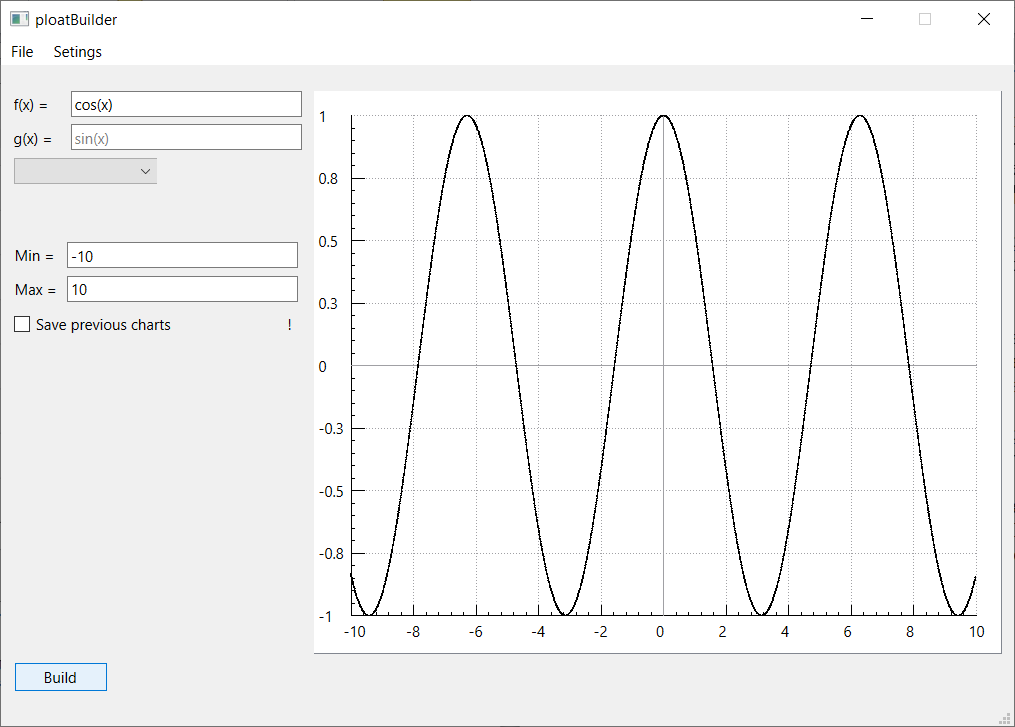


Рисунок 5.2 – Отрисованный график

Флажок “Save previous charts” при активации которого сохраняются ранее отрисованные графики. (рис 5.3)

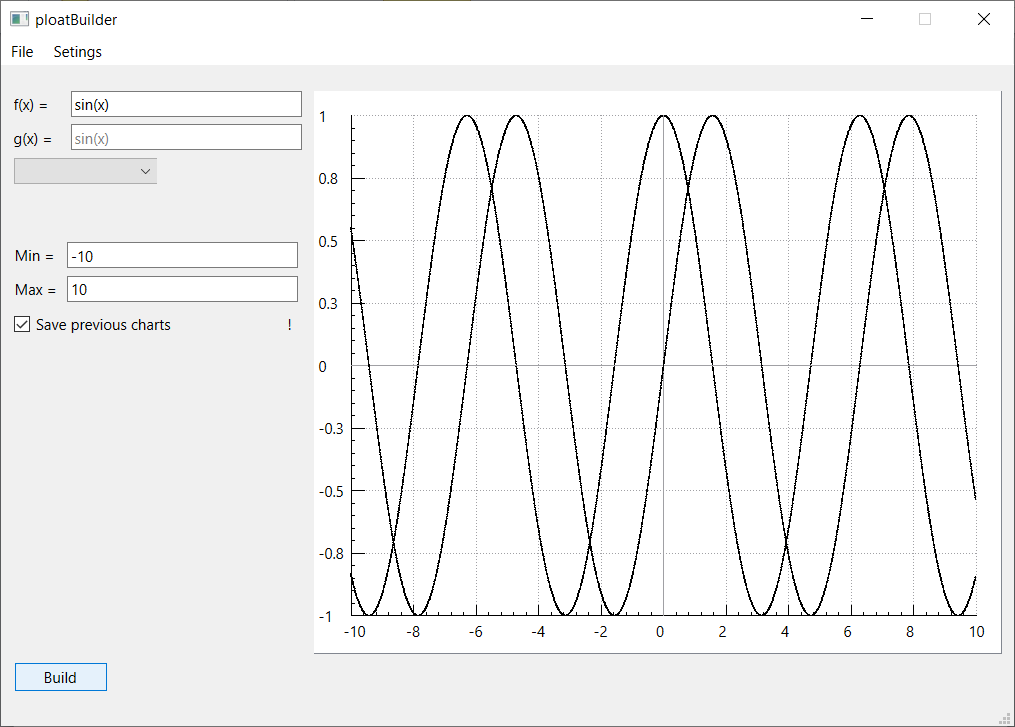


Рисунок 5.3 – Сохранение ранее отрисованных графиков

При переходе по “Setings 🡪 parameters” открывается окно с найстройками графика (рис. 5.4).

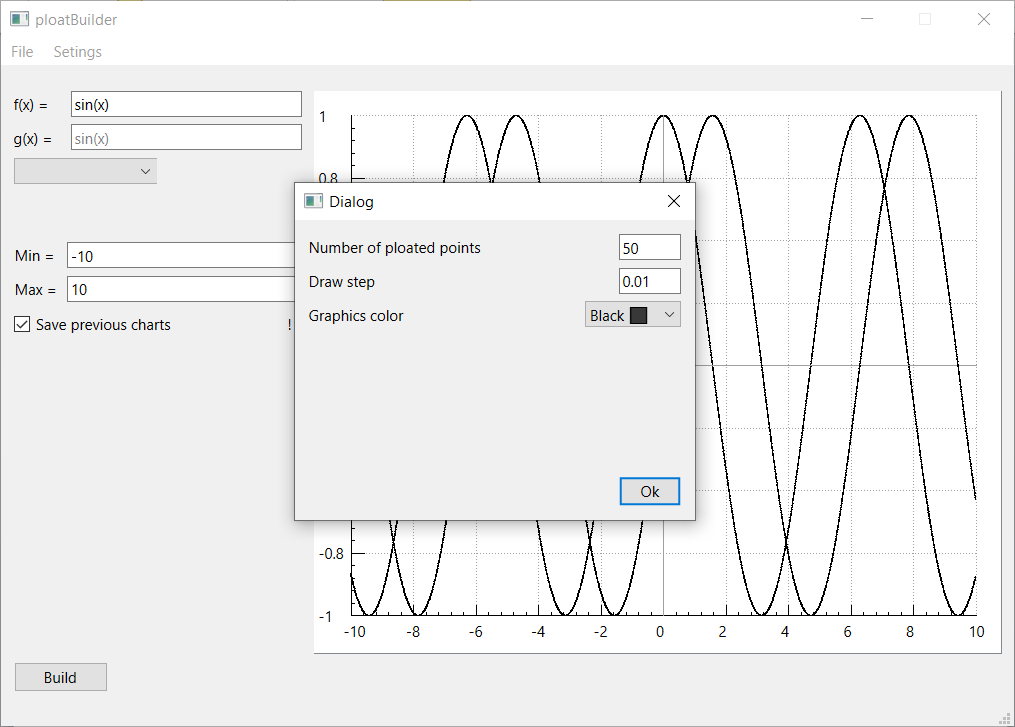


Рисунок 5.4 – Окно настроек графика

Также во вкладке “File” присутствуют пункты меню для сохранения графика, изображения графика и открытия ранее созданного графика (рис. 5.5).

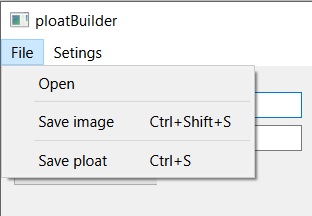


Рисунок 5.5 – Вкладка “File”

# ЗАКЛЮЧЕНИЕ

В заключение проекта подчеркну важность достигнутых результатов и усилий, вложенных в разработку графического приложения для построения математических графиков. В ходе работы было достигнуто несколько ключевых моментов:

1. Функциональность и удобство использования:

- Создано понятное приложение с простым и понятным интерфейсом, что делает построение графиков доступным даже для пользователей без специальных навыков.

2. Стабильность и безопасность:

- Реализованы механизмы обработки ошибок, что обеспечивает стабильную работу программы и предотвращает возможные проблемы при отображении графиков.

3. Масштабирование и адаптация:

- Разработан механизм автоматического масштабирования, учитывающий форму и размер виджета, что обеспечивает корректное отображение графиков при любых условиях.

5. Алгоритмы и оптимизация:

- Реализованы алгоритмы для преобразования выражений и вычисления значений функций, что улучшило общую производительность приложения.

Этот проект был увлекательным вызовом, и результаты, достигнутые в ходе работы, приносят удовлетворение. Программа готова к использованию и может быть полезной для тех, кто интересуется визуализацией математических функций.

# ЛИТЕРАТУРА

[1] WolframAlpha [Электронный ресурс] // Сайт для вычислений и получения ответов на вопросы. [Режим доступа]: https://www.wolframalpha.com/.. – Дата доступа: 25.09.2023

[2] Бьерн Страуструп. Язык программирования С++/ . Пер. с англ. – М.: «Издательство БИНОМ», 2004. – 1098 с.

[3] Скляров В.А. Язык С++ и объектно-ориентированное программирование/. – Мн.: Выш.шк., 1997г. – 478 с.: ил.

[4] Луцик Ю.А., Ковальчук А.М., Лукьянова И.В. Объектноориентированное программирование на языке С++/. – М.: БГУИР, 2003. – 203 с.: ил.

# ПРИЛОЖЕНИЕ А

**(**обязательное**)**

Диаграмма классов

# ПРИЛОЖЕНИЕ Б

**(**обязательное**)**

Схема метода numberAxes

# ПРИЛОЖЕНИЕ В

**(**обязательное**)**

Схема метода replaceAllOccurrences

# ПРИЛОЖЕНИЕ Г

(обязательное)

Исходный текст программы