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

Старший преподаватель

А. В. Мерзалюк

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

ПОСТРОИТЕЛЬ ДВУХМЕРНЫХ ГРАФИКОВ ФУНКЦИЙ

БГУИР КР 1-40 02 01 226 ПЗ

Студент А.В.Антоненко

Руководитель А.М.Ковальчук

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 Антоненко Александру Владимировичу

1. Тема проекта Построитель двухмерных графиков функций

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)

*Ковальчук А. М*.

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– –––––––––*Ковальчук А. М.*

Задание принял к исполнению –Х Антоненко *А. В.*

#### ВВЕДЕНИЕ

Эта курсовая работа посвящена разработке программного средства для построения двухмерных графиков функций.

Целью работы является приобретение навыков разработки программных средств для визуализации математических функций на компьютере. В данной работе ставится задача создать удобное приложение, которое позволит пользователям строить графики функций и анализировать их.

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 ввода математической функции.

2. Окно для настройки параметров графика, таких как масштаб и цвета.

3. Окно с самим графиком функции.

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 построения графиков математических функций.

**1 ПОСТАНОВКА ЗАДАЧИ**

Цель работы:

Разработать программное средство для построения двухмерных графиков математических функций. Программа должна предоставлять удобный и интерактивный интерфейс для ввода, настройки и визуализации графиков функций.

Задачи работы:

1. Изучить основные принципы построения графиков функций и математические методы для их вычисления.

2. Разработать пользовательский интерфейс программы, включающий в себя:

- Окно для ввода математических функций.

- Окно для настройки параметров графиков (масштаб, цвета и т. д.).

- Окно для отображения графиков функций.

3. Реализовать алгоритмы построения графиков функций на основе введенных пользователем данных.

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

5. Подготовить отчет по курсовой работе, включающий в себя введение, описание разработанного программного средства, результаты тестирования и заключение.

Ожидаемые результаты:

В результате выполнения данной курсовой работы ожидается создание программного средства, которое позволит пользователям вводить математические функции, настраивать параметры отображения графиков и получать визуализацию этих функций. Приложение должно быть удобным в использовании и обладать высокой функциональностью.

**2 ОБЗОР ЛИТЕРАТУРЫ**

2.1. Обзор методов и алгоритмов решения поставленной задачи

В данном разделе проведем обзор существующих методов и алгоритмов, используемых для построения двухмерных графиков функций. Этот обзор поможет определить подходы, которые можно применить при разработке программного средства "Построитель двухмерных графиков функций" на основе библиотеки Qt и языка программирования C++.

1. Алгоритмы построения точек графика:

- Метод дискретизации: Один из наиболее распространенных методов. Функция оценивается в дискретных точках на горизонтальной оси, и затем точки соединяются линиями. Этот метод обеспечивает приемлемую скорость построения и хороший результат для большинства функций.

- Алгоритм Брезенхема: Применяется для построения линейных отрезков. Эффективен при рисовании прямых линий и может быть использован для построения графиков.

2. Управление масштабом и интерактивность:

- Масштабирование и перемещение: Для удобства пользователей важно предоставить возможность изменения масштаба. Это может быть реализовано путем изменения диапазона значений на оси X и Y.

- Интерактивные элементы: Добавление элементов управления, таких как ползунки для масштабирования, кнопки для переключения между графиками, анимации и подписи к осям, улучшает интерактивность приложения.

3. 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.

4. Исследование аналогичных приложений:

-Графические приложения для построения графиков: Исследование существующих графических приложений, таких как WolframAlpha, Gnuplot, Matplotlib и Microsoft Excel, может дать представление о стандартах и ожиданиях пользователей.

В заключение, обзор литературы показывает, что существуют разнообразные методы и подходы для построения двухмерных графиков функций. Интерактивность и удобство использования играют важную роль в разработке подобных приложений. Применение библиотеки Qt и языка C++ предоставляет мощные инструменты для создания высококачественных графических приложений.

**3. ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

В рамках данной главы рассмотрим основные аспекты функционального проектирования для построения 2D графиков функций. 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

Перед тем как приступить к построению графиков, необходимо определиться с тем, каким образом программа будет взаимодействовать с пользователями. Входные данные могут представлять собой уравнение функции, диапазон значений переменной, параметры отображения графика и другие необходимые параметры. Выходные данные могут включать в себя сам построенный график и, возможно, дополнительную информацию о нем.

3.2. Разработка диаграммы классов

Диаграмма классов представлена на рисунке 3.1

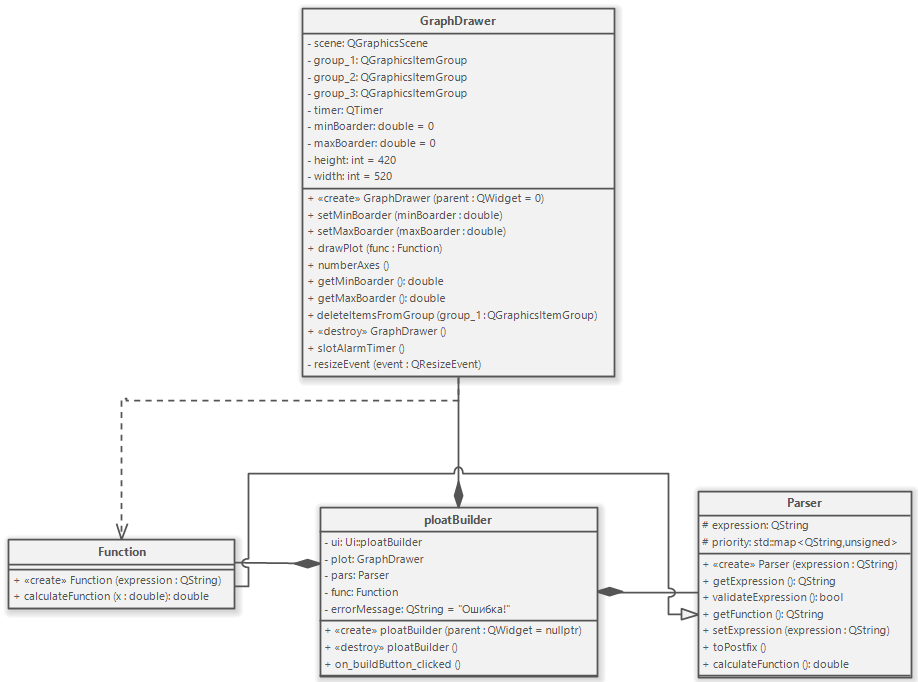


Рисунок 3.1 – Диаграмма классов

3.3 Описание классов

ploatBuilder:

ploatBuilder - это класс, унаследованный от QMainWindow, который представляет собой основное окно приложения для построения графиков.

Конструктор: Принимает указатель на родительский виджет (по умолчанию nullptr).

Деструктор: Очищает ресурсы, выделенные для объекта.

Слоты: on\_buildButton\_clicked(): Обрабатывает нажатие кнопки "Build" и запускает процесс построения графика.

Приватные члены:ui: Указатель на интерфейс ploatBuilder.

plot: Указатель на объект GraphDrawer для отрисовки графика.

pars: Указатель на объект Parser для обработки математических выражений.

func: Указатель на объект Function, являющийся производным от Parser и предназначенным для работы с функциями.

errorMessage: Строка для хранения сообщения об ошибке.

Parser:

Parser - это класс для обработки математических выражений.

Конструктор: Принимает выражение в виде строки.

Методы: getExpression(): Возвращает текущее математическое выражение.

validateExpression(): Проверяет корректность выражения.

getFunction(): Возвращает функцию, представленную в выражении.

setExpression(): Устанавливает новое выражение.

toPostfix(): Преобразует инфиксное выражение в постфиксное.

calculateFunction(): Вычисляет значение функции.

Защищенные члены: expression: Текущее математическое выражение.

priority: Карта приоритетов операторов.

GraphDrawer:

GraphDrawer - это класс, унаследованный от QGraphicsView, предназначенный для отображения графиков функций.

Конструктор: Принимает указатель на родительский виджет (по умолчанию 0).

Методы:

setMinBoarder(): Устанавливает минимальную границу.

setMaxBoarder(): Устанавливает максимальную границу.

drawPlot(): Рисует график функции.

numberAxes(): Нумерует деления на осях.

getMinBoarder(): Возвращает минимальную границу.

getMaxBoarder(): Возвращает максимальную границу.

deleteItemsFromGroup(): Удаляет элементы из группы.

resizeEvent(): Обрабатывает изменение размера окна.

Приватные члены: scene: Объект сцены для отображения графика.

group\_1, group\_2, group\_3: Группы элементов на сцене.

timer: Таймер для анимации.

minBoarder, maxBoarder: Минимальная и максимальная границы графика.

height, width: Высота и ширина окна.

Function:

Function - это класс, унаследованный от Parser, предназначенный для работы с математическими функциями.

Конструктор: Принимает выражение в виде строки.

Методы:calculateFunction(): Вычисляет значение функции для заданного аргумента x.