

Муниципальное общеобразовательное учреждение
«Средняя общеобразовательная школа №33 имени П. А. Столыпина»
Энгельсского муниципального района Саратовской области

Индивидуальный проект по информатике:
«Построение графиков для решения практических задач»

Выполнила:

Ученица 10в класса

Сотникова Елизавета Дмитриевна

Руководитель:

Смирнова Инна Витальевна

2023г.

Оглавление

Введение	3
Глава 1	5
Глава 2	12
Заключение	19
Список источников	20
Приложение 1	21
Приложение 2	22

Введение

В рамках данного индивидуального проекта по информатике в 10 классе будет рассмотрена тема "Построение графиков для решения практических задач". Проект предусматривает разработку приложения, написанного на языке программирования, которое будет предназначено для построения и исследования поведения графиков для решения практических задач. Это приложение будет предоставлять широкий набор инструментов для создания разнообразных типов графиков, а также возможность работы с различными форматами данных.

- **Актуальность проекта:**

Знание и понимание основных математических функций является важным компонентом в обучении математике и ее применении в различных научных и инженерных областях. Графики функций позволяют наглядно представить их поведение и свойства, что помогает визуализировать и анализировать математические модели и решать разнообразные задачи. Однако, ручное построение и исследование графиков может быть трудоемким и требовать значительных усилий. Поэтому разработка приложения, которое облегчит и автоматизирует этот процесс, имеет большую актуальность и практическую значимость. Графический метод решения задач позволяет провести параллель с физикой, где при решении задач достаточно часто используется система координат.

Для определения актуальности проекта, был проведён опрос среди учеников 10-х классов. Результаты опроса (рис.1) показали, что графики используют около 79% участников опроса.

Используете ли вы графики при решении задач школьного курса математики и физики?

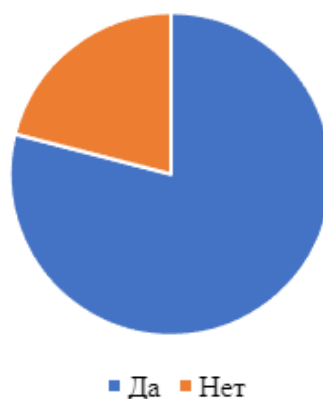


Рис.1 Результаты опроса

- **Цели проекта:**

Разработать приложение на языке программирования Python, которое позволит построить и исследовать графики основных математических функций. Приложение должно быть интуитивно понятным и удобным в использовании, а также обладать функционалом для настройки параметров функций и их визуализации. Проект будет полезен школьникам, студентам и другим людям, связанным с математикой и физикой.

- **Назначение проекта:**

Назначение данного проекта состоит в том, чтобы предоставить пользователям удобный инструмент для построения и исследования графиков основных математических функций для решения практических задач. Приложение должно облегчить процесс работы с функциями и автоматизировать процесс их построения.

- **Задачи проекта:**

1. Изучение основ программирования на языке Python и его библиотек для работы с графиками.
2. Разработка интерфейса приложения для ввода и настройки математических функций.
3. Разработка алгоритмов для построения графиков функций и их визуализации.
4. Реализация функционала для исследования и анализа поведения графиков.
5. Тестирование и отладка приложения для обеспечения его корректной работы.
6. Подготовка документации и презентации проекта.

Успешное выполнение данного проекта позволит создать полезное и удобное приложение для построения и исследования графиков математических функций, которое сможет наглядно продемонстрировать процесс их построения и станет незаменимым инструментом для решения практических задач в различных областях. Это позволит пользователям более эффективно изучать и применять математические функции в обучении и повседневной жизни.

Глава 1

1.1. Роль графиков:

Графики в нашей жизни играют значительную роль. По графикам решают уравнения и находят объемы тел, рассчитывают конструкторские и решают экономические задачи, вычисляют данные для запуска ракет и исследуют реальные процессы. Но, несмотря на практическое значение графиков, в школьной математике графики играют в основном вспомогательную роль и служат обычно для иллюстрации и лучшего запоминания свойств изучаемых функций.

- Благодаря своей наглядности графики позволяют лучше понять решаемую задачу.
- График дает возможность сразу определить, есть ли у данной задачи решение и единственно ли оно. Если исходная задача может иметь несколько вариантов аналитического решения, то график помогает выбрать нужный вариант.
- Графики помогают считать, так как заменяют вычисления по сложным формулам простыми действиями с чертежами. Решать задачи по графикам можно быстро и с достаточной для графики точностью.
- Графики позволяют исследовать изучаемый процесс, подбирать данные и, тем самым, составлять новые интересные задачи. ^[2]

При решении задач графическим способом у учеников возникает понимание необходимости аккуратного отношения к построению графиков, появляется умение работать с ними: правильно выбирать масштаб, производить простые геометрические построения.

1.2. Способы задания функций.

1. Табличный способ.

При этом способе ряд отдельных значений аргумента x_1, x_2, \dots, x_k и соответствующий ему ряд отдельных значений функции y_1, y_2, \dots, y_k задаются в виде таблицы. Несмотря на простоту, такой способ задания функции обладает существенным недостатком, так как не дает полного представления о характере функциональной зависимости между x и y не является наглядным.

2. Словесный способ.

Обычно этот способ задания иллюстрируют примером функции Дирихле $y = D(x)$: если x - рациональное число, то значение функции $D(x)$ равно 1, а если число x — иррациональное, то значение функции $D(x)$ равно нулю. Таким образом, чтобы найти значение $D(x_0)$ при заданном значении $x = x_0$, необходимо каким-либо способом установить, рационально или иррационально число x_0 .

3. Аналитический способ.

При аналитическом способе задания известна формула, по которой по заданному значению аргумента x можно найти соответствующее значение функции y .

4. Графический способ.

Функциональная зависимость может быть задана с помощью графика функции $y = f(x)$. Преимуществом такого способа задания является наглядность, позволяющая установить важные черты поведения функции ^[1].

В рамках данного проекта мы рассмотрим решение задач графическим методом, используя инструменты для его оптимизации.

1.3. Преимущества выбора графического способа.

- *Визуализация данных:* графические методы позволяют наглядно представить информацию, что делает ее более понятной и доступной для анализа.
- *Идентификация трендов и закономерностей:* построение графиков позволяет быстро выявить изменения в данных, выявить тренды и цикличность.
- *Сравнение данных:* с помощью графиков легко сравнивать различные параметры и их изменения во времени, что помогает выявить взаимосвязи и зависимости между ними.

1.4 Виды основных графиков.

- График линейной функции (прямая)

Линейная функция задается уравнением $y = ax + b$. График линейной функций представляет собой **прямую**. Для того, чтобы построить прямую достаточно знать две точки.

Пример 1

Построить график функции $y = 2x + 1$. Найдём две точки. В качестве одной из точек выгодно выбрать ноль.

Если $x = 0$, то $y = 2 * 0 + 1 = 1$.

Берем еще какую-нибудь точку, например, 1.

Если $x = 1$, то $y = 2 * 1 + 1 = 3$.

При оформлении заданий координаты точек обычно сводятся в таблицу:

x	0	1
y	1	3

А сами значения рассчитываются устно или на черновике, калькуляторе.

Две точки найдены, выполним чертеж (рис. 2):

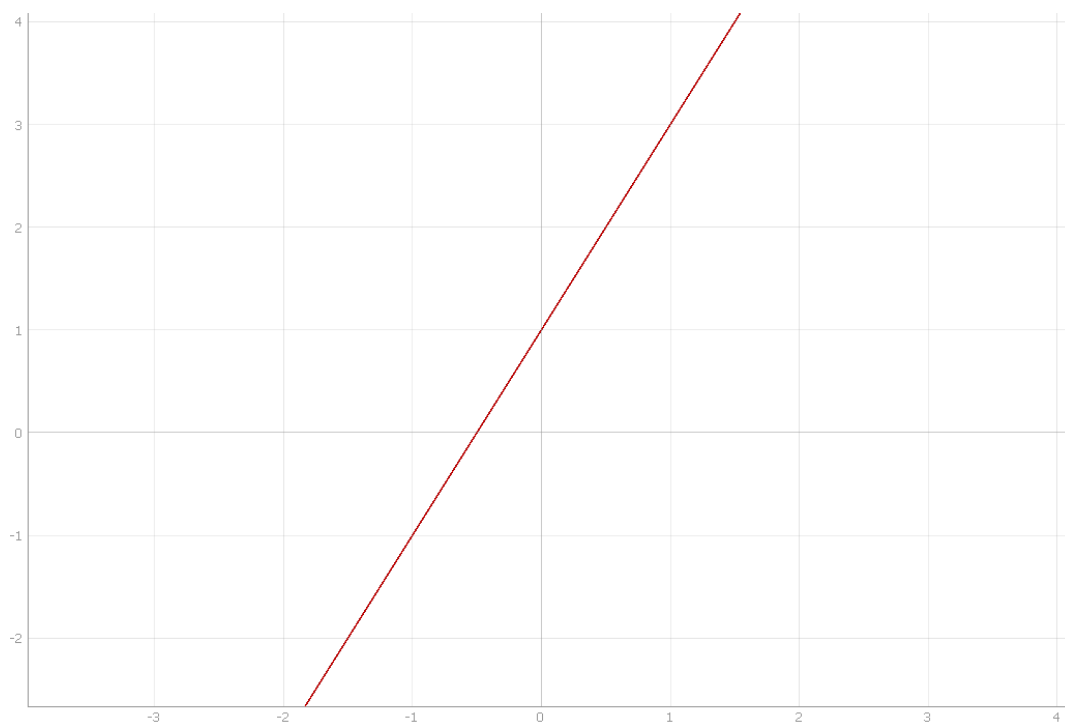


Рис.2 График линейной функции

- График квадратичной функции (парабола)

Парабола. График квадратичной функции $y = ax^2 + bx + c$ ($a \neq 0$) представляет собой параболу (рис. 3). Рассмотрим знаменитый случай: $y = x^2$

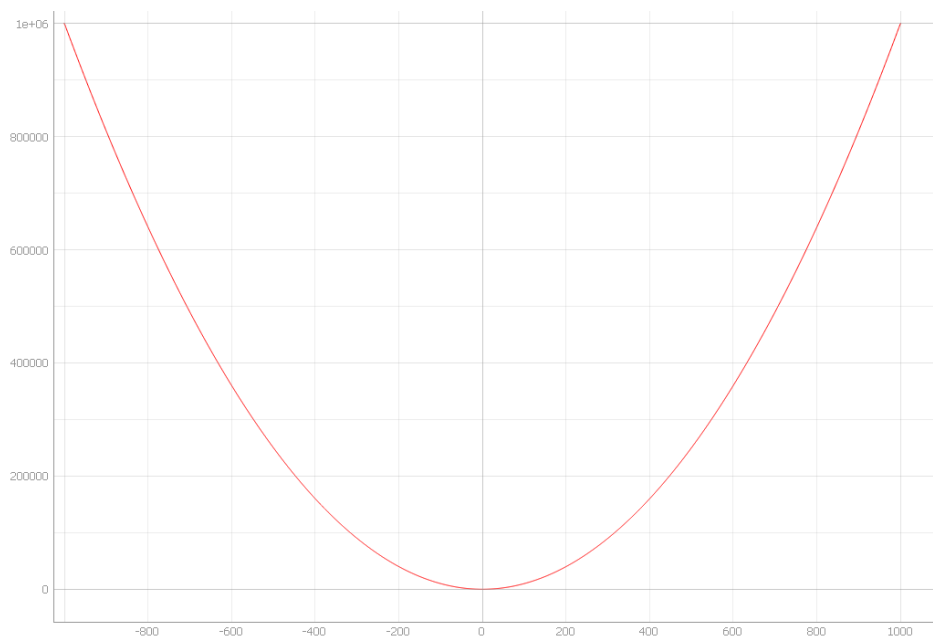


Рис.3 График квадратичной функции

- График кубической функции (кубическая парабола)

Кубическая парабола (рис.4) задается функцией $y = x^3$

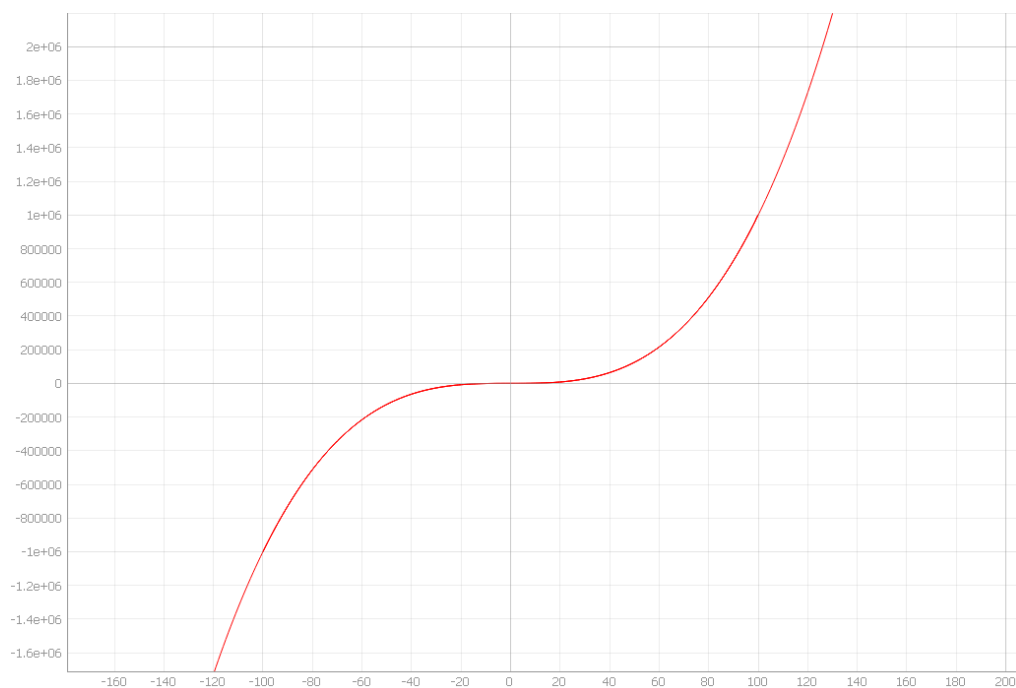


Рис.4 График кубической функции

- График функции $y = \sqrt{x}$

Он представляет собой одну из ветвей параболы. Выполним чертеж (рис. 5):

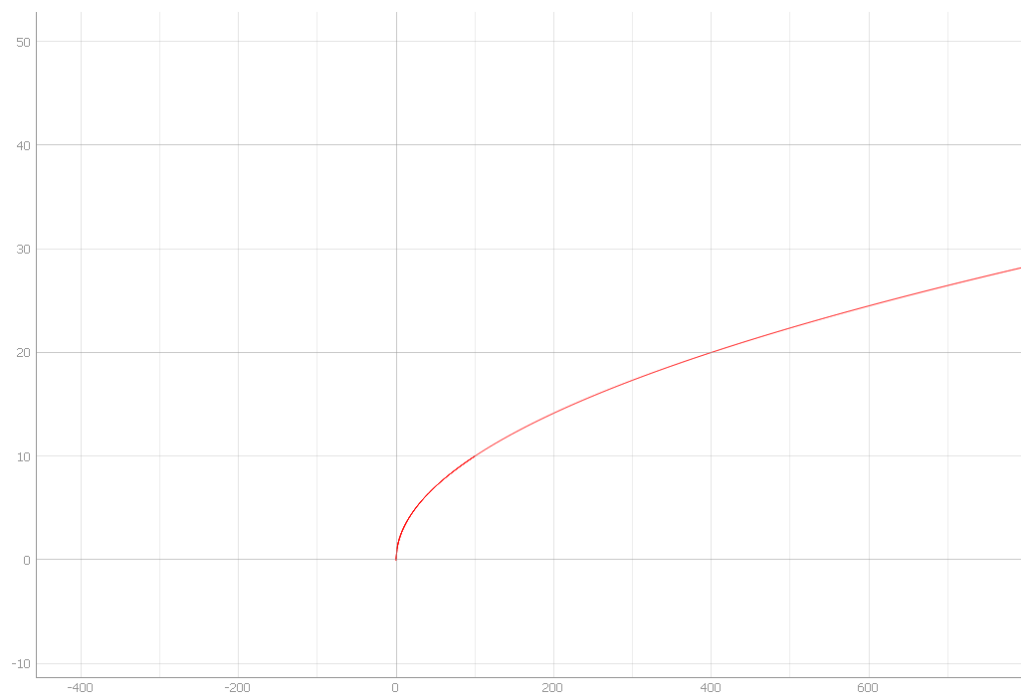


Рис.5 График функции $y = \sqrt{x}$

- График гиперболы

Гипербола задается функцией $y = \frac{1}{x}$.

Выполним чертёж (рис. 6):

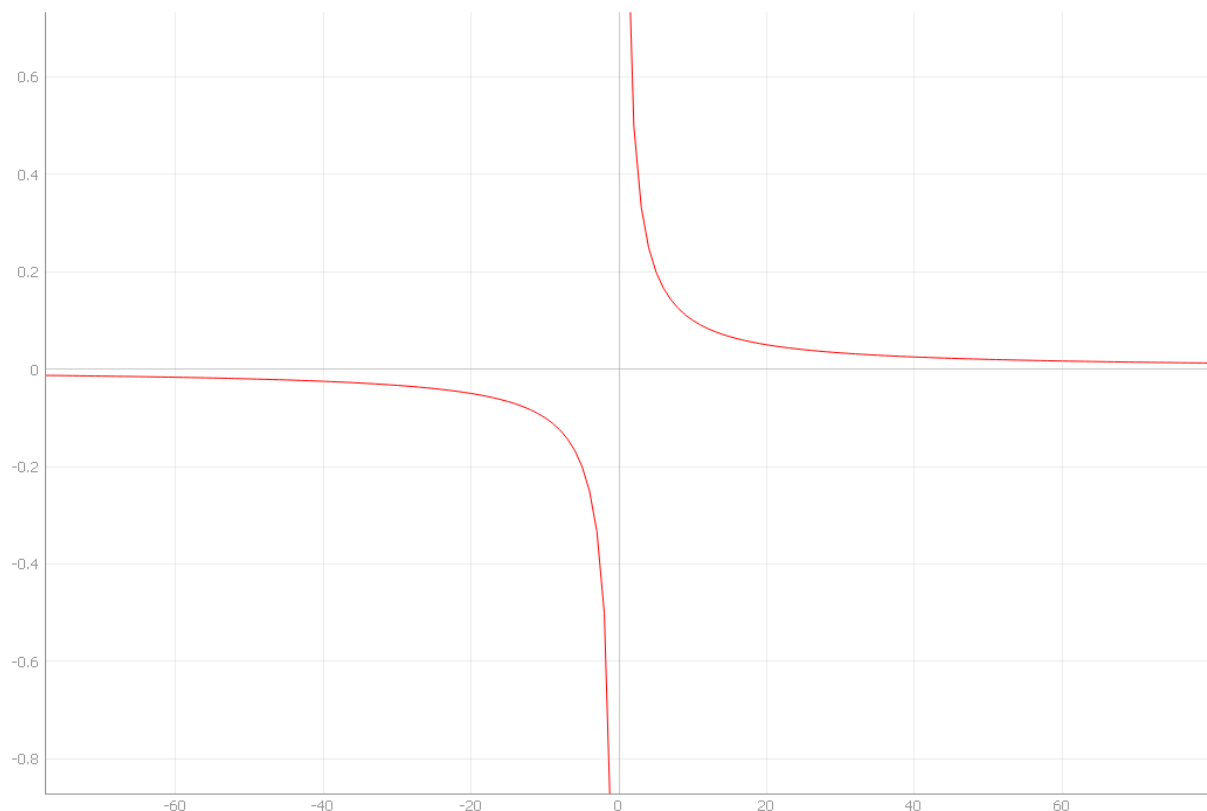


Рис.6 График гиперболы

- График показательной функции
- График логарифмической функции на примере
- Графики тригонометрических функций:
 - 1) синуса;
 - 2) косинуса;
 - 3) тангенса и котангенса.
- Графики обратных тригонометрических функций:
 - 1) арксинуса;
 - 2) арккосинуса;
 - 3) арктангенса и арккотангенса;

1.5. Пример использования графического способа для решения задач.

Пример 1: Два путника вышли одновременно из пункта А в пункт В. Один из них шел со скоростью 6 км/час, а другой шел со скоростью 4 км/час. Через какое время и на каком расстоянии от пункта А первый путник должен повернуть обратно, чтобы встретить второго ровно через 3 часа после выхода их из пункта А?

Решение:

1. Нарисуем траектории движения путников, предполагая, что путники начали движение в 0 часов (рис.7).
2. Из точки 3 оси времени проводим вертикаль до пересечения с траекторией движения второго путника. Точка пересечения, которую мы обозначим буквой С, соответствует месту встречи путников.
3. Двигаясь из точки С со скоростью первого путника, предположив при этом, что время течет в обратном направлении, мы можем восстановить траекторию, по которой первый путник придет в С.
4. Пересечение двух нарисованных траекторий движения первого путника даст нам точку D, соответствующую моменту поворота первого путника.
5. Спроецировав точку D на оси, получим время и место поворота. В данном случае первый путник должен повернуть назад через 2,5 часа после выхода на расстоянии 15 км от пункта А.

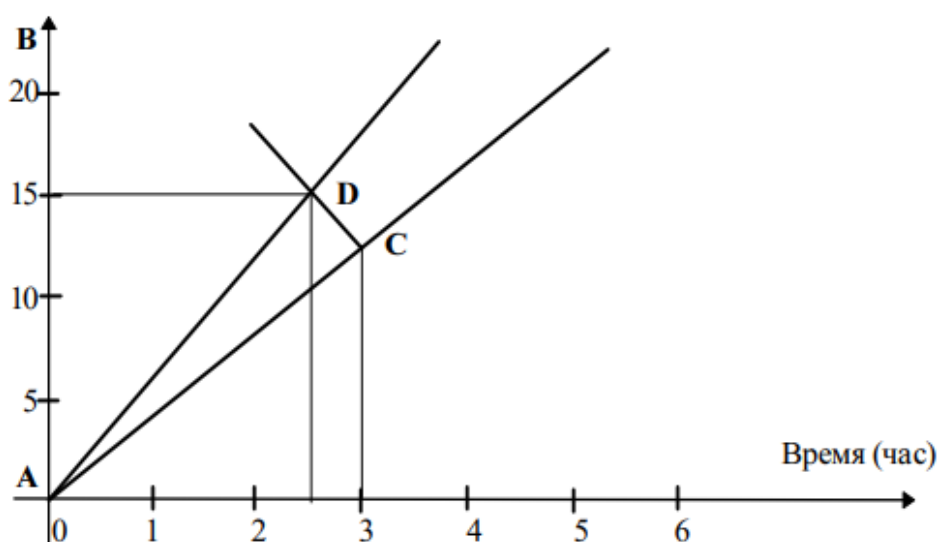


Рис.7 Графическое решение для примера 1

Проверим, решив задачу арифметическим способом.

1. За три часа второй путник пройдет 12 км.
2. Первый путник пройдет за 3 часа 18 км.
3. Первый путник прошел лишние 6 км, так как шел от точки встречи до точки поворота и обратно. Значит точка встречи находится на расстоянии 3 км от точки D.
4. Расстояние от А до D равно 15 км. Это расстояние первый путник пройдет за 2,5 часа.

Ответ: первый путник должен повернуть обратно через 2,5 часа после выхода из пункта А на расстоянии 15 км от него

В данном примере мы задали графики - прямые, используя данные из условия.

Пример 2: Из пункта А вышла грузовая машина со скоростью 60 км/ч. Через 2 часа вслед за ней из А вышла легковая машина со скоростью 90 км/ч. На каком расстоянии от пункта А легковая машина догонит грузовую?

Решение. За начальный отсчет времени берется момент выхода грузовой машины, тогда момент выхода легковой машины будет через два часа. Зная скорости движения объектов, построим графики движения (рис.8). По чертежу видно, что точка пересечения графиков показывает встречу машин, она состоялась на расстоянии 360 км.^[3]

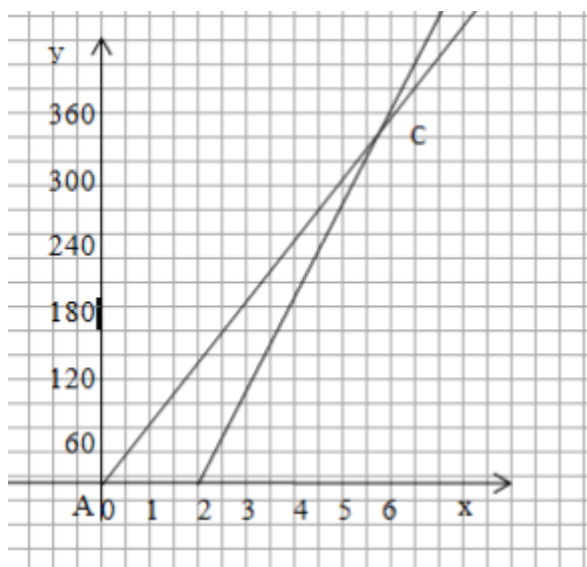


Рис.8 Графическое решение для примера 2

Ответ: 360 км.

Глава 2

Создание приложения

2.1 Создание графического интерфейса

Для создания окон приложения можно использовать готовую среду для разработки графического интерфейса. Например, “**Qt Designer**” - для программ, использующих библиотеку Qt. Устанавливаем программу с официального сайта (см. Приложение 1).

1. Создаем стартовое окно программы (рис. 9), используя текстовое поле для ввода названия проекта пользователя (*QTextEdit*), надписи (*QLabel*) и кнопку для запуска (*QPushButton*)

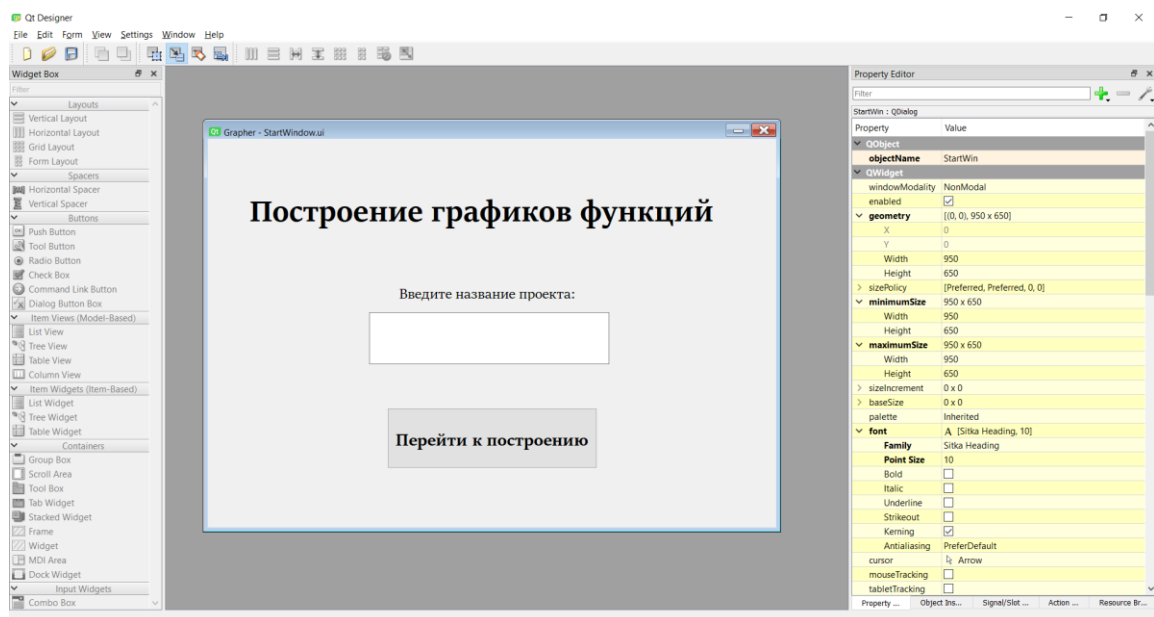


Рис.9 Создание стартового окна

2. Переходим к созданию основного окна (рис.11). Для отображения графиков используем класс пользовательского виджета - *PlotWidget*. Также создаем кнопку добавления графика (*QPushButton*) и кнопки в строке меню (“помощь” и “сохранить”).

Для создания поля для отображения графиков (*Plot Widget*):

- Открываем новый виджет в QtDesigner и добавляем *Graphics View*.
- Нажимаем на него правой кнопкой мыши и выбираем вкладку Преобразовать в... (Promote to...).
- Заполняем поля, как показано на картинке (рис.10):

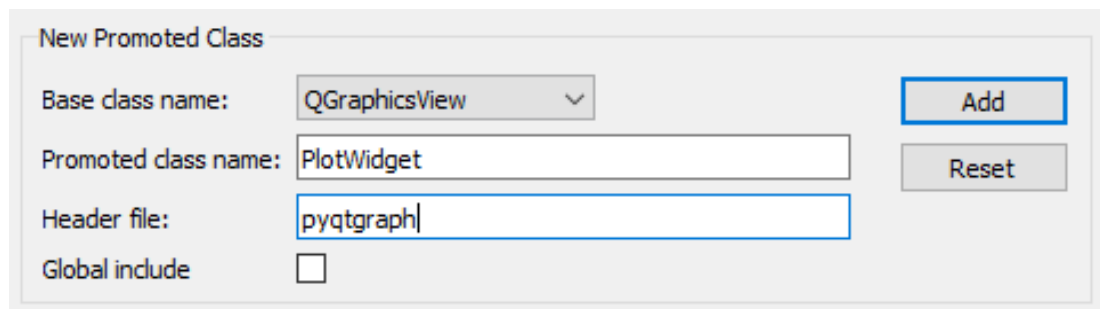


Рис.10 Создание поля для графиков

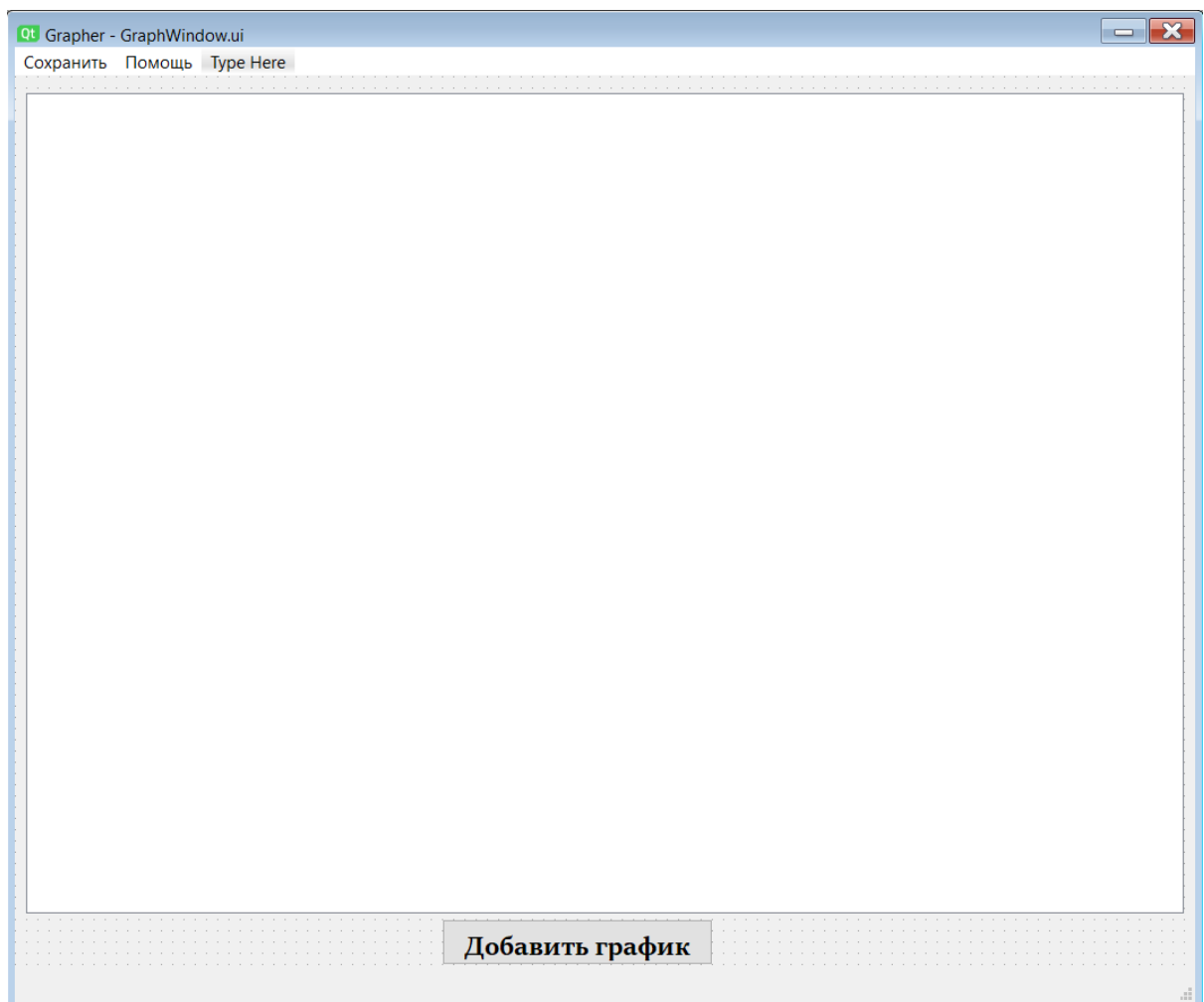


Рис.11 Создание основного окна

2.2 Загрузка файлов с интерфейсом в Python

1. Для начала скачиваем Python и PyCharm (см. Приложение 1).
2. Устанавливаем библиотеки "PyQt5", "PyQtGraph", "numpy", "numexpr" через командную строку:

pip install {имя библиотеки}

Библиотеки “PyQt5” и “PyQtGraph” - для графического интерфейса и отображения графиков.

“numpy” и “numexpr” - для расчета координат при построении.

3. Создаем проект в PyCharm и загружаем туда файлы с интерфейсом.
4. В файле “main.py” импортируем следующие библиотеки:

```
import os
import sys

import numpy as np
import numexpr as ne

from PyQt5 import uic
from PyQt5.Qt import *

import pyqtgraph as pg
import pyqtgraph.exporters
```

5. Для каждого окна создаем класс, наследуемый от определенного виджета. Для загрузки интерфейса добавляем при инициализации следующие строчки:

```
uic.loadUi('StartWindow.ui', self)
```

Для стартового окна, 'StartWindow.ui' - название файла.

```
uic.loadUi('GraphWindow.ui', self)
```

Для основного окна, 'GraphWindow.ui' - название файла.

И так далее.

```
if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = StartWin() # Стартовое окно (класс)
    exGr = GraphWin() # Основное окно (класс)
    exAdd = AddWin() # Окно добавления графиков (класс)
    exSaved = SavedFile() # окно: "Файл успешно сохранен" (класс)
    exHelp = ShowingHelp() # окно показа руководства пользователя
    ex.show() # Изначально - открытие стартового окна
    sys.exit(app.exec_())
```

Рис.12 Код для показа окон

2.3 Основная часть программы

Для прорисовки графика необходимо передать данные, введенные пользователем в окно добавления графика (рис.13):

Рис.13 Окно для добавления графиков

```
def add_OK(self): # функция получения данных, введенных пользователем
    colors = {'Красный': 'r', 'Зелёный': 'g', 'Синий': 'b', 'Чёрный':
'k', 'Фиолетовый': 'violet',
'Оранжевый': 'orange', 'Розовый': 'pink', 'Голубой':
'cyan', 'Белый': 'white'}
    equal = self.functionTE.toPlainText() # заданная функция
    left_limit = self.lineEditFROM.text() # данные из поля "диапазон"
    if left_limit == '':
        left_limit = -100
    left_limit = int(left_limit)
    right_limit = self.lineEditTO.text()
    if right_limit == '':
        right_limit = 100
    right_limit = int(right_limit)
    color = colors[self.colorBox.currentText()]
    mas = [equal, left_limit, right_limit, color] # создает массив
# всех параметров этого графика
    GraphWin.draw(GraphWin(), mas) # вызываем функцию другого класса
# (основного окна) для добавления графика в общий массив
    exAdd.close() # закрываем окно
```

Когда данные (рис.14) переданы в класс основного окна, отображаем график в системе координат (рис.15):

```
for mas in graphics: # отрисовываем каждый график из массива
    e = mas[0]
    x = list(np.arange(mas[1], mas[2] + 1)) # генерируем значения X
```

```
y = list(ne.evaluate(e)) # вычисляем значения для y
self.Field.plot(x, y, pen=mas[3])
```

Add Graphic

Добавить график функции:

x^{**2}

Например: $x^{**2} + 2*x + 3$

Цвет: Чёрный

Диапазон: по оси x

-100 100

ОК Отменить

Рис.14 Пример создания графика

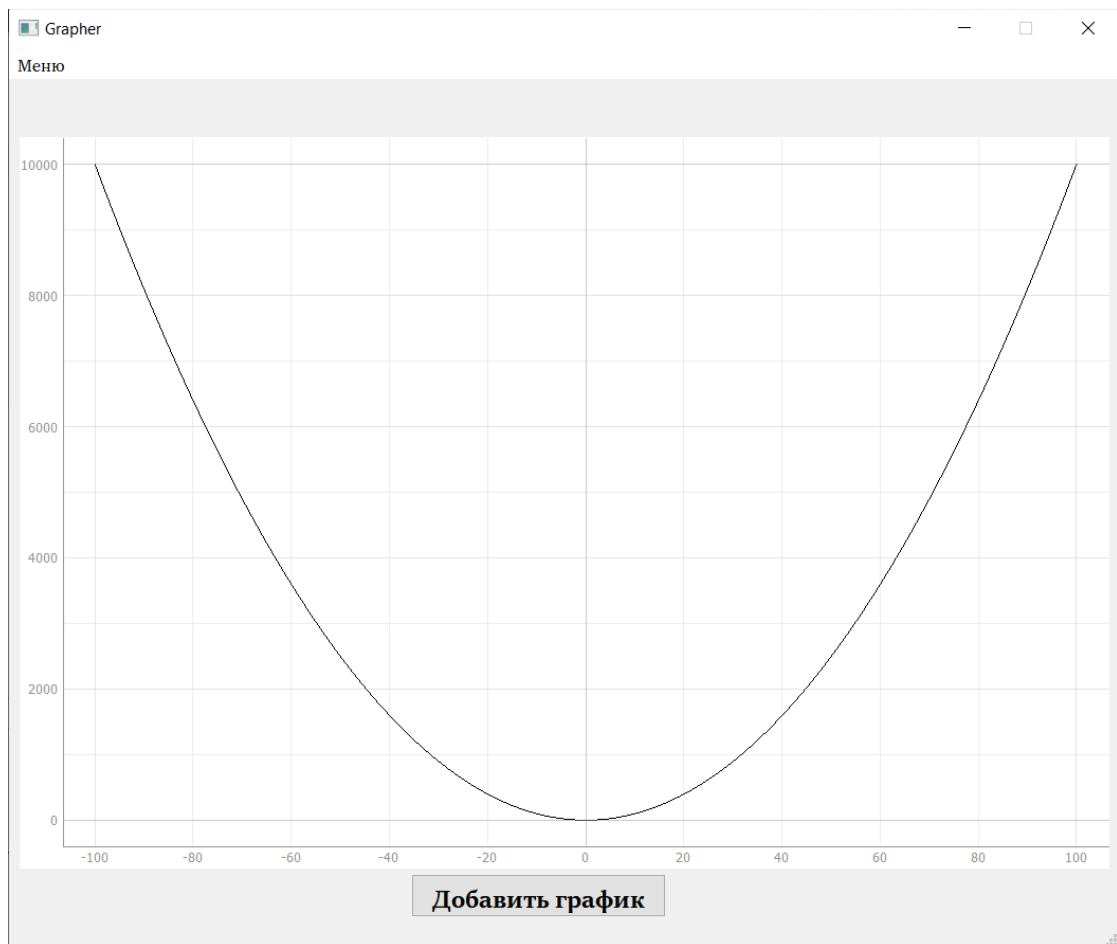


Рис.15 Пример работы программы

Далее полученный график можно сохранить на устройство в одном из форматов:

- PNG (*.png)
- JPEG (*.jpg *.jpeg)

По умолчанию файлы (рис.16) будут сохраняться в папку, из которой запущен проект.

Таким образом проекты не удалятся при закрытии программы и могут быть использованы многократно.

Функция для сохранения изображения:

```
def SaveImage(self): # функция для сохранения, pg - PyQtGraph
    exporter = pg.exporters.ImageExporter(self.Field.scene())
    path = str(os.getcwd()) + '\CREATED_IMAGES' # получение пути к
    текущей папке
    # открытие диалогового окна для сохранения файла:
    filePath, _ = QFileDialog.getSaveFileName(self, "Save Image",
    f"{path}", "PNG(*.png);;JPEG(*.jpg *.jpeg);;All Files(*.*) ")
    exporter.export(filePath) # экспорт поля с графиками
    exSaved.show() # показ следующего окна
```

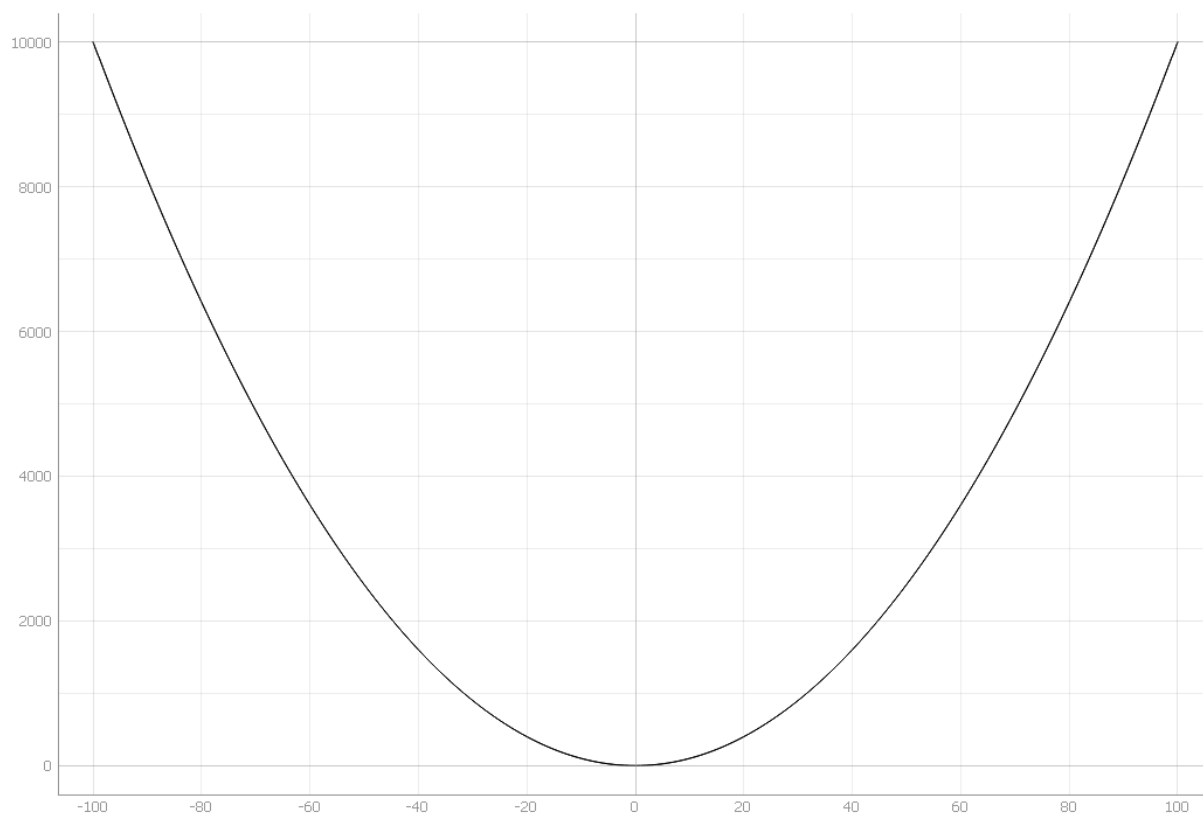


Рис.16 Пример сохранения работы программы в виде изображения

2.4 Помощь при работе с программой / инструкция

Для открытия *“Руководства пользователя”* нужно выбрать опцию *“Помощь”* в верхней строке - меню. Также можно использовать сочетание клавиш *“Ctrl+H”*.

Откроется окно с полной инструкцией к использованию и подробным описанием всех окон (см. Приложение 2).

Этот файл также можно открыть в формате PDF - из папки с проектом (*“HELP.pdf”*).

Заключение

В ходе выполнения индивидуального проекта было разработано полноценное приложение на языке программирования Python с использованием библиотеки PyQt. Программа позволяет удобно и эффективно строить графики таких основных математических функций, как линейная, квадратичная, показательная и т. д.

В процессе разработки были выполнены следующие задачи:

1. Разработка пользовательского интерфейса. Были созданы окна и виджеты, которые позволяют пользователю вводить функцию, выбирать интервалы для построения графика, а также настраивать параметры отображения графика (цвет, масштаб и т. д.).
2. Реализация функционала построения графиков. Были написаны алгоритмы для вычисления значений функции на заданном интервале и построения графика на основе этих значений. Также была реализована возможность добавления нескольких графиков на одно поле.
3. Реализация удобного сохранения созданных внутри приложения проектов на устройство.
4. Тестирование и отладка. Были проведены тесты для проверки корректности работы приложения и исправлены обнаруженные ошибки и недочеты.

Пользователь может легко взаимодействовать с графиками, изменять параметры функций, масштабировать оси координат, а также сохранять полученные результаты. Это приложение предоставляет удобный инструмент для изучения, построения и анализа поведения математических функций, помогая пользователям лучше понять их свойства и характеристики. Возможность исследования поведения графиков функций является одной из ключевых функциональных возможностей приложения: пользователь может изменять параметры функций, задавать интервалы значений переменных, а также находить точки пересечения графиков и другие важные характеристики. Использование языка программирования Python и библиотеки PyQt позволило создать удобный и интуитивно понятный пользовательский интерфейс для работы с графиками.

Данное приложение для построения и исследования графиков математических функций сможет наглядно продемонстрировать процесс их построения и станет незаменимым инструментом для решения практических задач в различных областях. Это позволит пользователям более эффективно изучать и применять математические функции в обучении и повседневной жизни: графики функций позволяют наглядно представить их поведение и свойства, что помогает визуализировать и анализировать математические модели и решать разнообразные задачи. Однако, ручное построение и исследование графиков может быть трудоемким и требовать значительных усилий, а разработанное приложение облегчит и автоматизирует этот процесс.

Программа, написанная в рамках проекта, позволит упростить применение графического способа для решения задач и увеличить эффективность процесса работы с функциями и их построения.

В целом, проект успешно реализовал поставленную цель и может быть полезен в различных научных и инженерных областях. В результате выполнения данного проекта удалось создать полезный инструмент для изучения и визуализации математических функций, который может быть использован для решения разнообразных задач.

Список источников

1. Графическое решение текстовых задач (пособие для учителей и школьников) // Лицей ТГУ URL: <https://lyceum.tsu.ru/wp-content/uploads/2016/05/grafich.pdf> (дата обращения: 29.11.2023).
2. РЕШЕНИЕ ЗАДАЧ ИЗ ШКОЛЬНОГО КУРСА МАТЕМАТИКИ С ПОМОЩЬЮ ГРАФИКОВ ФУНКЦИЙ // Cyberleninka URL: <https://cyberleninka.ru/article/n/reshenie-zadach-iz-shkolnogo-kursa-matematiki-s-pomoschu-grafikov-funktsiy> (дата обращения: 29.11.2023).
3. Решение текстовых задач с помощью графиков линейной функции. Данилкин П.С. // IX Международный конкурс научно-исследовательских и творческих работ учащихся "Старт в науке" URL: <https://school-science.ru/9/7/44929> (дата обращения: 13.12.2023).
4. Embedding custom widgets from Qt Designer // Python GUIs URL: <https://www.pythonguis.com/tutorials/pyside6-embed-pyqtgraph-custom-widgets/#pyqtgraph> (дата обращения: 24.01.2024).
5. QT. Установка дополнительных компонентов. PyQTGraph // Яндекс Лицей URL: <https://lms.yandex.ru/courses/1054/groups/8942/lessons/5948/materials/17119#1> (дата обращения: 26.01.2024).
6. How to use pyqtgraph // pyqtgraph.readthedocs.io URL: https://pyqtgraph.readthedocs.io/en/latest/getting_started/how_to_use.html (дата обращения: 29.01.2024).
7. Exporting // pyqtgraph.readthedocs.io URL: https://pyqtgraph.readthedocs.io/en/latest/user_guide/exporting.html (дата обращения: 29.01.2024).

Приложение 1

1. Ссылка для скачивания Qt Designer - <https://build-system.fman.io/qt-designer-download>
2. Ссылка для скачивания Python - <https://www.python.org/downloads/>
3. Ссылка для скачивания PyCharm - <https://www.jetbrains.com/pycharm/download>
4. Ссылка на репозиторий проекта на GitHub <https://github.com/llliizzz/SCproject>



4. Ссылка для установки проекта (Яндекс.Диск) - https://disk.yandex.ru/d/oFZcAgtiOscL_A



Приложение 2

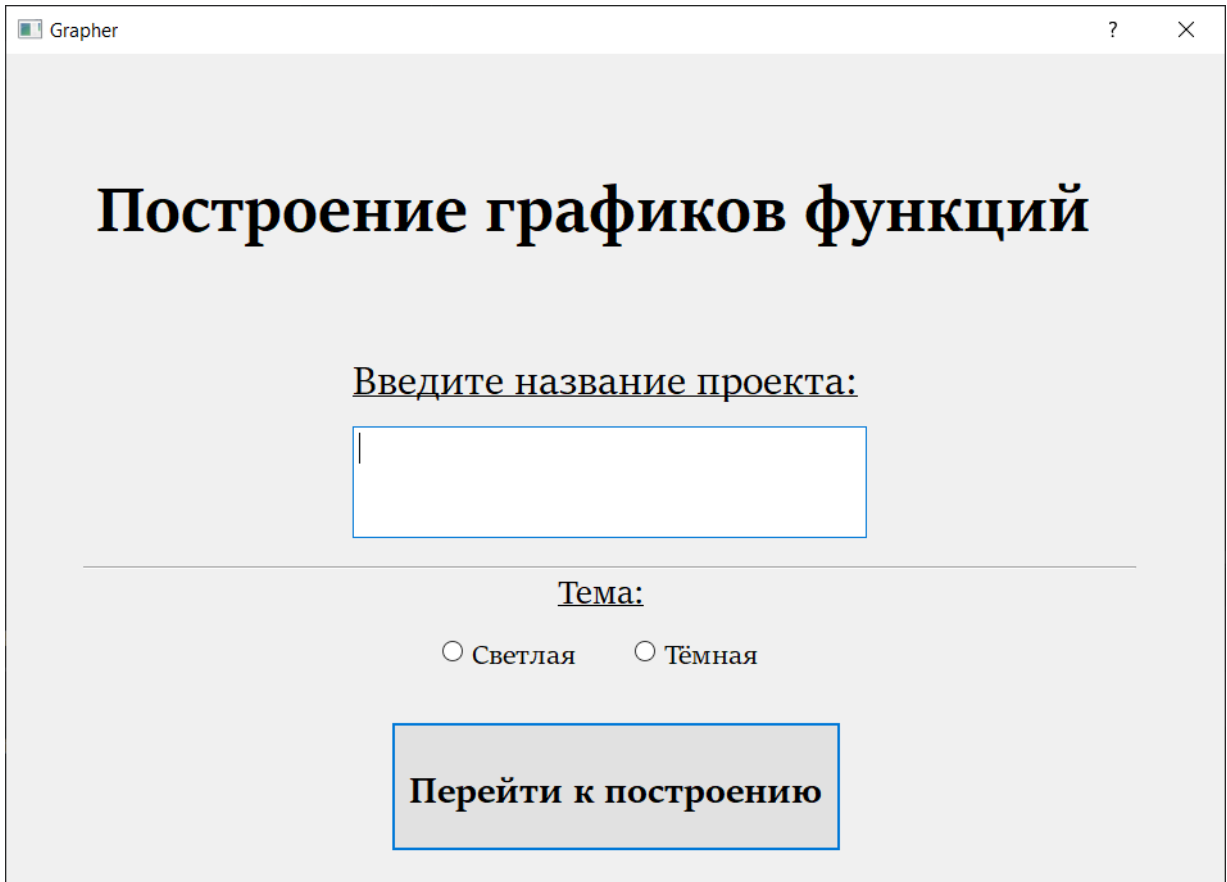
Руководство пользователя

1. Запуск приложения.

Убедитесь, что Python и все библиотеки, перечисленные в файле `requirements.txt` установлены на Вашем устройстве. Установить библиотеки можно через командную строку или PyCharm. (<https://pythonru.com/uroki/python-pip-uroki-po-python-dlja-nachinajushhih>)

Для запуска необходимо открыть и запустить файл `GRAPH.py`.

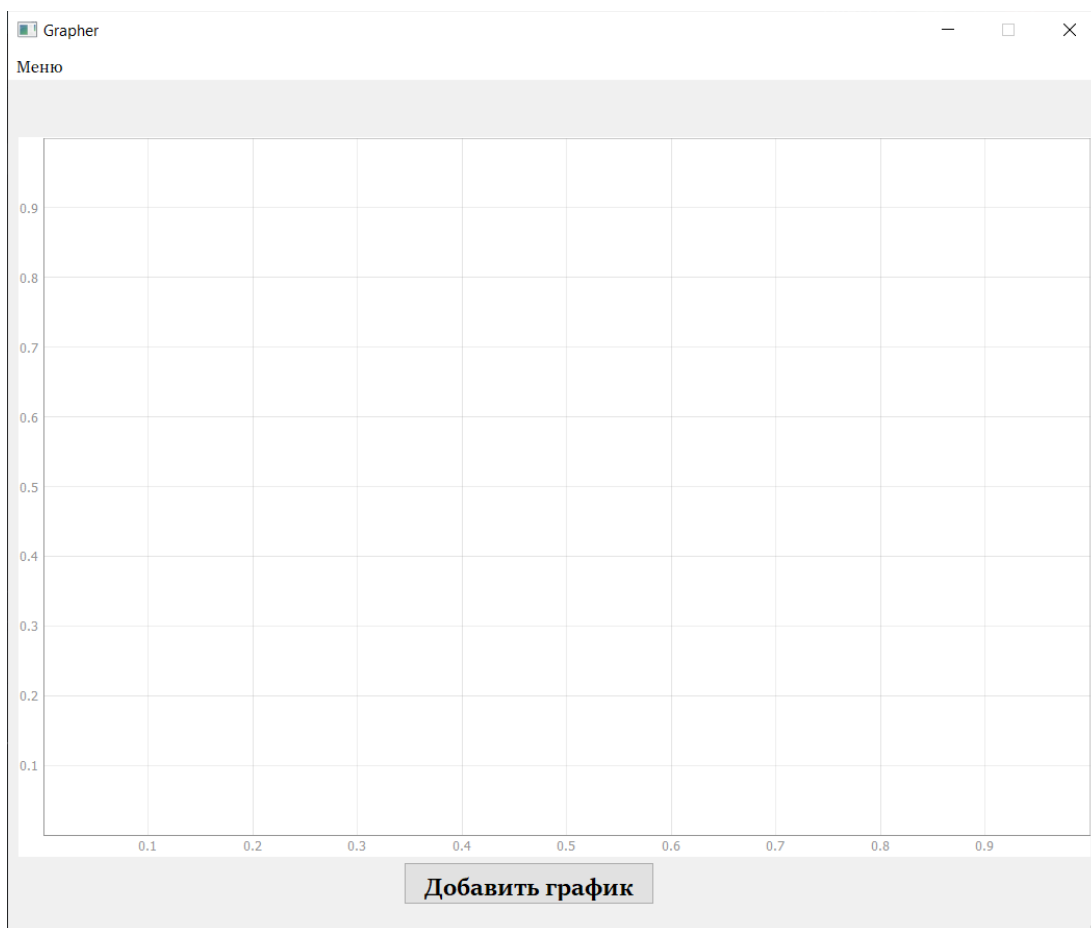
2. Стартовое окно.



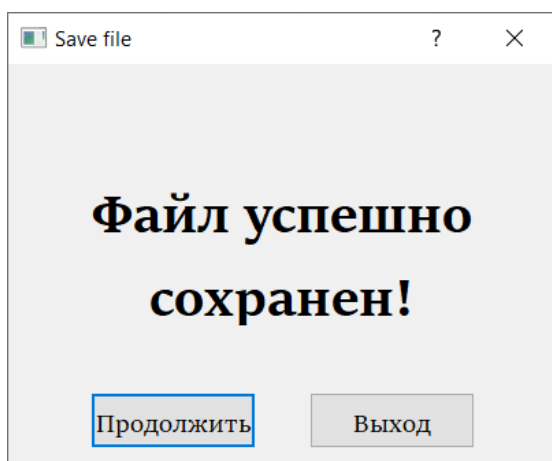
На данном окне пользователь может выбрать название для проекта, а также выбрать тёмную или светлую тему. Чтобы перейти к построению графиков необходимо нажать кнопку “Перейти к построению”.

3. Основное окно.

На этом окне находится поле для построения графиков, его размер и положение можно изменять с помощью мыши.



На верхней панели - меню. Если нажать в меню кнопку “*Сохранить*” (или сочетание клавиш “Ctrl+S”), откроется диалоговое окно для сохранения файла. Пользователь сам выбирает имя и расположение файла, по умолчанию рисунок сохраняется в папку с проектом.



Такое окно откроется если файл сохранен успешно.

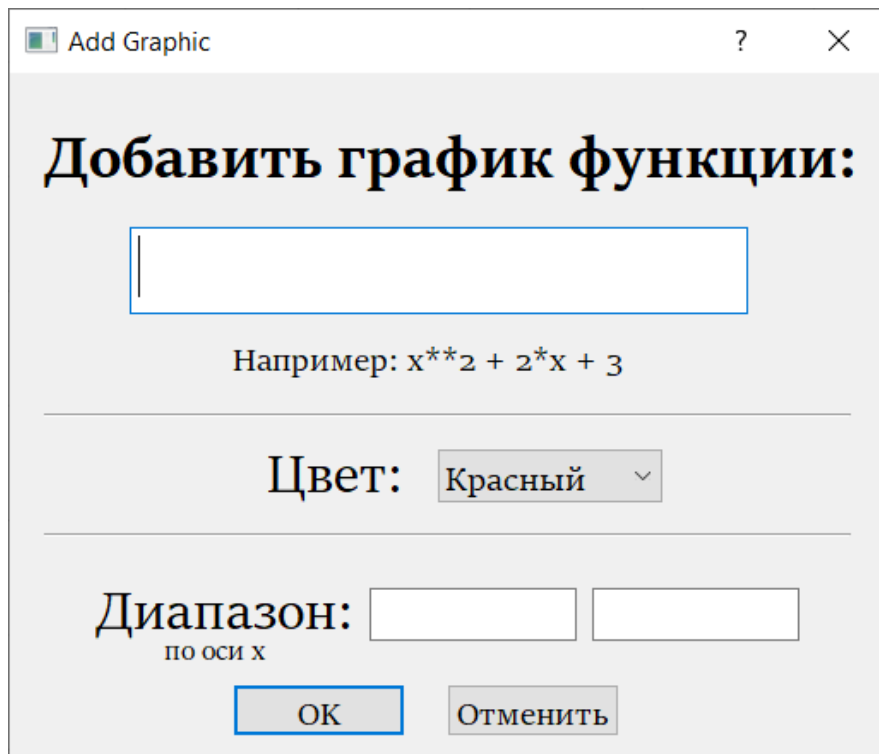
При нажатии на кнопку “*Продолжить*”, Вы вернетесь к работе с графиками. Кнопка “*Выход*” вернет Вас на начальное окно.

Если нажать в меню кнопку “*Помощь*” (или сочетание клавиш “Ctrl+H”), откроется этот файл в отдельном окне для просмотра.

Если нажать кнопку “Добавить график” откроется следующее окно:

4. Окно добавления графика.

На этом окне находится поле для записи функции для задания графика. **Обратите внимание, что при записи функции переменная должна называться именно ‘x’.**



Add Graphic

Добавить график функции:

Например: $x^2 + 2x + 3$

Цвет: Красный

Диапазон:
по оси x

ОК Отменить

Также пользователь может выбрать цвет графика из выпадающего списка и диапазон его координат по оси X (по оси Y - автоматически).

При нажатии на кнопку “ОК” график отобразится на основном поле.