РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ НЕФТИ И ГАЗА (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ИМЕНИ И.М. ГУБКИНА

**Кафедра информатики**

Расчётно-графическая работа №2 по дисциплине:

***«Основы алгоритмизации и программирования»***

Выполнила: студентка группы АС-23-05

Астраханкина Алёна Ярославовна

Проверил:

доцент, кандидат технических наук

Сидоров Валерий Васильевич

Москва, 2024г

Содержание:

[Введение 3](#_Toc167387219)

[§1. Использованные библиотеки 4](#_Toc167387220)

[§2. Описание кода 5](#_Toc167387221)

[§3. Внешний вид 9](#_Toc167387222)

[Заключение. 11](#_Toc167387223)

[Список использованных источников и литературы 12](#_Toc167387224)

[Список иллюстраций 13](#_Toc167387225)

# Введение

В современном мире, где технологии становятся все более важными, умение программировать и понимание основ алгоритмизации играют ключевую роль. Курс "Основы алгоритмизации и программирования" помогает студентам освоить основные принципы создания программ и разработки алгоритмов.

Цель данной расчётно-графической работы – научиться решать уравнения третьей степени комбинированным методом хорд и касательных, научиться находить приближенные значения интегралов по формуле «трех восьмых», используя двойные просчеты и научиться решать задачи Коши для уравнения первого порядка усовершенствованным методом ломаных на определенном интервале с каким-либо шагом, имея начальное условие х и у.

В работе будет проведен анализ существующих методик вычисления интегралов, уравнений третьей степени и решения задач Коши.

Ниже представлен список задач, которые необходимо выполнить для получения желаемого результата:

1. Ознакомиться с заданиями и выбрать библиотеки Python, с помощью которых будет написана программа
2. Изучить выбранные библиотеки и выделить команды, необходимые для написания кода
3. Написать код программы с использованием ранее изученной информации, проверить его на быстродействие и наличие ошибок

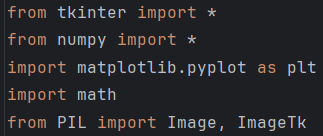
# §1. Использованные библиотеки

В процессе выполнения работы использовались следующие библиотеки:

1. Tkinter – для создания графического оформления игры
2. Math – для записи математических функций
3. PIL – для иллюстрации задания курсовой работы на экране
4. Numpy – для построения графика функции

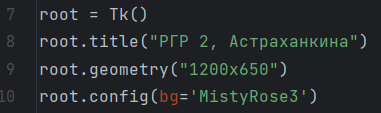
# §2. Описание кода

В данном фрагменте кода импортируются библиотеки, с помощью которых будет создаваться программа.



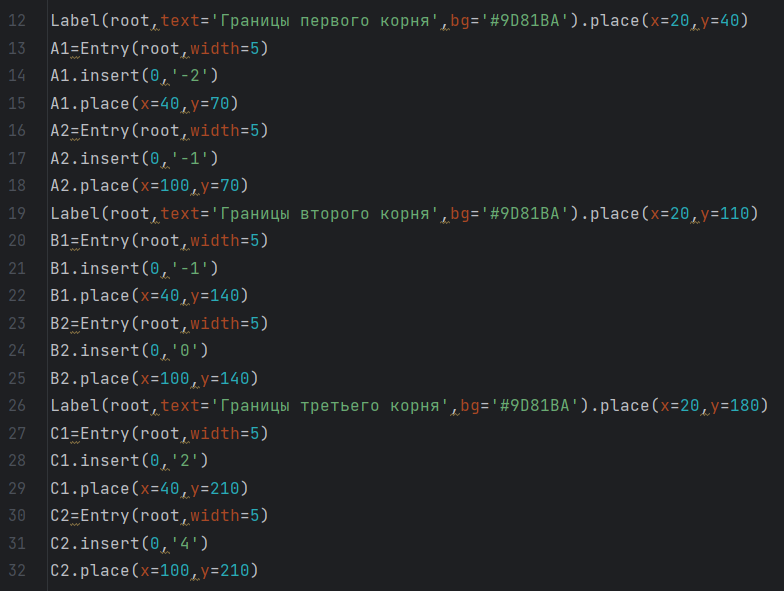
Рисунок

Далее я создаю окно, в котором будут находится ответы на задания.



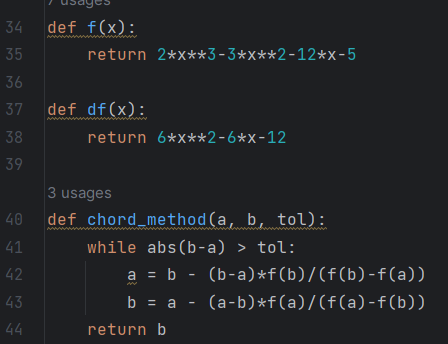
Рисунок

Создаю окошки, в которые вводятся границы корней уравнения.



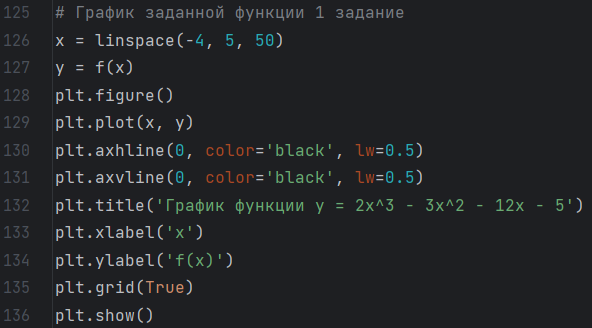
Рисунок

Ввожу необходимые функции для 1 задания и пишу в отдельных функциях код для вычисления кубического уравнения различными методами.



Рисунок

Пишу код для создания графика с заданной функции (требования самого задания)



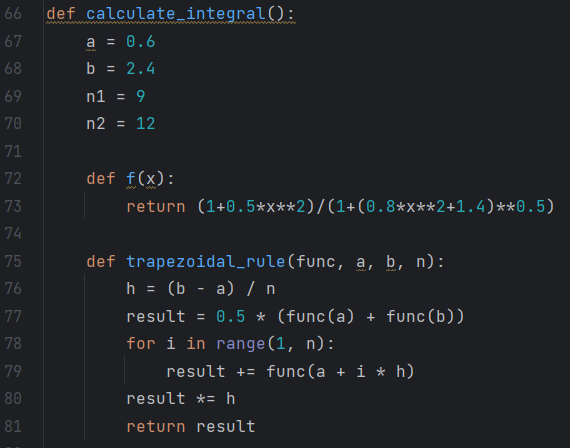
Рисунок

В отдельной функции ввожу переменные, в которых применяю функцию для вычисления уравнений, а затем помещаю найденные значения в окошко с ответами. Затем – добавляю фотографию графика функции, на котором можно посмотреть графическое изображение решение и сравнить его с найденными корнями.



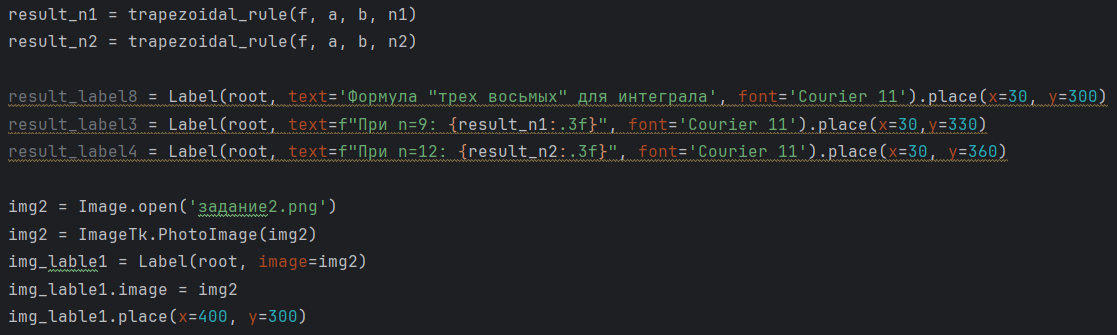
Рисунок

Создаю функцию, в которой прописываю метод «трех восьмых» для вычисления интеграла.



Рисунок

Ввожу переменные, в которых фиксирую получившиеся значения интеграла при помощи написанного ранее кода, помещаю ответы в окошко Tkinter и добавляю изображение с заданием для наглядности.



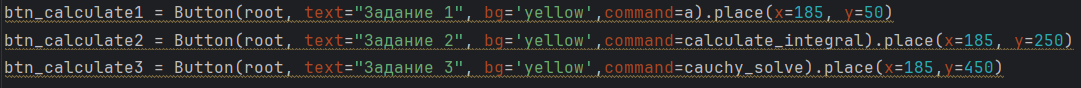
Рисунок

Создаю последнюю функцию для выполнения третьего задания методом «ломаных». Создаю массив, с которым проделываю некоторые действия, присущие заданному методу в цикле, а потом добавляю в новый массив эти переменные. В следующих двух строчках идет округление каждого элемента массива до 4 знаков после запятой (требование задание)



Рисунок

Прописываю код для создания кнопок, при помощи которых будут выводиться ответы на задания (одна кнопка – одно задание)



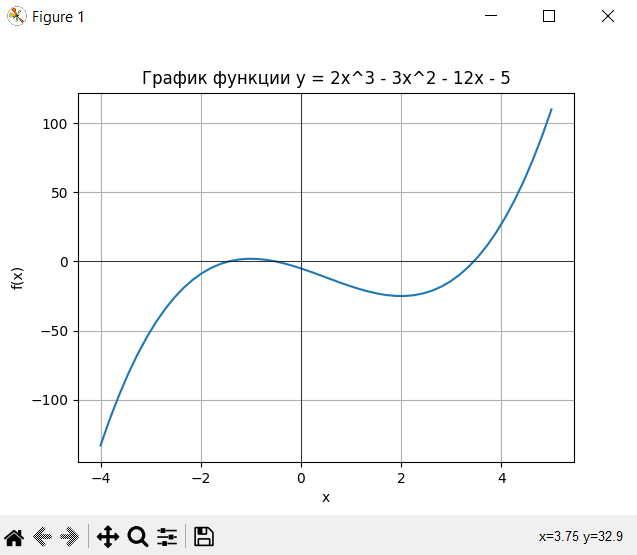
Рисунок

# §3. Внешний вид

После запуска программы выводится окно, в котором можно ввести границы корней для первого задания. Также выводится график функции для первого задания.

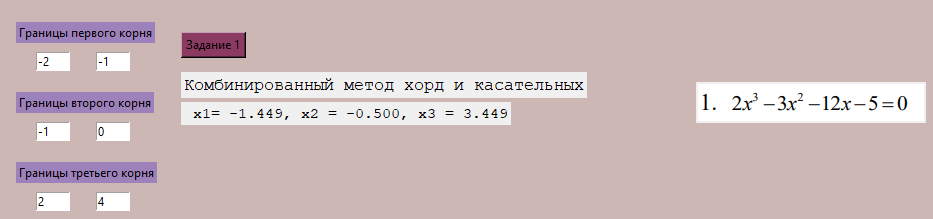


Рисунок



Рисунок

Вводим границы корней и нажимаем на кнопку «Задание 1». Выводятся корни и задание.



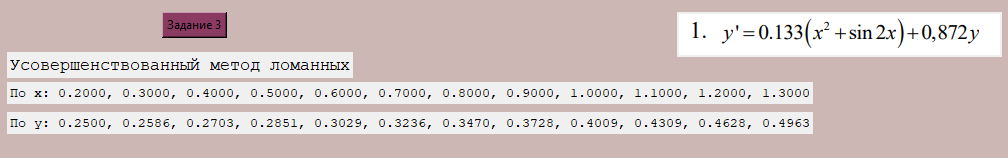
Рисунок

Задание 2:



Рисунок

Задание 3:



Рисунок

# Заключение.

В ходе выполнения данного домашнего задания я имела возможность расширить и закрепить свои знания и навыки по построению и реализации алгоритмов, а также созданию компьютерных программ для решения классических вычислительных задач. Работа над заданиями способствовала развитию логического мышления, умению анализировать проблемы и находить оптимальные решения.

Я познакомились с различными методами решения задач, научилась применять их на практике, что помогло мне улучшить свои навыки программирования и повысить уровень своей компетенции в области информационных технологий. Решение классических вычислительных задач требует не только технического мастерства, но и творческого подхода к поиску оптимальных решений.

# Список использованных источников и литературы

1. Библиотека Time в Python [Электронный ресурс] URL: <https://docs.python.org/3/library/time.html> (Дата обращения: 28.03.2024)
2. Библиотека Tkinter в Python [Электронный ресурс] URL: <https://docs.python.org/3/library/tkinter.html> (Дата обращения: 20.03.2024)
3. Библиотека PIL в Python [Электронный ресурс] URL: <https://python-scripts.com/pillow> (Дата обращения: 23.03.2024)

# Список иллюстраций

[Рисунок 1 5](#_Toc167387204)

[Рисунок 2 5](#_Toc167387205)

[Рисунок 3 5](#_Toc167387206)

[Рисунок 4 6](#_Toc167387207)

[Рисунок 5 6](#_Toc167387208)

[Рисунок 6 7](#_Toc167387209)

[Рисунок 7 7](#_Toc167387210)

[Рисунок 8 8](#_Toc167387211)

[Рисунок 9 8](#_Toc167387212)

[Рисунок 10 8](#_Toc167387213)

[Рисунок 11 9](#_Toc167387214)

[Рисунок 12 9](#_Toc167387215)

[Рисунок 13 10](#_Toc167387216)

[Рисунок 14 10](#_Toc167387217)

[Рисунок 15 10](#_Toc167387218)