Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«РАБОТА В ГРАФИЧЕСКОМ РЕЖИМЕ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Куликов Иван Алексеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров, 2022

**Цель работы:** освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.

**Задание (Вариант 12):**

1. Дополнить программу, реализованную в ходе предыдущей домашней контрольной работе, режимом визуализации.
2. Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
3. Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании.
4. Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация расчёта интеграла.

**Схема алгоритма**

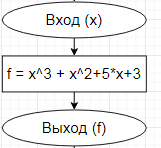
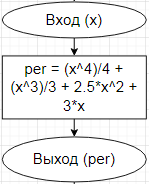


Рисунок 2 – Схема алгоритма подпрограммы perv

Рисунок 1 – Схема алгоритма подпрограммы fun

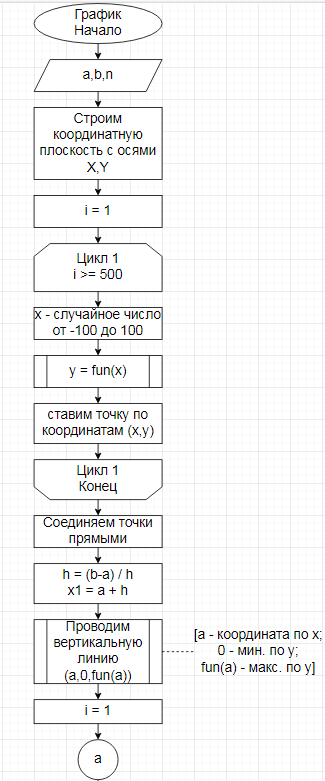
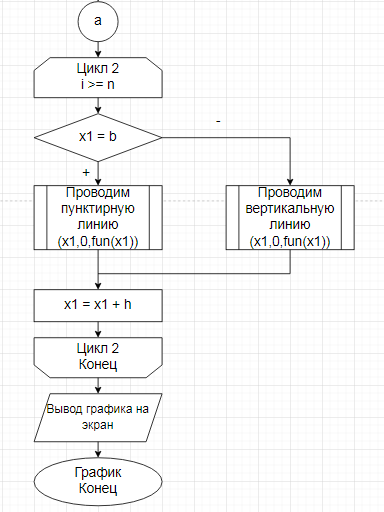


Рисунок 3 – Схема алгоритма подпрограммы grafic

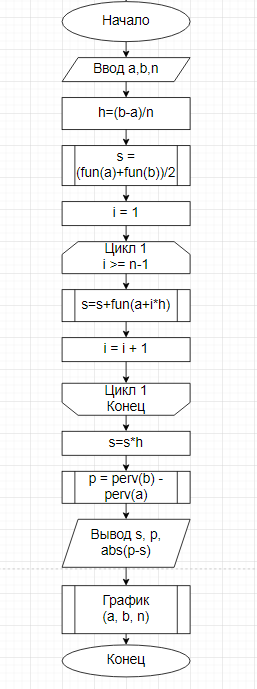


Рисунок 4 – Схема алгоритма основной программы

**Код программы:**

from tkinter import \*

from tkinter import messagebox

from math import \*

import matplotlib.pyplot as plt

import numpy as np

def function(x):

f = x\*\*3 + x\*\*2 + 5\*x + 3

return f

def pervoobraznaya(x):

per = (x\*\*4)/4 + (x\*\*3)/3 + 2.5\*x\*\*2 + 3\*x

return per

def integrel():

a = int(aa.get())

b = int(bb.get())

n = int(nn.get())

h = (b-a)/n

s = (function(a)+function(b))/2

x = a+h

for i in range(n-1):

s += function(x)

x += h

s \*= h

p = pervoobraznaya(b)-pervoobraznaya(a)

messagebox.showinfo('Результат', f"Точное значение = {p}")

def trap():

a = int(aa.get())

b = int(bb.get())

n = int(nn.get())

h = (b-a)/n

s = (function(a)+function(b))/2

x = a+h

for i in range(n-1):

s += function(x)

x += h

s \*= h

p = pervoobraznaya(b)-pervoobraznaya(a)

messagebox.showinfo('Результат', f"Приближённое значение (метод трапеций) ≈ {s}",

detail=f'Погрешность метода = {abs(p-s)}')

def grafic():

a = int(aa.get())

b = int(bb.get())

n = int(nn.get())

fig, ax = plt.subplots()

ax.set\_title('График функции: y=x^3+x^2+5\*x+3')

ax.grid(True)

ax.set\_xlim(-10, 10)

ax.set\_ylim(-10, 10)

ax.spines["left"].set\_position("center")

ax.spines["bottom"].set\_position("center")

ax.spines['top'].set\_visible(False)

ax.spines['right'].set\_visible(False)

ax.set\_aspect("equal")

x = np.linspace(-100, 100, 500)

y = function(x)

plt.ylabel("Ось X",labelpad = 110)

plt.xlabel("Ось Y",labelpad = 120)

h = (b-a)/n

x1 = a+h

ax.vlines(a,0,function(a),color = 'r')

ax.text(a-0.3,-0.5,'a',color ='r')

for i in range(n):

if x1==b:

ax.vlines(x1,0,function(x1),linestyle='dashed',color = 'r')

else:

ax.vlines(x1,0,function(x1),color = 'r')

x1+=h

ax.text(x1-h,-0.6,'b',color ='r')

ax.plot(x,y)

plt.show()

def ExitApp():

MsgBox = messagebox.askquestion(

'Выход из программы', 'Вы уверены, что хотите выйти?', icon='error')

if MsgBox == 'yes':

root.destroy()

else:

messagebox.showinfo('С возвращением!',

'Мы рады, что вы остались с нами!')

root = Tk()

root.title("Вычисление площади фигуры, ограниченной кривой")

root.geometry("675x450")

frame = Frame(root, padx=10, pady=10)

frame.pack(expand=True)

label = Label(frame, text='Задание (Вариант 12):', font=("Arial", 14))

label.pack()

label\_1 = Label(

frame, text='1. Реализовать программу вычисления площади фигуры, ограниченной кривой 1\*x^3+(1)\*x^2+(5)\*x+(3) и осью ОХ.')

label\_1.pack()

label\_2 = Label(

frame, text='2. Вычисление определенного интеграла должно выполняться численно, с применением метода трапеций')

label\_2.pack()

label\_3 = Label(

frame, text='3. Пределы интегрирования вводятся пользователем.')

label\_3.pack()

label\_4 = Label(

frame, text='4. Взаимодействие с пользователем должно осуществляться посредством case-меню.')

label\_4.pack()

label\_5 = Label(

frame, text='5. Требуется реализовать возможность оценки погрешности полученного результата.')

label\_5.pack()

label\_6 = Label(

frame, text='6. Необходимо использовать процедуры и функции там, где это целесообразно.')

label\_6.pack()

button = Button(frame, text="Понятно!", command=lambda: button.pack\_forget())

button.pack()

a = Label(frame, text="Введите точку а (a<b)")

a.pack()

aa = Entry(frame)

aa.pack()

b = Label(frame, text="Введите точку b")

b.pack()

bb = Entry(frame)

bb.pack()

n = Label(frame, text="Введите количество разбиений (n)")

n.pack()

nn = Entry(frame)

nn.pack()

btn = Button(frame, text='Расчитать точное значение', command=integrel)

btn.pack(fill=X)

butn = Button(frame, text='Вычислить методом трапеций', command=trap)

butn.pack(fill=X)

buttn = Button(frame, text='График', command=grafic)

buttn.pack(fill=X)

buttonEg = Button(frame, text='Выход', command=ExitApp)

buttonEg.pack(anchor=SE)

def motionUP(event):

children = frame.winfo\_children()

if event.widget in children:

index = children.index(event.widget)

index -= 1

if index > -1:

children[index].focus\_set()

def motionDOWN(event):

children = frame.winfo\_children()

if event.widget in children:

index = children.index(event.widget)

index += 1

if index < len(children):

children[index].focus\_set()

root.bind('<Up>', motionUP)

root.bind('<Down>', motionDOWN)

root.mainloop()

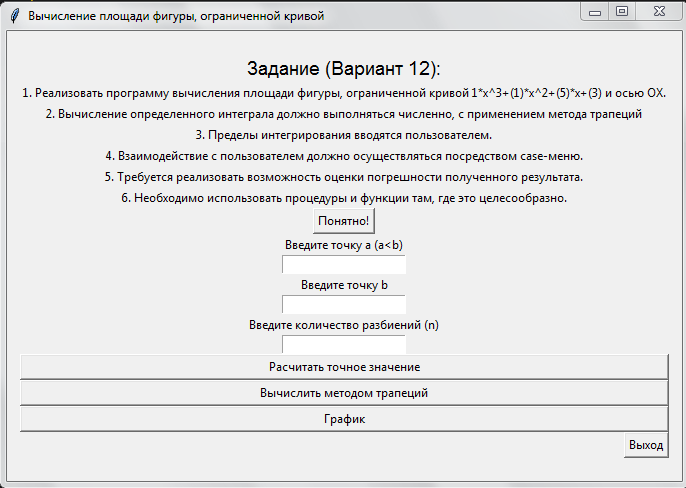
**Результат выполнения программы**

Рисунок 5 – Результат выполнения программы (Пользовательский интерфейс)

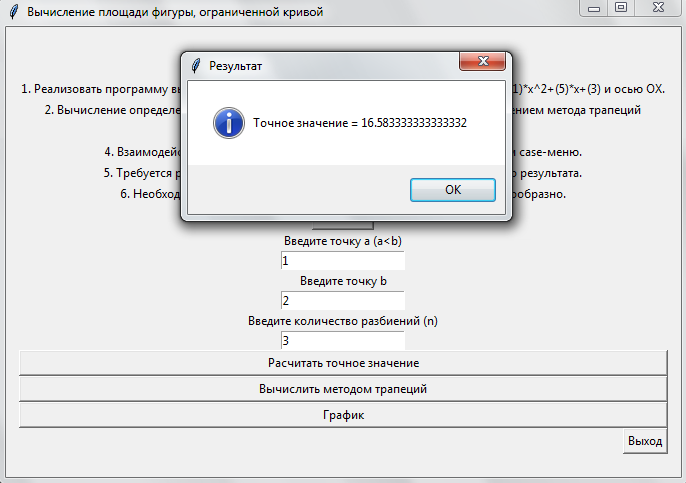
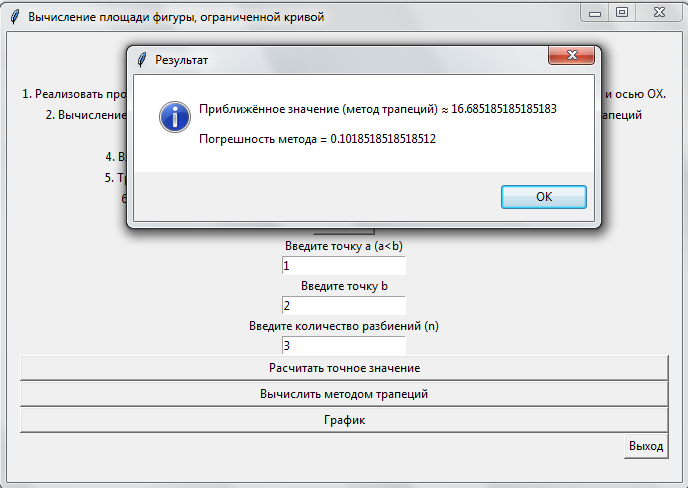


Рисунок 6 – Результат выполнения программы (Точное значение)

****Рисунок 7 – Результат выполнения программы (Метод трапеций)

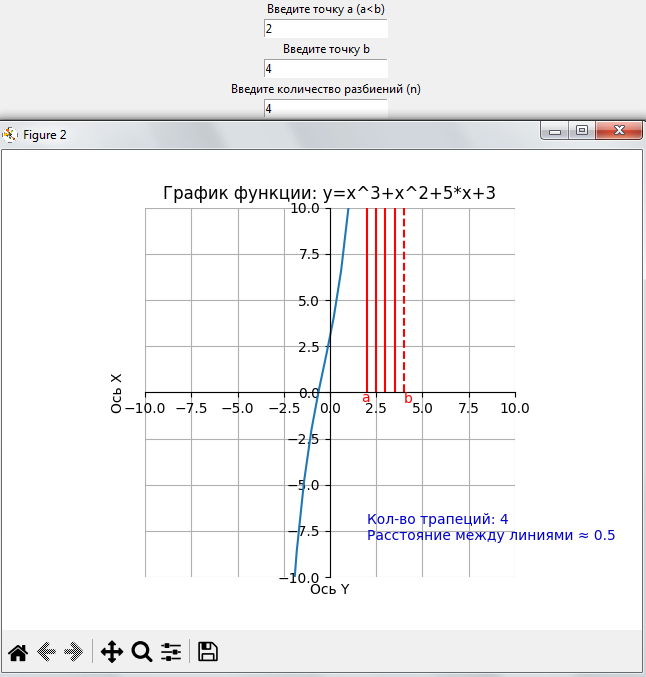


Рисунок 7 – Результат выполнения программы (График(Ввод 1))

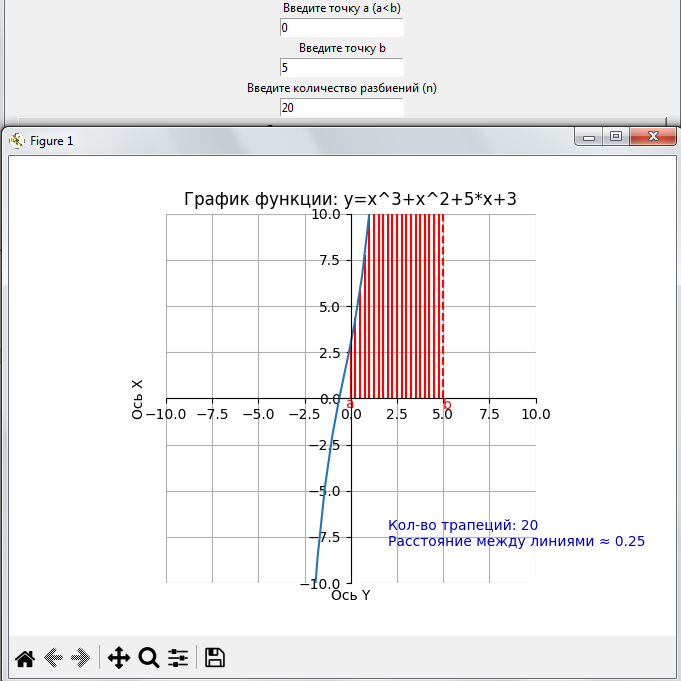


Рисунок 8 – Результат выполнения программы (График (Ввод 2))

****

Рисунок 22 – Элемент управления графиком №4 (Увеличение выделенной области)

Рисунок 10 – Элемент управления графиком №2 (Откат одного действия)

****

Рисунок 11 ­– Элемент управления графиком №3(Перемещение по осям)

Рисунок 9 – Элемент управления графиком №1 (Откат к первоначальному графику)

Рисунок 14 – Элемент управления графиком №6 (Сохранение графика в PNG)

Рисунок 13 ­– Элемент управления графиком №5 (Настройки расположения графика)

**Вывод:**

Исходя из общего положения и пользуясь в качестве довода и логического обоснования совокупность ранее упомянутых эмпирических знаний можно сделать закономерный вывод, что в ходе работы мы использовали знания, полученные нами на столь замечательной и бесподобной дисциплине, как Основы алгоритмизации и программирования. После домашней контрольной работы №4 мы можем смело сказать, что полученные на лекционных занятиях знания и умения в полной мере пригодились для грамотного и быстрого выполнения данной работы.

В ходе работы была изучена кроссплатформенная событийно-ориентированная графическая стандартная библиотека Python – Tkinter, благодаря которой был написан пользовательский интерфейс для программы, вычисляющей площадь фигуры, ограниченной кривой. Также были отработаны навыки работы с функциями и процедурами в языке программирования Python. Их использование было необходимо для избежание дублирования кода при многократном его использовании.

Для визуализации графика функции была использована библиотека языка программирования Python – Mathplotlib. Matplotlib – это комплексная библиотека для создания статических, анимированных и интерактивных визуализаций на Python. Использование именно этой библиотеки аргументированно тем, что только эта библиотека ЯП Python целиком и полностью предназначена для построения графиков различной сложности. Для работы с этой библиотекой было необходимо её установить с помощью pip. Во время установки библиотеки проблем не возникло.

Подводя итоги, можно смело утверждать, что выполнение домашней контрольной работы №4 помогло освоить принципы работы в графическом режиме, получить базовые навыки взаимодействия с графическими примитивами.