Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №3**

**«ИЗУЧЕНИЕ БАЗОВЫХ ПРИНЦИПОВ ОРГАНИЗАЦИИ ПРОЦЕДУР И ФУНКЦИЙ»**

**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 синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

**Задание (Вариант 12):**

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 1 \* x ^ 3 + (1) \* x ^ 2 + (5) \* x + (3) и осью ОХ (в положительной части по оси OY).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода трапеций.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством case-меню.
5. Требуется реализовать возможность оценки погрешности полученного результата.
6. Необходимо использовать процедуры и функции там, где это целесообразно.

**Схема алгоритма**

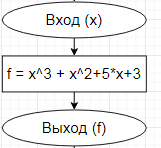
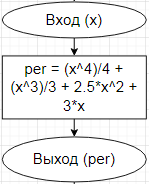


Рисунок 2 – Схема алгоритма подпрограммы perv

Рисунок 1 – Схема алгоритма подпрограммы fun

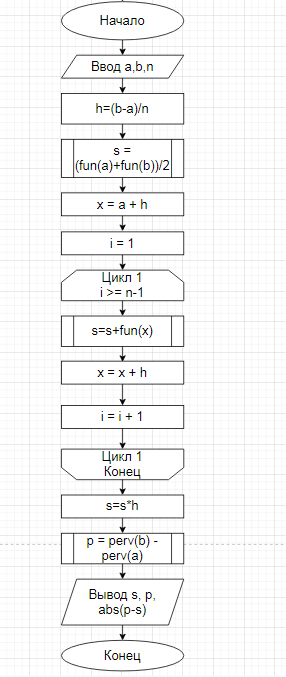


Рисунок 3 – Схема алгоритма основной программы

**Код программы:**

from tkinter import \*

from tkinter import ttk

from tkinter import messagebox

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'Приближённое значение (м-д трапеций) ≈ {s}',

detail=f"Точное значение = {p} \n Погрешность = {abs(p-s)} ",)

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("670x400")

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',command = integrel)

btn.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**

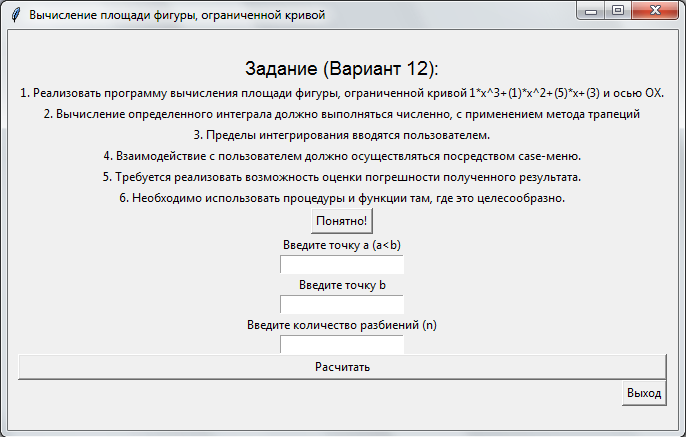
****

Рисунок 4 – Результат выполнения программы (Пользовательский интерфейс)

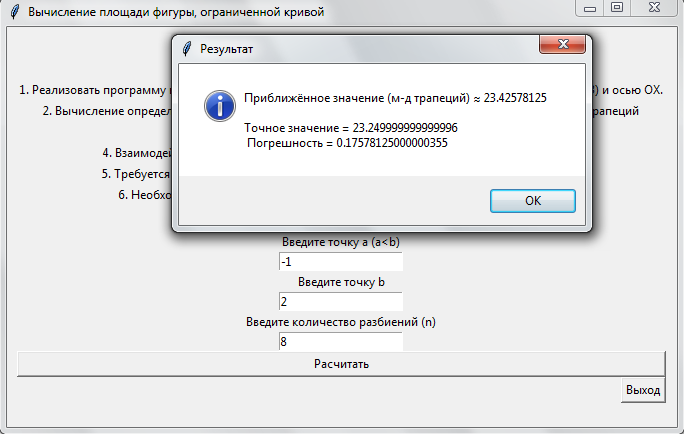
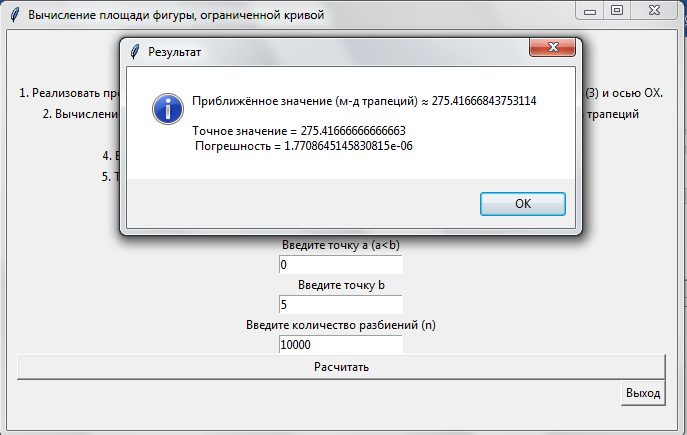


Рисунок 5 – Результат выполнения программы (Ввод данных 1)

****Рисунок 6 – Результат выполнения программы (Ввод данных 2)

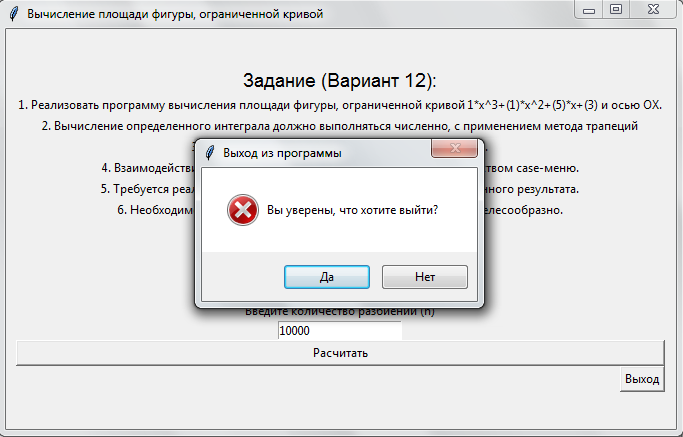


Рисунок 7 – Результат выполнения программы (Нажатие на кнопку "Выход")

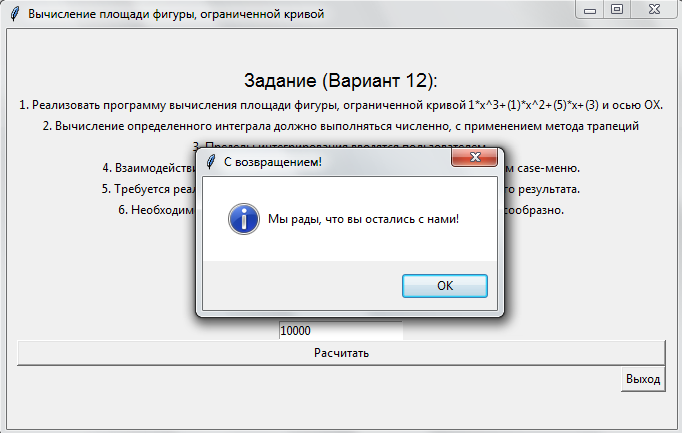


Рисунок 8 – Результат выполнения программы (При нажатии на кнопку "Нет")

**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 №3 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.

Подводя итоги, можно смело утверждать, что выполнение домашней контрольной работы №3 помогло освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.