Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический университет имени

В.Ф. Уткина»

Кафедра «Вычислительной и прикладной математики»

Отчёт по лабораторной работе №16

**Модули**

по дисциплине

“Алгоритмические языки и программирование”

|  |
| --- |
|  |

**Выполнили:**

ст. гр. 245

Луковкин Иван

**Проверил:**

профессор кафедры ВПМ

Пылькин А.Н.

Рязань 2023

**Цель работы:** Научиться создавать собственные модули и подключать их в программу; объединять в пакеты набор модулей, посвященных решению задач из некоторой предметной области.

**Задание:**

**Вариант 15.**

Реализовать пакет для приближенного вычисления значения определенного интеграла функции f(x) на отрезке от a до b с использованием: 1) формулы прямоугольников; 2) формулы трапеций; 3) формулы Симпсона. При выборе вида функции f(x) можно использовать меню или встроенную функцию eval() для динамического исполнения выражений из ввода.

Схема алгоритма основной программы представлена на рисунке 1.



Рисунок 1- Схема алгоритма программы

Схемы алгоритмов элементов модулей представлены в приложении А.

**Текст программы-модуля:**

def rectangular\_rule(f, a, b, n):

dx = (b - a) / n

x\_values = [(a + i \* dx) for i in range(n)]

areas = [f((x\_values[i] + x\_values[i+1]) / 2) \* dx for i in range(n-1)]

return sum(areas)

def trapezoid\_rule(f, a, b, n):

dx = (b - a) / n

x\_values = [(a + i \* dx) for i in range(n+1)]

areas = [(f(float(x\_values[i])) + f(x\_values[i+1])) \* dx / 2 for i in range(n)]

return sum(areas)

def simpson\_rule(f, a, b, n):

if n % 2 != 0:

n += 1

dx = (b - a) / n

x\_values = [(a + i \* dx) for i in range(n+1)]

areas = [(f(float(x\_values[i])) + 4\*f((x\_values[i] + x\_values[i+1])/2) + f(x\_values[i+1])) \*

(dx/6) for i in range(0, n, 1)]

return sum(areas)

**Текст основной программы:**

from lab16 import \*

from math import \*

def main():

func = str(input("Введите функцию f(x): "))

a = float(eval(input("Введите начало отрезка a: ")))

b = float(eval(input("Введите конец отрезка b: ")))

n = 4

f = lambda x: eval(func)

result\_rectangular = rectangular\_rule(f, a, b, 10 \*\* n)

result\_trapezoid = trapezoid\_rule(f, a, b, n)

result\_simpson = simpson\_rule(f, a, b, n)

print(

"Результаты вычисления:"

"\nФормула прямоугольников: ", round(result\_rectangular, 5),

"\nФормула трапеций: ", round(result\_trapezoid, 5),

"\nФормула Симпсона: ", round(result\_simpson, 5)

)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Результаты:**

Введите функцию f(x): x\*\*2 + 3\*x - 5

Введите начало отрезка a: 0

Введите конец отрезка b: 10

Результаты вычисления:

Формула прямоугольников: 433.20834

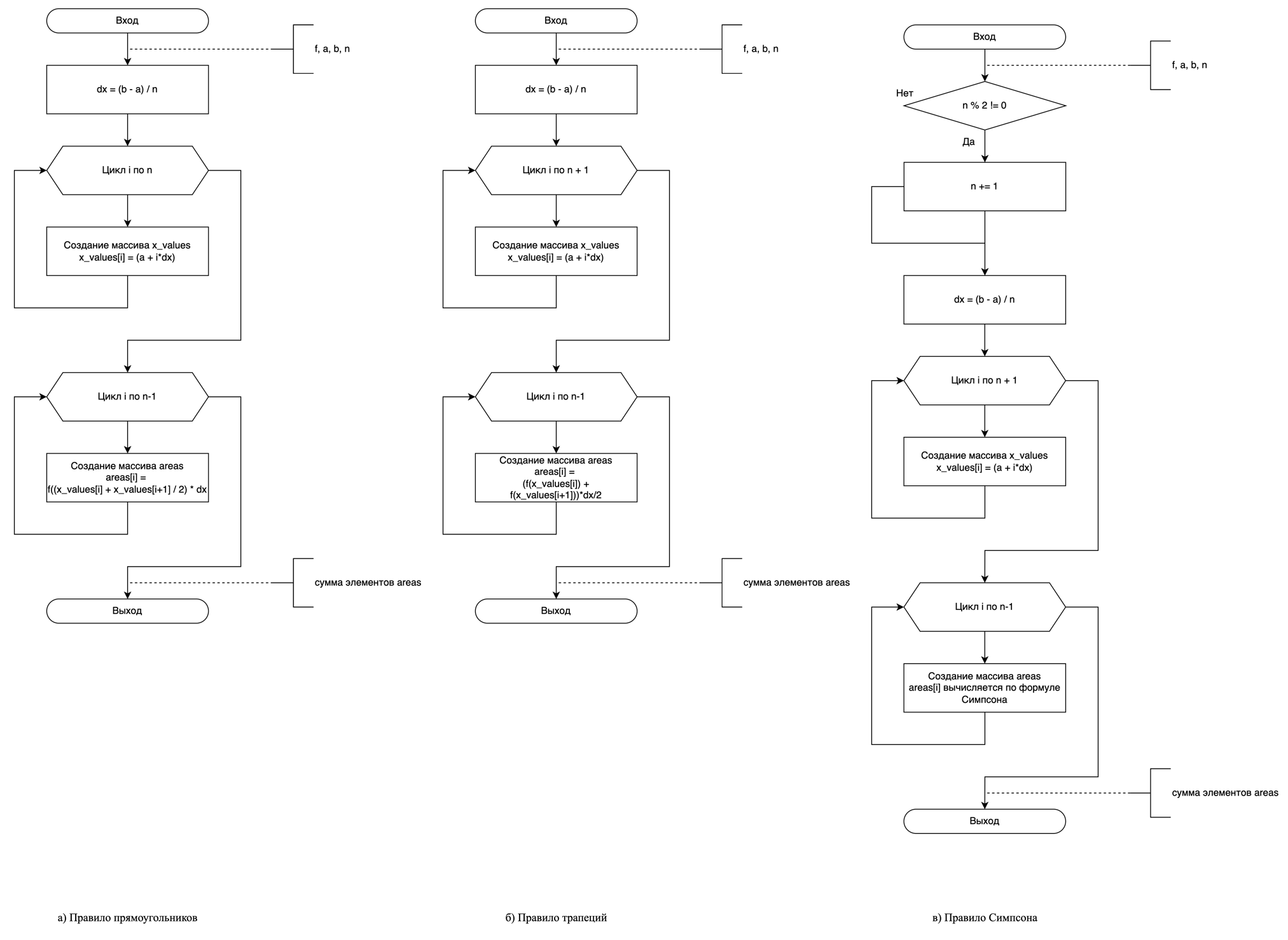
Формула трапеций: 443.75

Формула Симпсона: 433.33333

**Заключение**

Программа работает корректно и без выявления ошибок.

**Приложение А. Схемы алгоритмов элементов модуля**

****