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-202-52-00

Холин Павел Эдуардович

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

**Оглавление**

[**Цель работы** 3](#_Toc147786659)

[**Формулировка задания (с вариантом)** 3](#_Toc147786660)

[**Описание алгоритма** 3](#_Toc147786661)

[**Схема алгоритма с комментариями** 3](#_Toc147786662)

[**Код программы** 5](#_Toc147786663)

[**Результат выполнения программы** 6](#_Toc147786664)

[**Вывод** 7](#_Toc147786665)

# **Цель работы**

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

# **Формулировка задания (с вариантом)**

Вариант: 22

1. Реализовать программу вычисления пощади фигуры, ограниченной кривой 2\*x^3+2\*x^2 – 2\*x +19 и осью OX (в положительной части 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.

# **Описание алгоритма**

1. Функция Integral(x: real): real; вычисляет значение функции 2x^3 + 2x^2 - 2x + 19 и возвращает результат.

2. Функция Trap(a, b: real; n: integer): real; рассчитывает интеграл методом трапеций. Внутри функции определены переменные h, sum, x, i для вычисления интеграла.

3. Функция perv(x: real): real; вычисляет производную исходной функции, и возвращает результат.

4. Функция Newton(a, b: real): real; вычисляет значение интеграла по формуле Ньютона-Лейбница.

5. Функция errorCalculation(result, Newton: real): real; вычисляет абсолютную погрешность вычисленного интеграла.

6. Процедура DisplayMenu() выводит в консоль меню программы.

7. Выполнение основной части программы:

- Пользователю предлагается выбор: вычислить интеграл или выйти из программы.

- При выборе расчета интеграла вводятся нижний предел интегрирования (a), верхний предел интегрирования (b) и количество разбиений (n).

- После ввода данных выполняется вычисление интеграла методом трапеций и вывод результата.

- Далее выводится значение интеграла по формуле Ньютона-Лейбница и абсолютная погрешность.

8. Повторение (цикл) вывода меню до тех пор, пока не будет выбран выход из программы.

**Схема алгоритма с комментариями**

# 

Рис. 1 Функция Integral

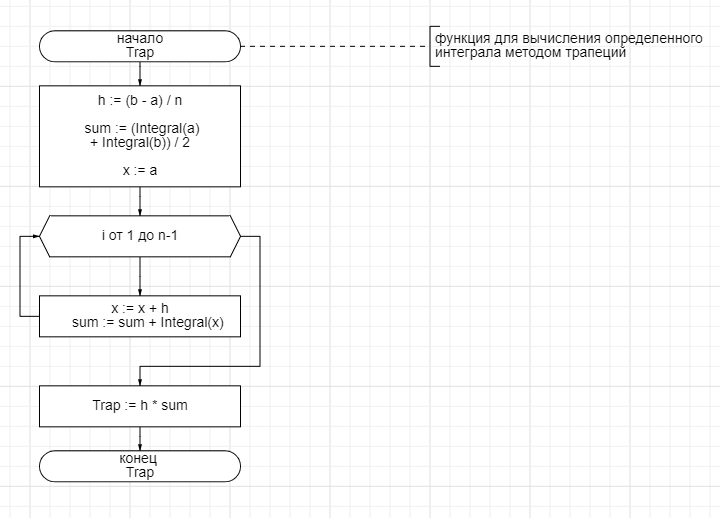


Рис. 2 Функция Trap

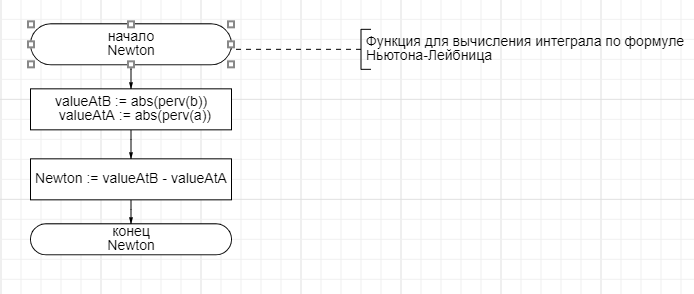


Рис. 3 Функция Newton

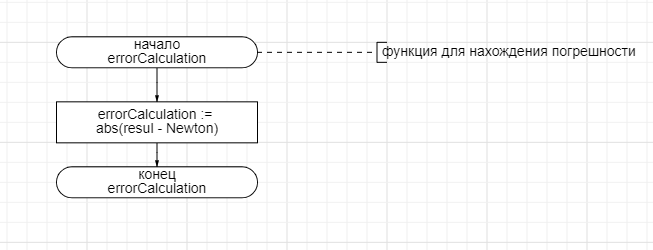


Рис. 4 Функция errorCalculation

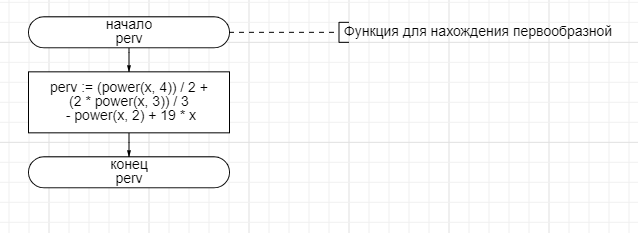


Рис. 5 Функция perv

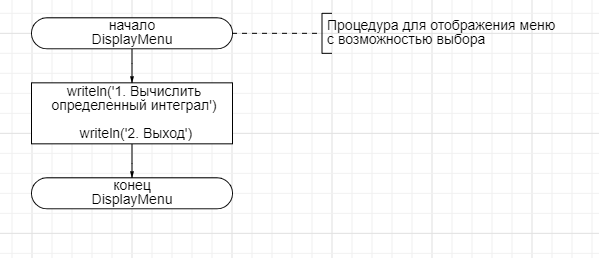


Рис. 6 Процедура DisplayMenu

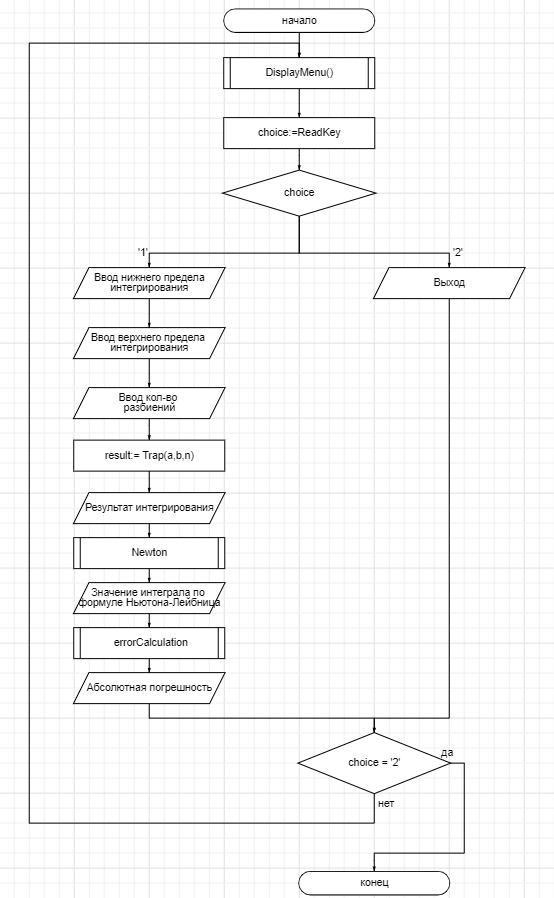
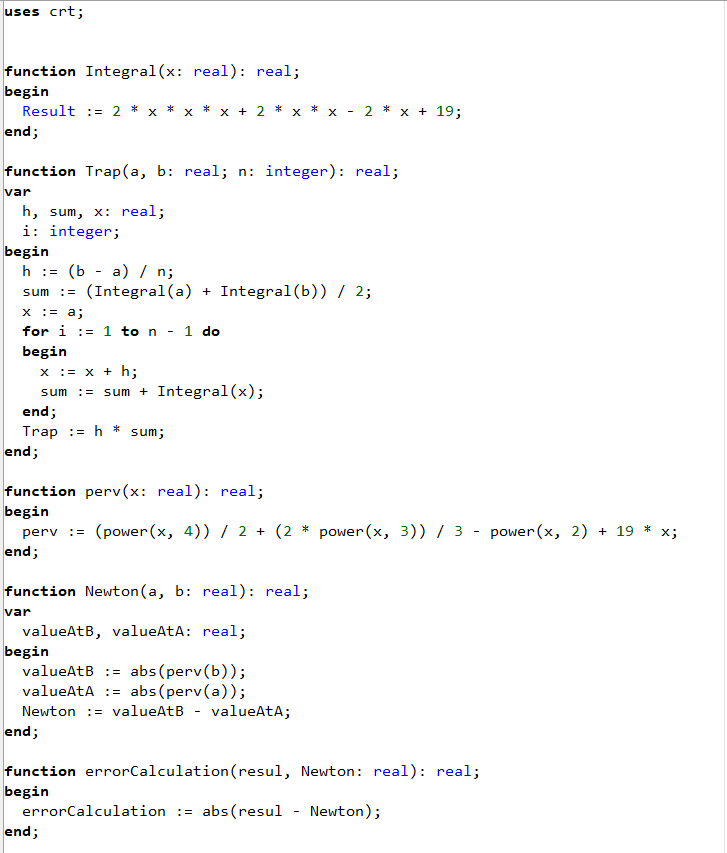
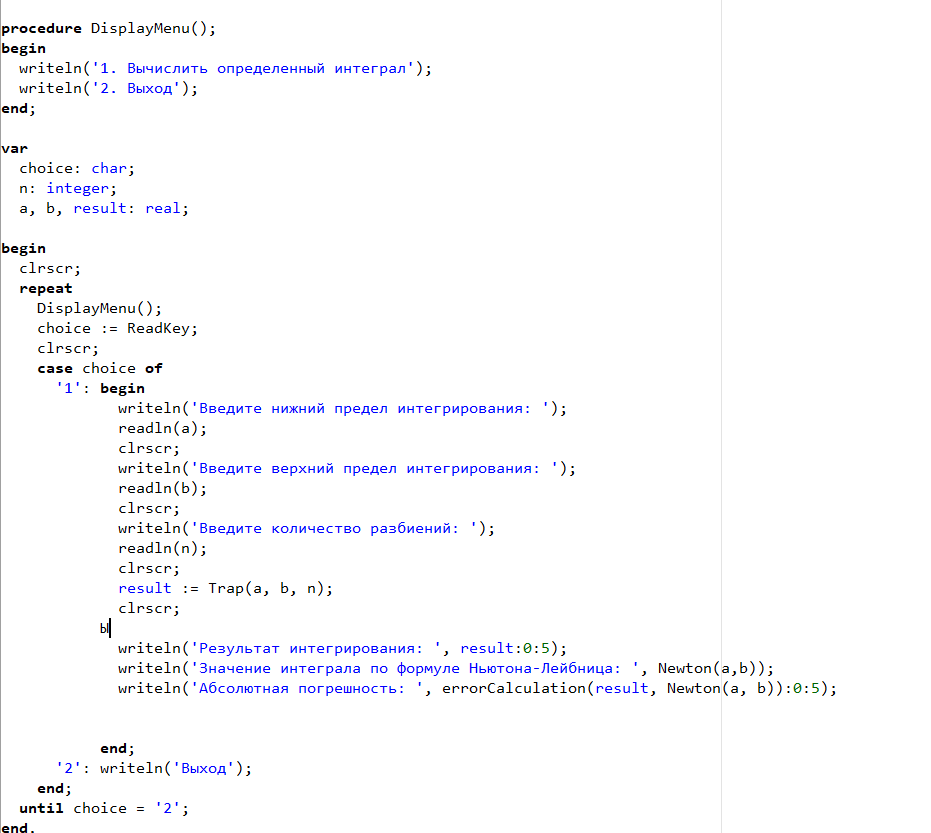


Рис. 7 Тело программы

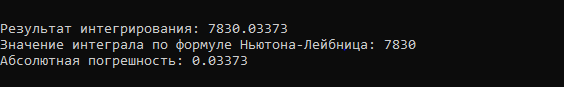
# **Код программы**





# 

# **Результат выполнения программы**



# 

# **Вывод**

Программа была успешно исправлена и доработана с учетом всех поставленных требований:

1) Была реализована программа для вычисления площади фигуры, ограниченной кривой 2\*x^3+2\*x^2 – 2\*x +19 и осью OX в положительной части оси OY.

2) Для вычисления определенного интеграла был использован численный метод трапеций.

3) Пределы интегрирования, количество разбиений и оценка погрешности были введены пользователем.

4) Реализовано взаимодействие с пользователем через case-меню с помощью CRT.

5) Была добавлена возможность оценки погрешности полученного результата.

6) Для повышения читаемости и структурированности кода функции и процедуры использовались там, где это целесообразно.

Таким образом, программа была успешно адаптирована для выполнения всех поставленных задач.

После успешного освоения синтаксиса построения процедур и функций, изучения способов передачи данных в подпрограммы и получения навыков организации минимального пользовательского интерфейса, я уверенно могу создавать более сложные и структурированные программы. Это помогло мне лучше понять принципы модульного программирования и улучшило мою способность разрабатывать более эффективные и читаемые программы.