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»**

Выполнила: студент учебной группы

ИСПк-202-52-00

Савватеев Аркадий Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

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

**1.Формулировка задания 1. Вариант 19**

**Постановка задачи**

Реализовать программу вычисления площади фигуры, ограниченной кривой 1\*x^3 + (0) \* x^2 + (1)\*x + (12) и осью OX (в положительной части по оси OY)

**2.Описание алгоритма**

1. Определение функции: пишем функцию, которую хотим интегрировать, то есть данную нам в условии задачи

2. Функция с методом трапеций: реализуем основной алгоритм для вычисления интеграла с помощью нахождения высоты трапеции по формуле и цикла for для подсчета их количества.

3. Case-меню: создаём функцию, которая запрашивает данные у пользователя и в зависимости от его выбора выводит разный результат.

4. Точный интеграл: добавляем функцию для вычисления точного значения интеграла путем вычитания из первообразной значение функции по примененному методу трапеций.

5. Вывод значений: в зависимости от выбора пользователя в предоставленном case-меню, вывести результат в соответствии с ним, или же закончить программу, если пользователь нажал «0» с помощью функции Untill.

**3.Схема алгоритма**

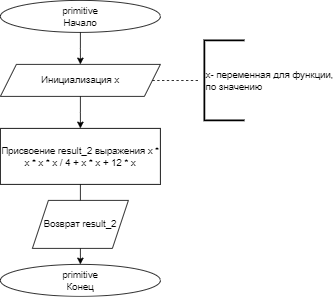
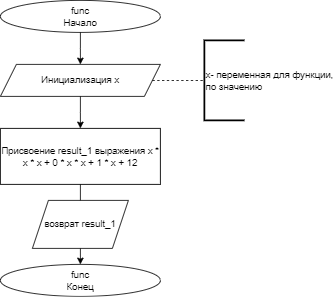
****

Рисунок 1.1 – схема алгоритма

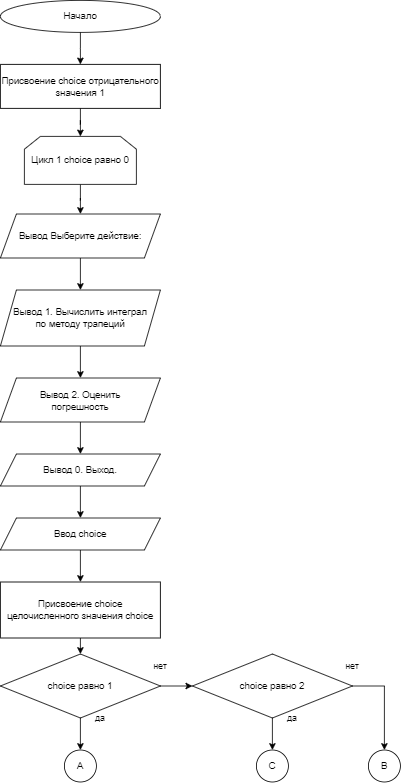


Рисунок 1.2 – схема алгоритма

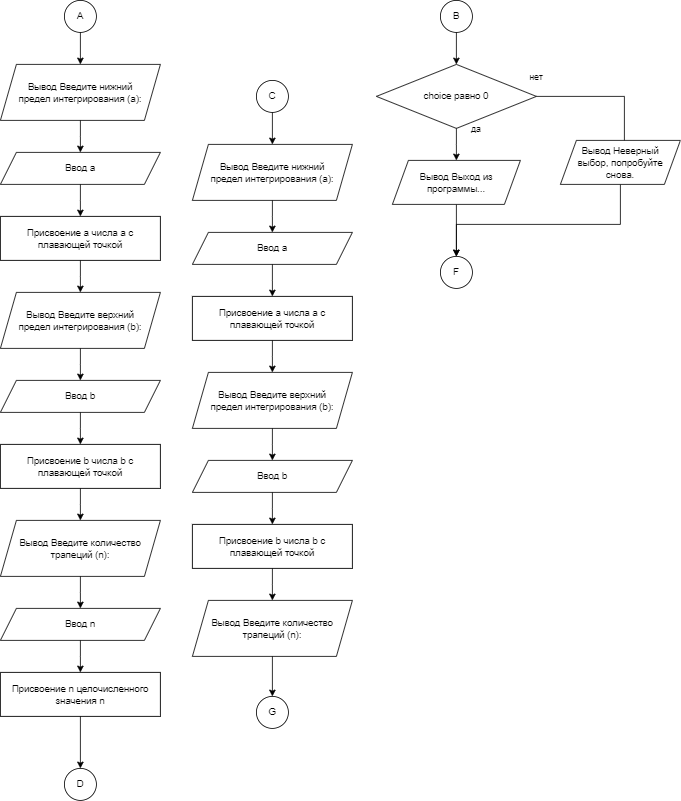


Рисунок 1.3 – схема алгоритма

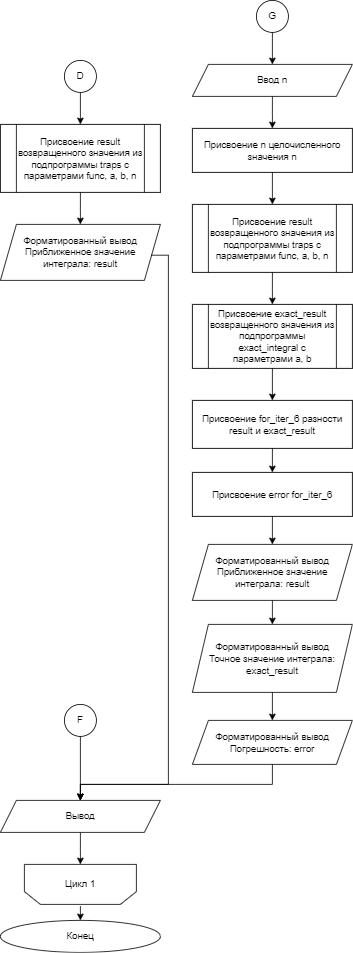


Рисунок 1.4 – схема алгоритма

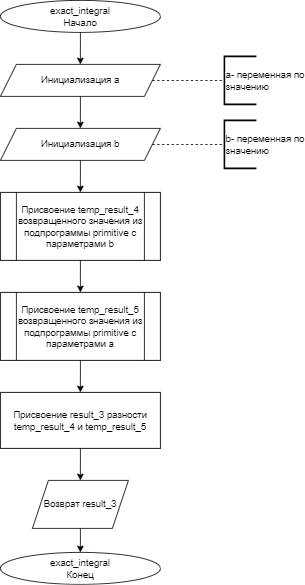


Рисунок 1.5 – схема алгоритма

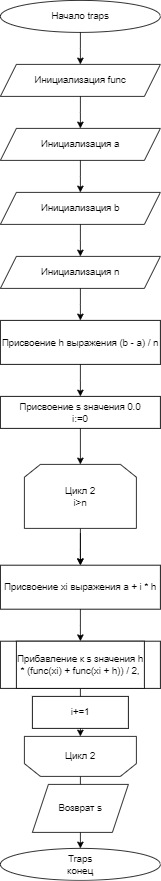
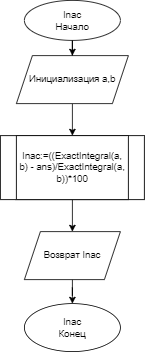


Рисунок 1.6 – схема алгоритма

**4.Код программы**

**program** qq;

**type**

FuncType = **function**(x: Real): Real;

**function** Func(x: Real): Real;

**begin**

Func := x \* x \* x + 1 \* x + 12;

**end**;

**function** Primitive(x: Real): Real;

**begin**

Primitive := (x \* x \* x \* x) / 4 + (x \* x) / 2 + 12 \* x;

**end**;

**function** ExactIntegral(a, b: Real): Real;

**begin**

ExactIntegral := Primitive(b) - Primitive(a);

**end**;

**function** Traps(Func: FuncType; a, b: Real; n: Integer): Real;

**var**

h, s, xi: Real;

i: Integer;

**begin**

h := (b - a) / n;

s := 0.0;

**for** i := 0 **to** n - 1 **do**

**begin**

xi := a + i \* h;

s := s + h \* (Func(xi) + Func(xi + h)) / 2;

**end**;

Traps := s;

**end**;

**function** Inac(a, b, ans: real): real;

**begin**

Inac:=((ExactIntegral(a, b) - ans)/ExactIntegral(a, b))\*100

**end**;

**var**

a, b, exactResult, error: Real;

result: Real;

n: Integer;

choice: Integer;

**begin**

**repeat**

Writeln('Выберите действие:');

Writeln('1. Вычислить интеграл по методу трапеций');

Writeln('2. Оценить погрешность');

Writeln('0. Выход.');

Readln(choice);

**case** choice **of**

1: **begin**

Write('Введите нижний предел интегрирования (a): ');

Readln(a);

Write('Введите верхний предел интегрирования (b): ');

Readln(b);

Write('Введите количество трапеций (n): ');

Readln(n);

result := Traps(Func, a, b, n);

Writeln('Приближенное значение интеграла: ', result);

**end**;

2: **begin**

Write('Введите нижний предел интегрирования (a): ');

Readln(a);

Write('Введите верхний предел интегрирования (b): ');

Readln(b);

Write('Введите количество трапеций (n): ');

Readln(n);

result := Traps(Func, a, b, n);

exactResult := ExactIntegral(a, b);

error := Abs(Inac(a, b, result));

Writeln('Приближенное значение интеграла: ', result);

Writeln('Точное значение интеграла: ', exactResult);

Writeln('Погрешность: ', error:0:8);

**end**;

0: Writeln('Выход из программы...');

**else**

Writeln('Неверный выбор, попробуйте снова.');

**end**;

Writeln;

**until** choice = 0;

**end**.

**5.Результат выполнения программы**

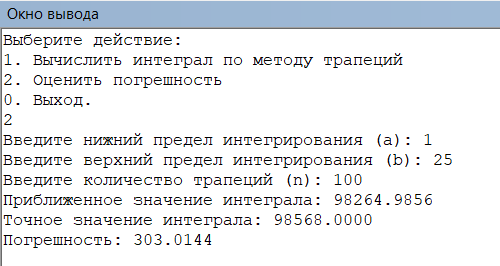
****

Рисунок 2 – примеры вводимых значений

1. **Вывод**

В ходе работы мы освоили синтаксис построения процедур и функций, изучили способы передачи данных в подпрограммы, получили навыки организации минимального пользовательского интерфейса.

Для создания алгоритмических схем было изучено правильное построение письменного алгоритма для достижения верного решения задачи, а также и само построение схем в программе «draw.io».

Однако в ходе работы мы столкнулись и с проблемами. Например, при создании схемы алгоритмов было необходимо вписать несколько функций в основной алгоритм, что представлялось неясным. Решением данной проблемы стало написание для функций отдельных схем алгоритмов как подпрограмм, который в конечном итоге использовались и в основном алгоритме.

В конечном итоге при помощи полученных знаний и исправленных ошибок у нас получилось добиться поставленной цели – решить задачу при помощи кода с условиями и циклами, а также узнать о языке программирования Pascal новые знания.