Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №2**

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

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

Выполнил: студент учебной группы

ИСПк-202-52-00

Широков Максим Сергеевич

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

1. **Цель работы**: освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.
2. **Вариант: 24**

**Задание:**

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 2\*x^3+(-1)\*x^2+(1)\*x+(11) и осью OX(в положительной части по оси OY).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода средних прямоугольников
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.
7. **Описание алгоритма:**

У пользователя, посредством case-меню, запрашивается желаемое действие. Если пользователь выбирает действие 2 (выйти из программы) – программа останавливается. Если пользователь выбирает действие 1 (рассчитать площадь фигуры), у него запрашиваются на ввод значения: пределы интегрирования и количество отрезков координатной прямой для расчёта площади фигуры, ограниченной некоторой функцией и осью ОХ. После происходит обращение к функции MidpointRectanglesIntegration, на вход которой передаются введенные пользователем значения. Эта функция обращается к функции F, которая вычисляет значение F=2\*x^3+(-1)\*x^2+(1)\*x+(11), и вычисляет площадь заданной фигуры при помощи метода средних прямоугольников. На экран выводится полученная площадь. Далее пользователь выбирает, нужно ли произвести оценку погрешности полученного результата. Если пользователь указал необходимость проведения оценки погрешности, программа вычисляет погрешность как разницу между исходной площадью и площадью при удвоенном количестве отрезков, указанном ранее пользователем. Выводится площадь фигуры на заданном пользователем интервале и погрешность вычислений. Программа повторяет действие, до того момента, пока пользователь не выберет действие выхода из программы.

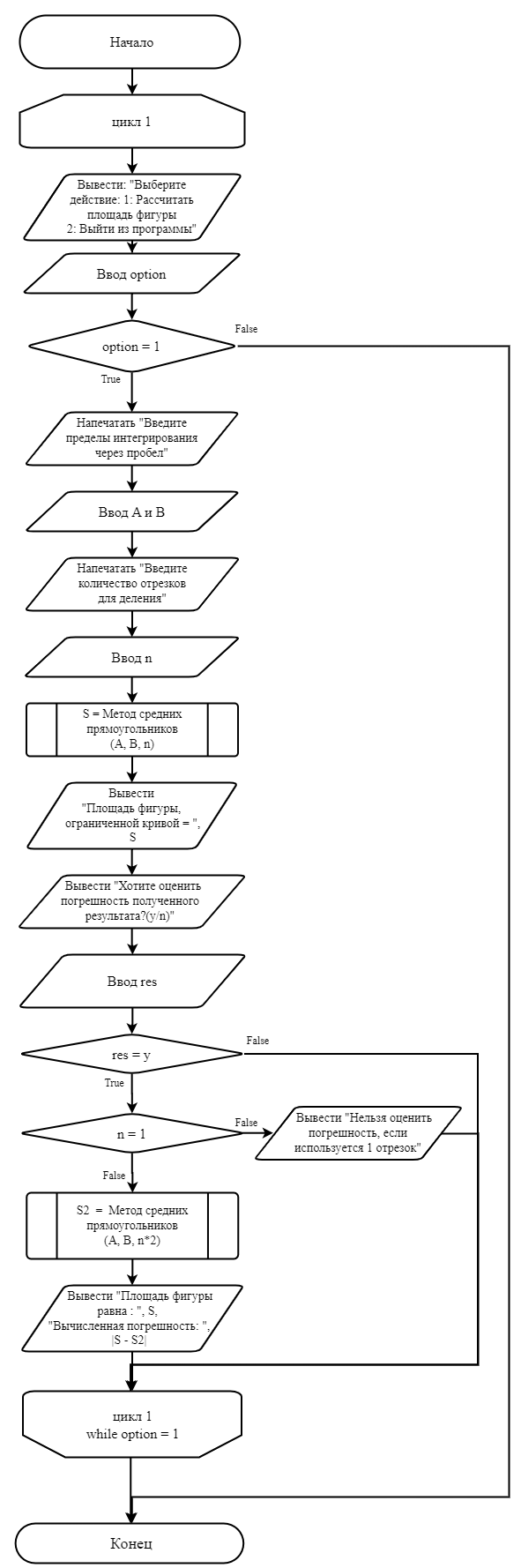
1. Схема алгоритма представлена на рис.1a), рис.1(б)

Рис. 1a) – Схема алгоритма

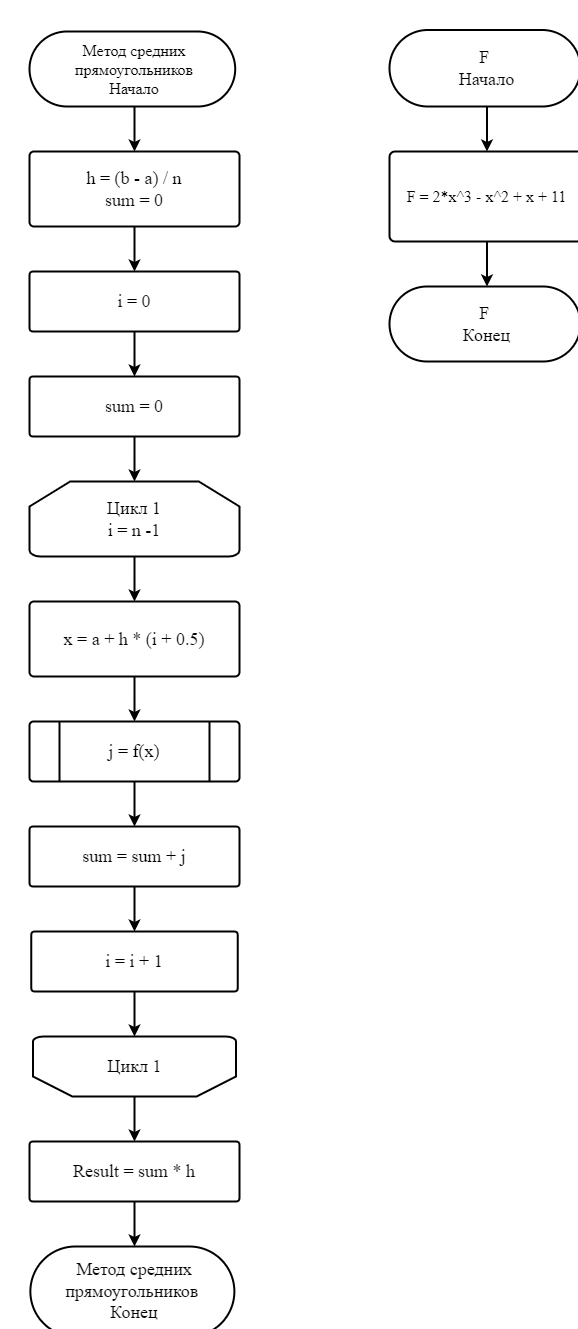


Рис. 1b) – Метод средних прямоугольников

1. **Код программы**

**program** KP3;

**uses** Crt;

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

**begin**

F := 2\*x\*x\*x - x\*x + x + 11;

**end**;

**function** MidpointRectanglesIntegration(a, b: Real; n: Integer): Real;

**var**

h, x, sum: Real;

i: Integer;

**begin**

h := (b - a) / n;

sum := 0;

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

**begin**

x := a + h \* (i + 0.5);

sum := sum + F(x);

**end**;

MidpointRectanglesIntegration := h \* sum;

**end**;

**var**

a, b, S: Real;

n: Integer;

res: Char;

option: Integer;

**begin**

**repeat**

Writeln('Выберите действие:');

Writeln('1. Рассчитать площадь фигуры');

Writeln('2. Выйти из программы');

Readln(option);

ClrScr;

**case** option **of**

1:

**begin**

Writeln('Введите пределы интегрирования a и b через пробел:');

Readln(a, b);

Writeln('Введите количество отрезков для деления:');

Readln(n);

ClrScr;

S := MidpointRectanglesIntegration(a, b, n);

Writeln('Площадь фигуры, ограниченной кривой, равна: ', S:0:10);

Writeln('Хотите оценить погрешность полученного результата? (y/n)');

Readln(res);

**if** (res = 'y') **or** (res = 'Y') **then**

**begin**

**if** n = 1 **then**

**begin**

Writeln('Нельзя оценить погрешность, если используется 1 отрезок.');

**end**

**else**

**begin**

Writeln('Площадь фигуры равна: ', S:0:10);

Writeln('Вычисленная погрешность: ', Abs(S - MidpointRectanglesIntegration(a, b, n\*2)));

**end**;

**end**;

**end**;

2: Writeln('До свидания!');

**end**;

**until** option = 2;

**end**.

1. **Результат выполнения программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 10  9  y | Площадь фигуры равна: 4791.0000000000  Вычисленная погрешность: 18 |
| 1 10  50  y | Площадь фигуры равна: 4814.2224000000  Вычисленная погрешность: 0.583199999999124 |
| -5 25  100  y | Площадь фигуры равна: 190366.7250000000  Вычисленная погрешность: 9.95625000001746 |
| 0 5  200  y | Площадь фигуры равна: 338.3296875000  Вычисленная погрешность: 0.00273437500010232 |
| 0 1  500  y | Площадь фигуры равна: 11.6666660000  Вычисленная погрешность: 5.00000009395762E-07 |

1. **Вывод:**

Цель контрольной работы была достигнута, мной был освоен синтаксис построения процедур и функций, изучены способы передачи данных в подпрограммы, также я получил навыки организации минимального пользовательского интерфейса, изучил численный метод средних прямоугольников. Написанная для решения поставленной задачи программа работает корректно, полностью выполняется её функционал, выходные данные программы представлены в удобном для чтения и анализа виде. В процессе выполнения задания возникали трудности, такие как: “ожидалось: функция, встречено: процедура”, “описание переменной цикла for должно производиться в заголовке цикла”, “ожидалось: “;”, встречено: “var””, однако неисправности были своевременно выявлены и устранены. Считаю проделанную работу успешной, так как после выполнения данной контрольной работы я смогу писать более сложные программы, реализовывать сложные интерфейсы. Я стал лучше понимать структуру и работу языка программирования Pascal.