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 ОРГАНИЗАЦИЙ ПРОЦЕДУР И ФУНКЦИЙ»** **ПО ДИСЦИПЛИНЕ «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

Выполнил: студент учебной группы

ИСПк-203-52-00

Суровцев Александр Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

Содержание

1. Цель лабораторной работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод

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

**2.Формулировка задания(Вариант:19)**

1. Реализовать программу вычисления площади фигуры, ограниченной криво и осью OX (в положительной части по оси OY).

2. Вычисление определённого интеграла должно выполняться численно, с применением метода левых 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.

**3.Описание алгоритма**

Данный алгоритм создаёт интерфейс на консоли, в которой предоставляется возможность найти площадь криволинейной прямой функции с помощью двух методов: метод интеграла и метод левых прямоугольников. Также в главном меню можно назначить другие пределы и оценить погрешность полученного результата между двумя методами.

**4. Схема алгоритма с комментариями**

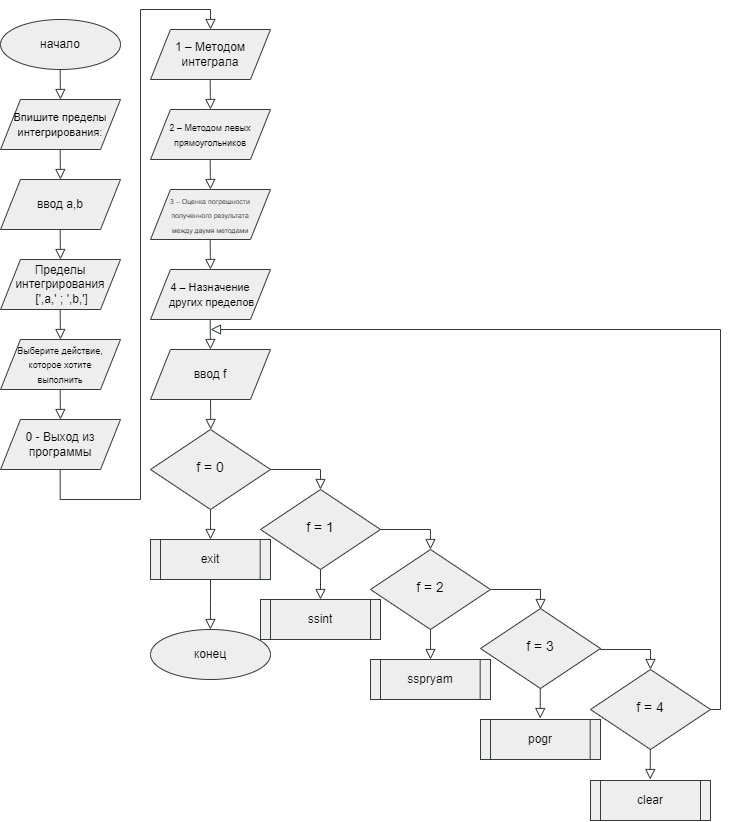


Рисунок 1 – Схема главного алгоритма

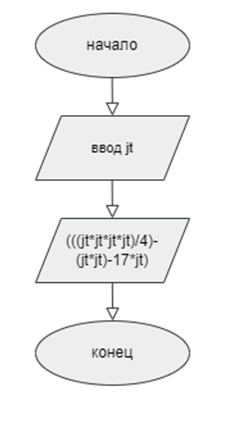


Рисунок 2 – Схема алгоритма процедуры integral

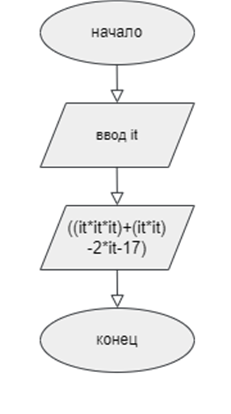


Рисунок 3 – Схема алгоритма функции lpr

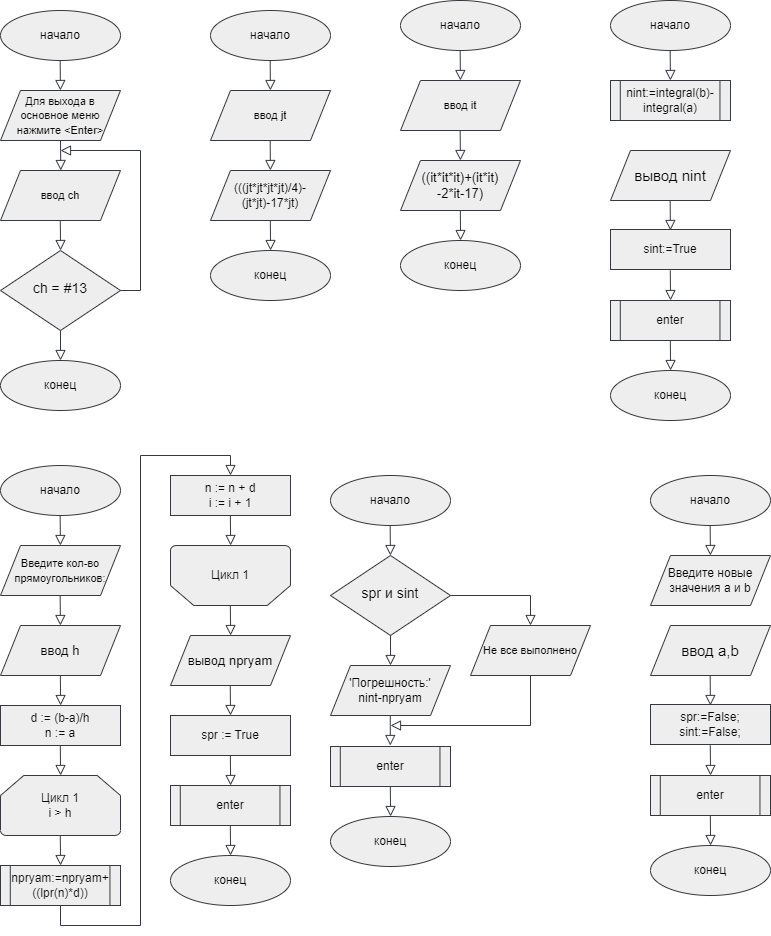


Рисунок 4 – Схема алгоритма процедуры enter

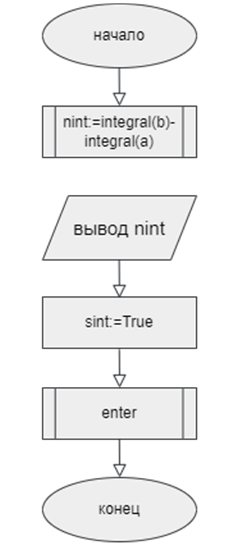


Рисунок 5 – Схема алгоритма процедуры ssint

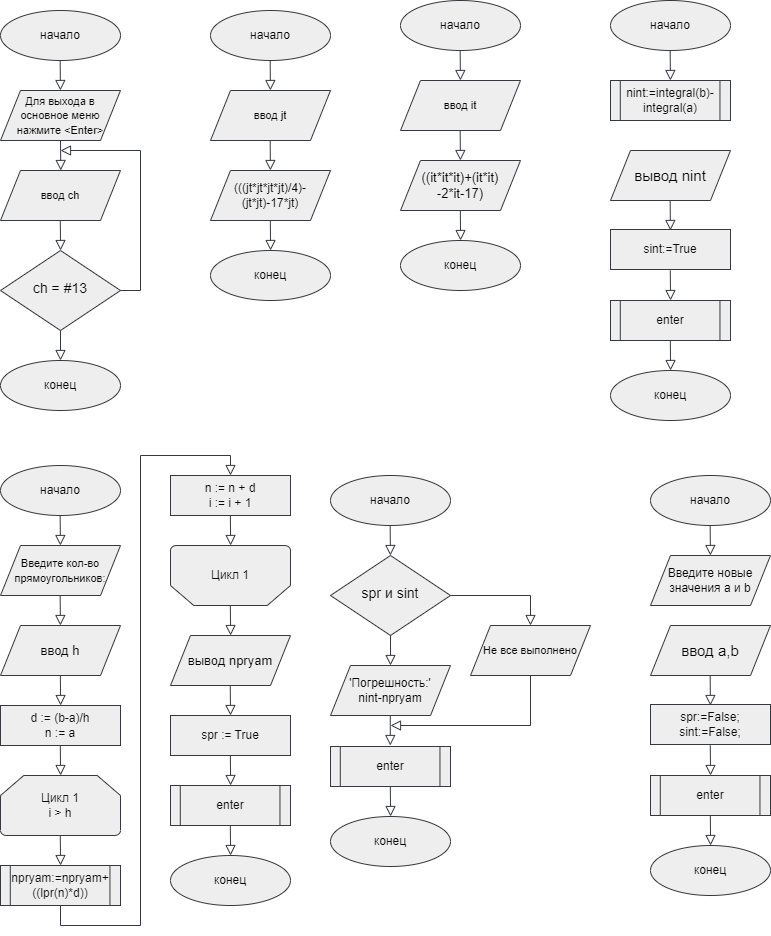


Рисунок 6 – Схема алгоритма процедуры sspryam

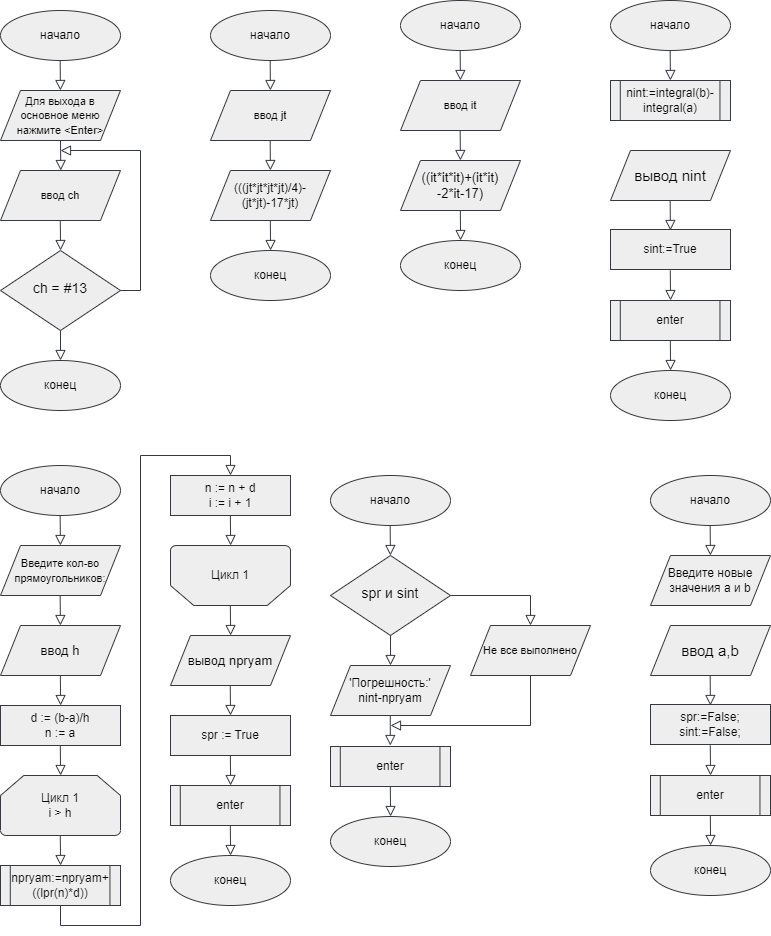


Рисунок 7 – Схема алгоритма процедуры pogr

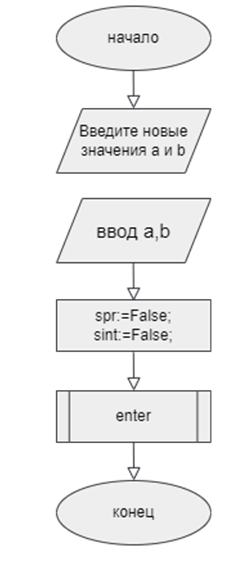


Рисунок 8 – Схема алгоритма процедуры clear

**5. Код программы:**

**uses** crt;

**var** a,b,nint,npryam,d,n:real;ch:char;sint,spr:boolean;h,f:integer;

//левые прямоугольники

**function** lpr(it:real):real;

**begin**

Result:=((it\*it\*it)+(it\*it)-2\*it-17);

**end**;

//интеграл

**function** integral(jt:real):real;

**begin**

Result:=(((jt\*jt\*jt\*jt)/4)-(jt\*jt)-17\*jt);

**end**;

//enter

**procedure** enter;

**begin**

writeln('Для выхода в основное меню нажмите <Enter>');

**repeat**

ch:=readkey;

**until** ch= #13;

**end**;

//интеграловcкая площадь

**procedure** ssint;

**begin**

nint:=integral(b)-integral(a);

writeln('Площадь фигуры ограниченной прямой, вычисленная с помощью интеграла равна ',nint);

sint:=True;

enter;

**end**;

//прямоугольническая площадь

**procedure** sspryam;

**begin**

**var** i:integer;

npryam:=0;

writeln('Введите кол-во прямоугольников:');

read(h);

d := (b-a)/h;

n := a;

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

**begin**

npryam:=npryam+((lpr(n)\*d));

n:=n+d;

**end**;

writeln('Площадь фигуры ограниченной прямой, вычисленная с помощью метода левых прямоугольников: ',npryam);

spr:=True;

enter;

**end**;

//сброс

**procedure** clear;

**begin**

writeln('Введите новые значения a и b');

readln(a,b);

spr:=False;

sint:=False;

enter;

**end**;

//погрешность

**procedure** pogr;

**begin**

**if** spr **and** sint **then**

writeln('Погрешность:',nint-npryam)

**else**

writeln('Не все выполнено');

enter;

**end**;

//case-menu

**begin**

writeln('Впишите пределы интегрирования:');

readln(a,b);

**repeat**

ClrScr;

writeln('Пределы интегрирования [',a,' ; ',b,']');

writeln('Выберите действие, которое хотите выполнить');

writeln('0 - Выход из программы');

writeln('1 – Методом интеграла');

writeln('2 – Методом левых прямоугольников');

writeln('3 – Оценка погрешности полученного результата между двумя методами');

writeln('4 – Назначение других пределов');

writeln( );

readln(f);

**case** f **of**

0: **exit**;

1: ssint;

2: sspryam;

3: pogr;

4: clear;

**end**;

**until** f = 0;

**end**.



Рисунок 9 - Результат выполнения программы

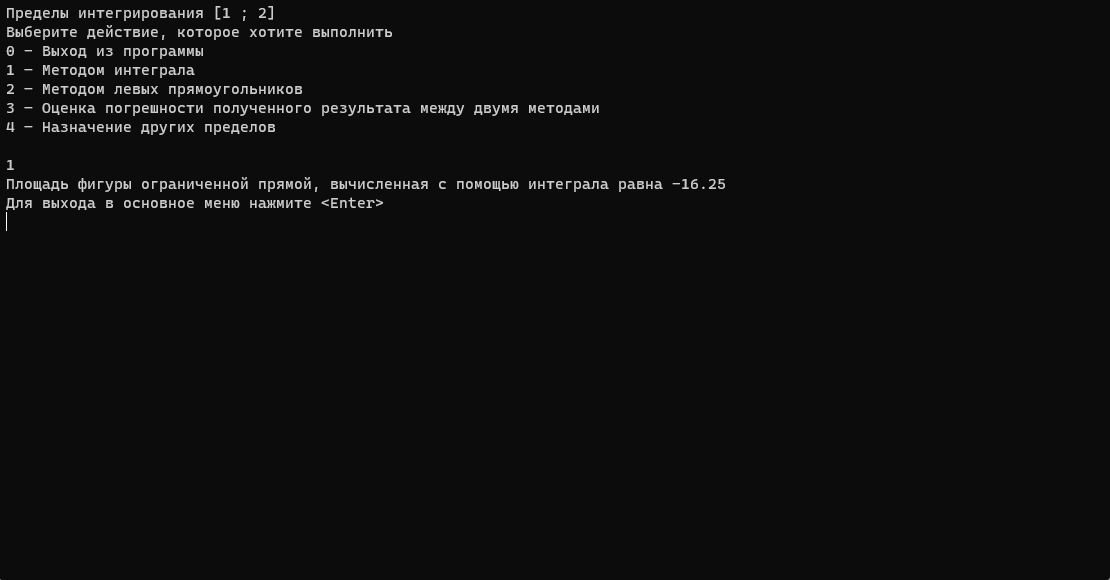


Рисунок 10 – Результат выполнения процедуры ssint

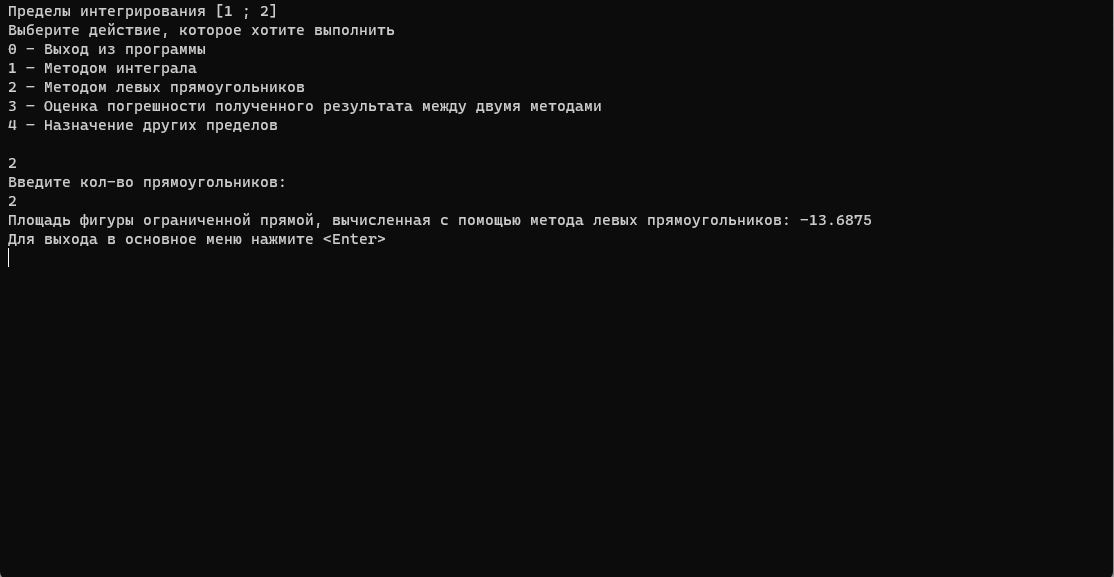


Рисунок 11 – Результат выполнения процедуры sspryam

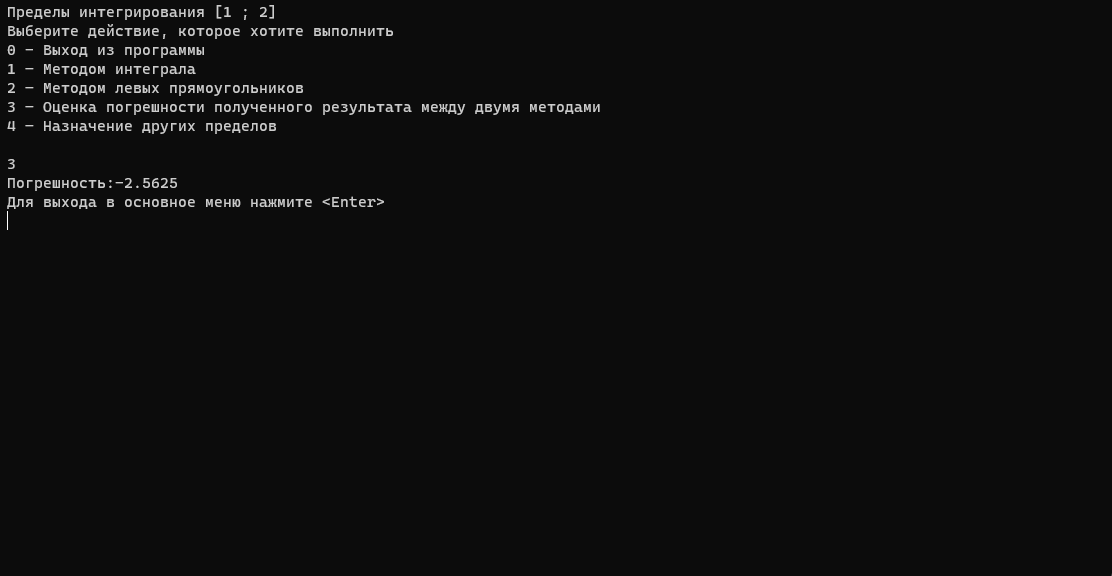


Рисунок 12 – Результат выполнения процедуры pogr

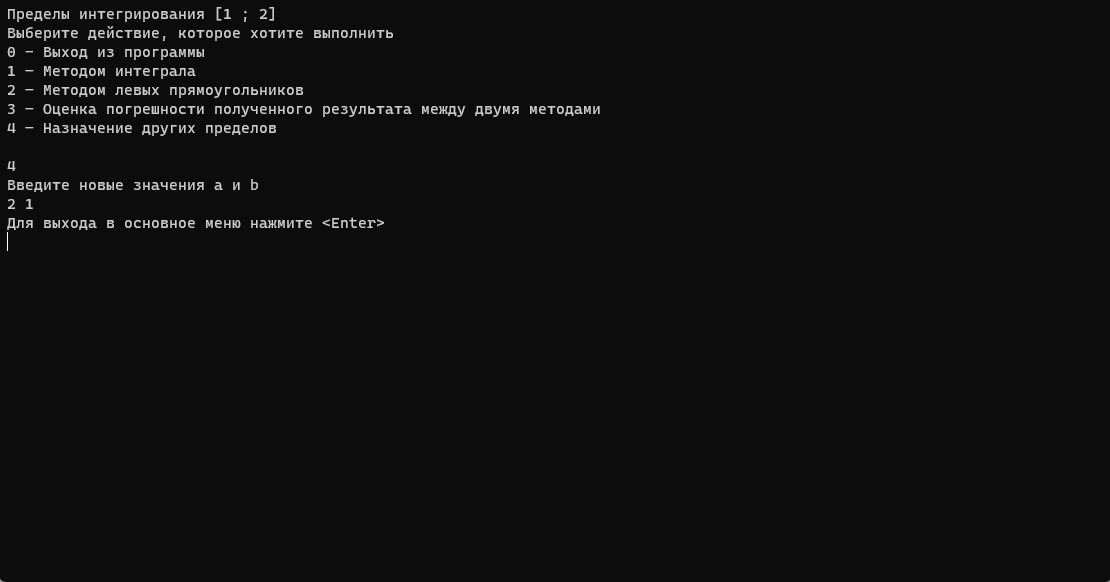


Рисунок 13 – Результат выполнения процедуры clear

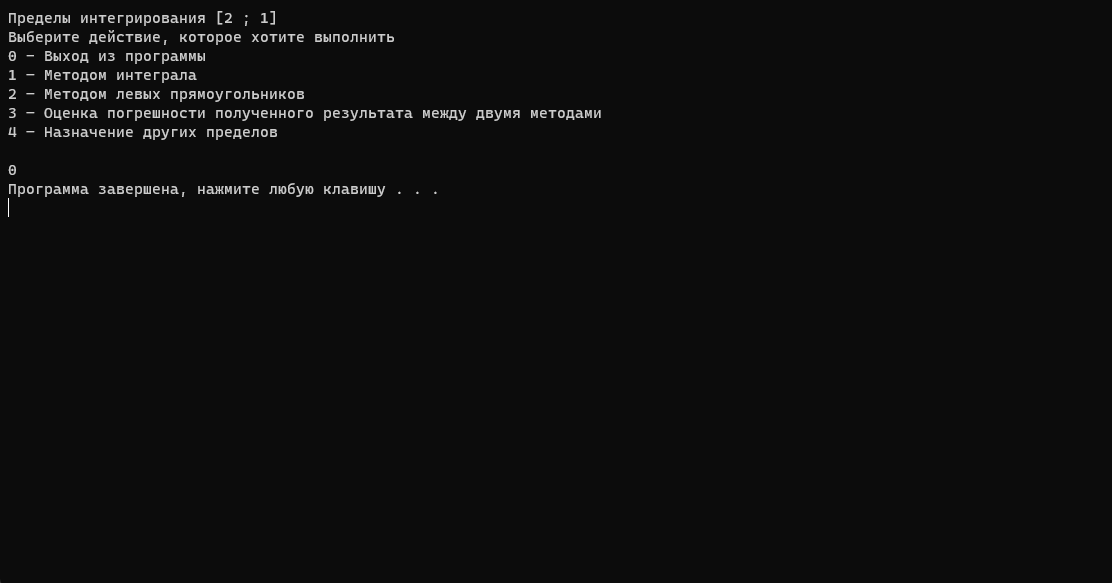


Рисунок 14 – Результат выполнения команды exit

**7. Вывод**

В ходе выполнения данной работы мы смогли применить изученный материал лабораторных занятий на практике, использовав библиотеку CRT, команду ClrScr, процедуры, функции, условный оператор, цикл while, создание переменных и операторов ввода и вывода.

В результате выполнения данной работы мы изучили базовую структуру организации программы и основные конструкции языка Pascal, а также выполнили данное нам практическое задание, использовав изученный материал на практике.

Для создания схемы алгоритмов мы использовали приложение Draw.io. Draw.io — это бесплатный онлайн-сервис, который помогает создавать схемы алгоритмов, прототипы, инфографику и диаграммы любого вида. Чаще всего его используют именно для построения диаграмм, поэтому недавно сервис переименовали в Diagrams.net. Но старое название по-прежнему в ходу. Сервисом пользуются разные IT-специалисты — от бизнес-аналитиков до разработчиков и дизайнеров.