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

Ситников Иван Владимирович

Преподаватель:

Кузьминых Ангелина Владимировна

Киров

2023

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

**Задание**

1. Реализовать программу вычисления площади фигуры, ограниченной кривой 2\*x^3+(-1)\*x^2+(1)\*x+(11) и осью OX (в положительной части оси OY).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода левых прямоугольников.
3. Взаимодействие с пользователем должно осуществляться посредством case-меню.

**Описание алгоритма.**

Алгоритм вычисляет площадь фигуры, ограниченной кривой 2\*x^3+(-1)\*x^2+(1)\*x+(11) и осью OX (в положительной части оси OY).

1. Создание массива:

При помощи цикла for, массив заполняется элементами, вводимыми с клавиатуры.

1. Использование цикла for для переменной i и вложенного в него цикла for для переменной j:

Сравниваются элемент массива на позиции i и элемент массива на позиции j.

1. Подсчет количества различных элементов:

Если встречается неповторяющийся элемент, то к переменной count прибавляется 1.

1. Вывод результата:

Выводим сообщение в формате 'Количество различных элементов в массиве: count', где count – количество различных элементов в массиве.

**Схема алгоритма**

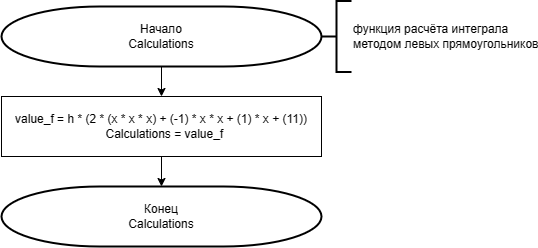


Рисунок 1.1 – Функция Calculations

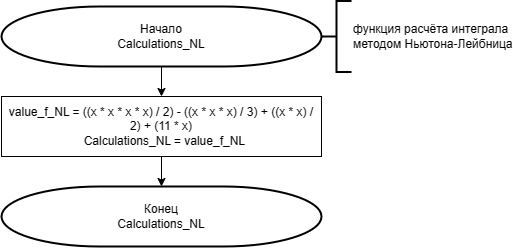


Рисунок 1.2 – Функция Calculations\_NL

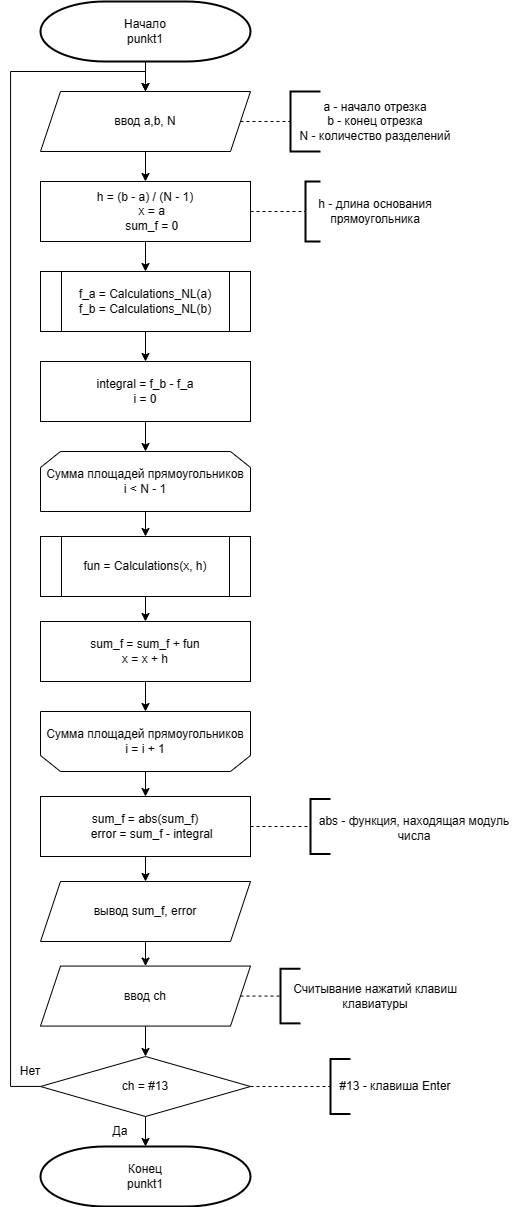


Рисунок 1.3 – Процедура punkt1

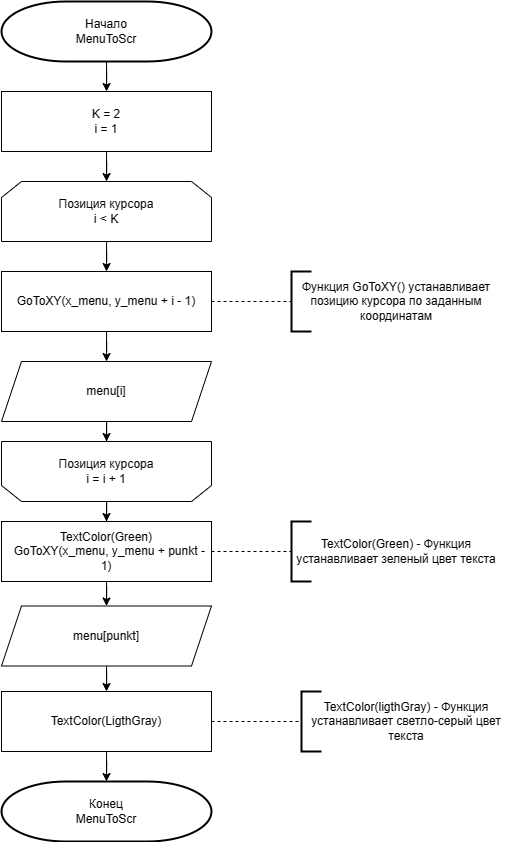


Рисунок 1.4 – Процедура MenuToScr

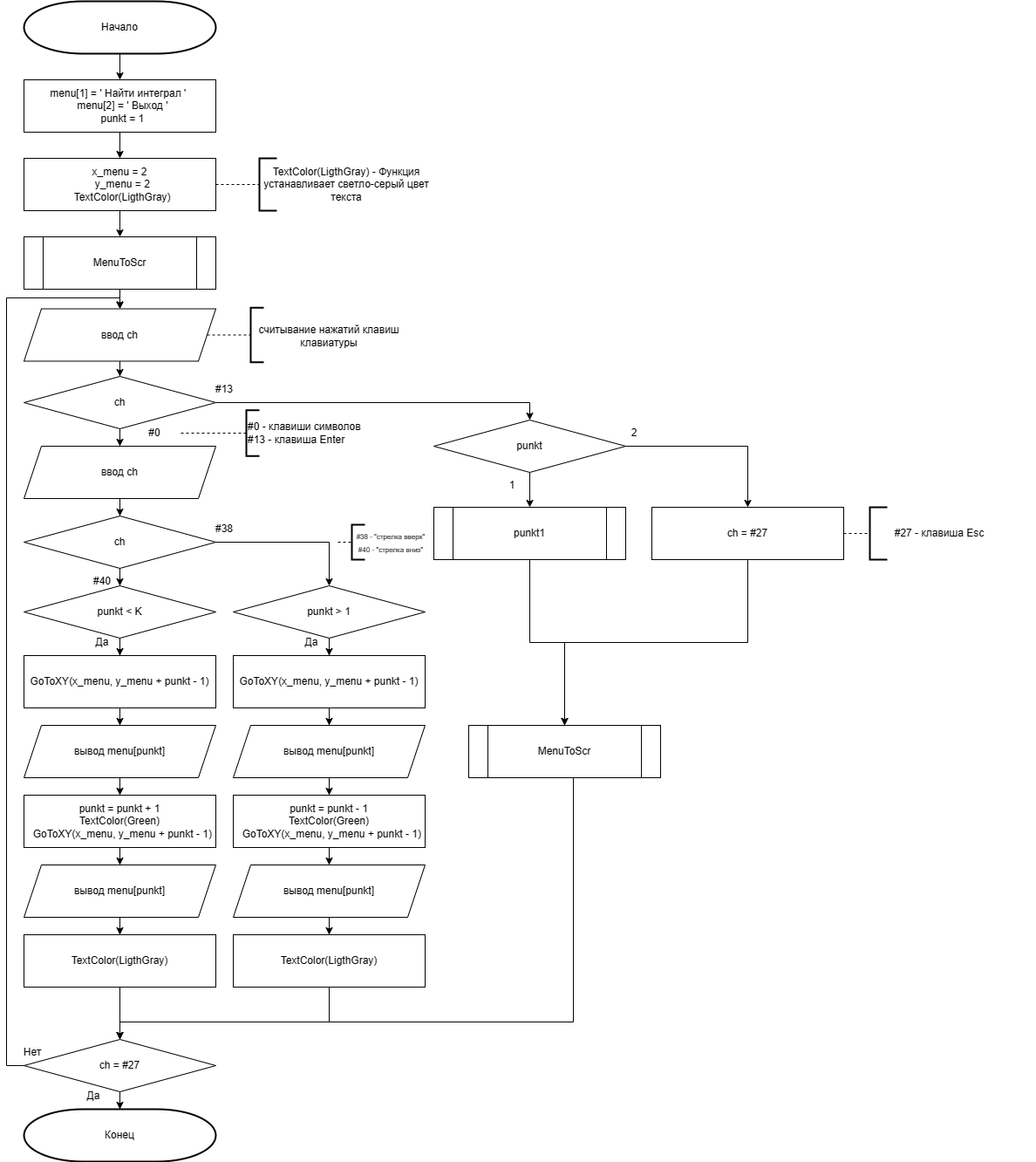


Рисунок 1.5 – Тело программы

**Код программы**

program FindIntegral;

uses CRT;

const

NORM = LightGray;

SEL = Green;

K = 2;

var

menu: array[1..K] of string;

punkt: integer;

ch: char;

x\_menu, y\_menu: integer;

function Calculations(var x, h: real): real;

var

value\_f: real;

begin

value\_f := h \* (2 \* (x \* x \* x) + (-1) \* x \* x + (1) \* x + (11));

Calculations := value\_f;

end;

function Calculations\_NL(var x: real): real;

var

value\_f\_NL: real;

begin

value\_f\_NL := ((x \* x \* x \* x) / 2) - ((x \* x \* x) / 3) + ((x \* x) / 2) + (11 \* x);

Calculations\_NL := value\_f\_NL;

end;

var

x, h, a, b, sum\_f, fun, integral, f\_a, f\_b, error: real;

i, N, choice: integer;

procedure punkt1;

begin

ClrScr;

var i: integer;

repeat

begin

writeln('Введите пределы интегрирования через пробел:');

readln(a, b);

CLRSCR;

writeln('Введите количество разделений:');

readln(N);

CLRSCR;

h := (b - a) / (N - 1);

x := a;

sum\_f := 0;

f\_a := Calculations\_NL(a);

f\_b := Calculations\_NL(b);

integral := f\_b - f\_a;

for i := 0 to (N - 1) do

begin

fun := Calculations(x, h);

sum\_f := sum\_f + fun;

x := x + h;

end;

sum\_f := abs(sum\_f);

error := sum\_f - integral;

writeln('Результат вычислений:', sum\_f);

writeln('Абсолютная погрешность: ', error);

writeln('<<Нажмите Enter чтобы выйти в меню>>');

ch := readkey;

end;

until ch = #13;

end;

procedure MenuToScr;

var

i: integer;

begin

ClrScr;

for i := 1 to K do

begin

GoToXY(x\_menu, y\_menu + i - 1);

write(menu[i]);

end;

TextColor(SEL);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

begin

menu[1] := ' Найти интеграл ';

menu[2] := ' Выход ';

punkt := 1;

x\_menu := 2;

y\_menu := 2;

TextColor(NORM);

MenuToScr;

repeat

ch := ReadKey;

if ch = #0 then

begin

ch := ReadKey;

case ch of

#40: {Вниз}

if punkt < K then

begin

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

punkt := punkt + 1;

TextColor(SEl);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

#38: {Вверх}

if punkt > 1 then

begin

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

punkt := punkt - 1;

TextColor(SEl);

GoToXY(x\_menu, y\_menu + punkt - 1);

write(menu[punkt]);

TextColor(NORM);

end;

end;

end

else

if ch = #13 then

begin

case punkt of

1: punkt1;

2: ch := #27;

end;

MenuToScr;

end;

until ch = #27;

end.

**Результаты выполнения программы**

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1 10  100 | Результат вычислений:4903.30578512396  Абсолютная погрешность: 88.3057851239637 |
| 1 100  231 | Результат вычислений:50101987.5165028  Абсолютная погрешность: 429232.51650276 |
| 21 200  200 | Результат вычислений:804463138.850025  Абсолютная погрешность: 7202210.5166918 |
| 1 121  100 | Результат вычислений:108746846.648302  Абсолютная погрешность: 2149286.64830163 |
| 1 10  10000 | Результат вычислений:4815.87042592537  Абсолютная погрешность: 0.870425925372729 |

**Вывод**

Программы и алгоритмы, разработанные в ходе работы, представляют собой пример построения процедур и функций, а также организации минимального пользовательского интерфейса на языке программирования Pascal.

В ходе написания программы для вычисления определенного интеграла методом левых прямоугольников была изучена теория по данной теме, а затем реализован алгоритм для нахождения интеграла. Также при разработке алгоритма был реализован минимальный пользовательский интерфейс с использованием case-меню.

В итоге, программа выводит результаты, удовлетворяющие условиям задачи. Был реализован алгоритм, как пример работы с функциями и процедурами в языке программирования Pascal. А также пример работы с функциями, процедурами, и подключением библиотеки Crt для разработки минимального пользовательского интерфейса.