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-205-52-00

Ильин Тимофей Анатольевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров 2023

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

**Задание 1 (Вариант 7)**

1. Реализовать программу вычисления площади фигуры, ограниченной кривой

1 \*x^3 + (-2) \*x ^ 2 + (4) \* x + (18) и осью ОХ (в положительной части по оси OY).

1. ﻿﻿﻿Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона.
2. ﻿﻿﻿Пределы интегрирования вводятся пользователем.
3. ﻿﻿﻿Взаимодействие с пользователем должно осуществляться посредством case-меню.
4. ﻿﻿﻿Требуется реализовать возможность оценки погрешности полученного результата.
5. ﻿﻿﻿Необходимо использовать процедуры и функции там, где это целесообразно.

**Описание алгоритма**

1. Процедура casemenu принимает на вход два вещественных числа a и b, которые задают границы отрезка, на котором вычисляется интеграл, а также две переменные area и error, которые будут использоваться для хранения значения интеграла и погрешности соответственно.

2. Внутри процедуры определяются и инициализируются необходимые переменные: n - количество разбиений отрезка, i - счетчик для цикла, x и h - вспомогательные переменные для вычисления значений функции f(x) на различных точках отрезка, sum1 и sum2 - промежуточные суммы.

3. Задается значение переменной n равное 100, что означает, что отрезок [a, b] будет разбит на 100 равных частей.

4. Вычисляется величина h, которая представляет собой длину каждого подотрезка и равна (b - a) / n.

5. Инициализируются переменные sum1 и sum2 нулевыми значениями.

6. В цикле от 1 до n div 2 (целая часть от деления n на 2) производятся вычисления значения функции f(x) в двух точках каждого подотрезка и добавляются к промежуточным суммам sum1 и sum2 соответственно.

7. После выполнения цикла получаем значения сумм sum1 и sum2, которые соответствуют треугольным частям площадей на графике функции f(x).

8. Вычисляется значение интеграла с использованием параболического правила Симпсона: sum = h / 3 \* (f(a) + 4 \* sum1 + 2 \* sum2 + f(b)), где f(a) и f(b) - значения функции на границах отрезка.

9. Вычисляется значение площади под графиком функции (интеграла) - area, как абсолютное значение значения sum.

10. Вычисляется значение погрешности вычисления интеграла - error, как абсолютное значение разности значений sum и area.

11. Результаты вычислений интеграла и погрешности сохраняются в переменных area и error соответственно.

12. Объявляем переменные a, b, area, error как вещественные числа и choice как целое число.

13. Начинаем цикл repeat...until, который будет выполняться до выбора пользователем пункта меню "Выход" (choice = 2).

14. Выводим на экран пункты меню: "Вычислить площадь фигуры" и "Выход".

15. Просим пользователя выбрать пункт меню и считываем его выбор в переменную choice.

16. Проверяем выбор пользователя с помощью оператора case. Если пользователь выбрал пункт 1, выполняем следующий набор действий:

17. Просим пользователя ввести нижнюю границу интегрирования и считываем ее в переменную a.

18. Просим пользователя ввести верхнюю границу интегрирования и считываем ее в переменную b.

19. Вызываем функцию casemenu, передавая ей значения a, b, area и error.

20. Выводим на экран значение площади фигуры с помощью функции writeln и форматируем вывод как вещественное число с 2 знаками после запятой (area:0:2).

21. Выводим на экран оценку погрешности с помощью функции writeln и форматируем вывод как вещественное число с 2 знаками после запятой (error:0:2).

22. Если пользователь выбрал пункт 2, выводим на экран "Программа завершена".

23. Если пользователь ввел некорректный выбор, выводим на экран "Неверный выбор. Попробуйте снова".

24. Переходим на новую строку.

25. Проверяем условие цикла until, если choice = 2, то выходим из цикла, иначе повторяем цикл с пунктом 3.

**Схема алгоритма с комментариями**

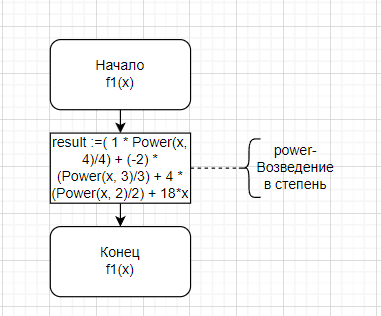


Рис.1 Схема алгоритма function f1(x).

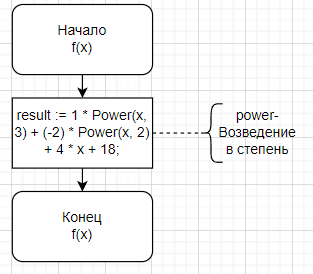
****

Рис.2 Схема алгоритма function f(x).

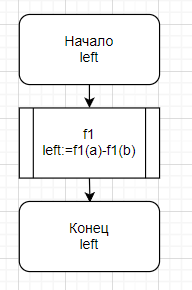


Рис.3 Схема алгоритма function left.

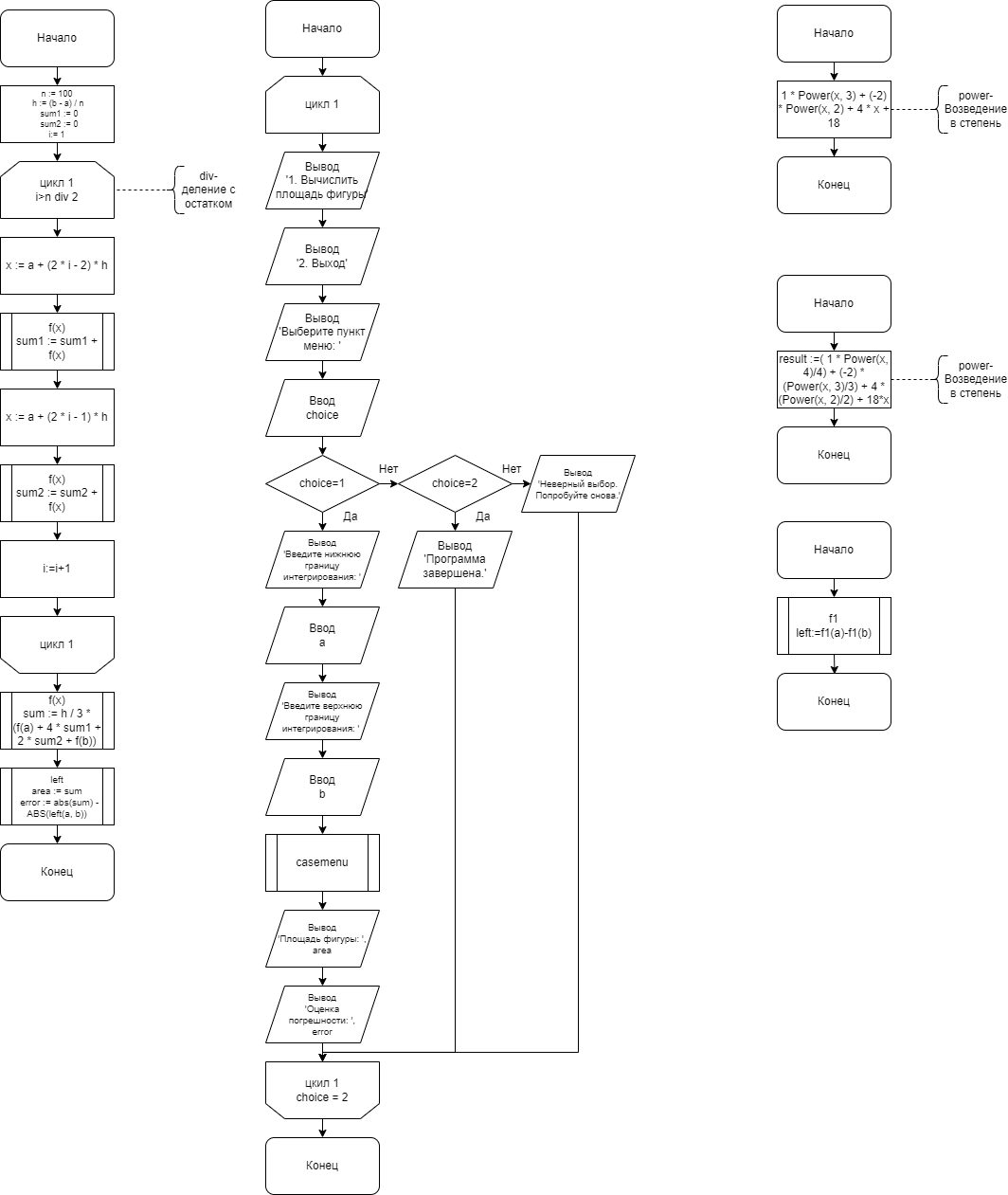


Рис4 Схема алгоритма procedure casemenu.

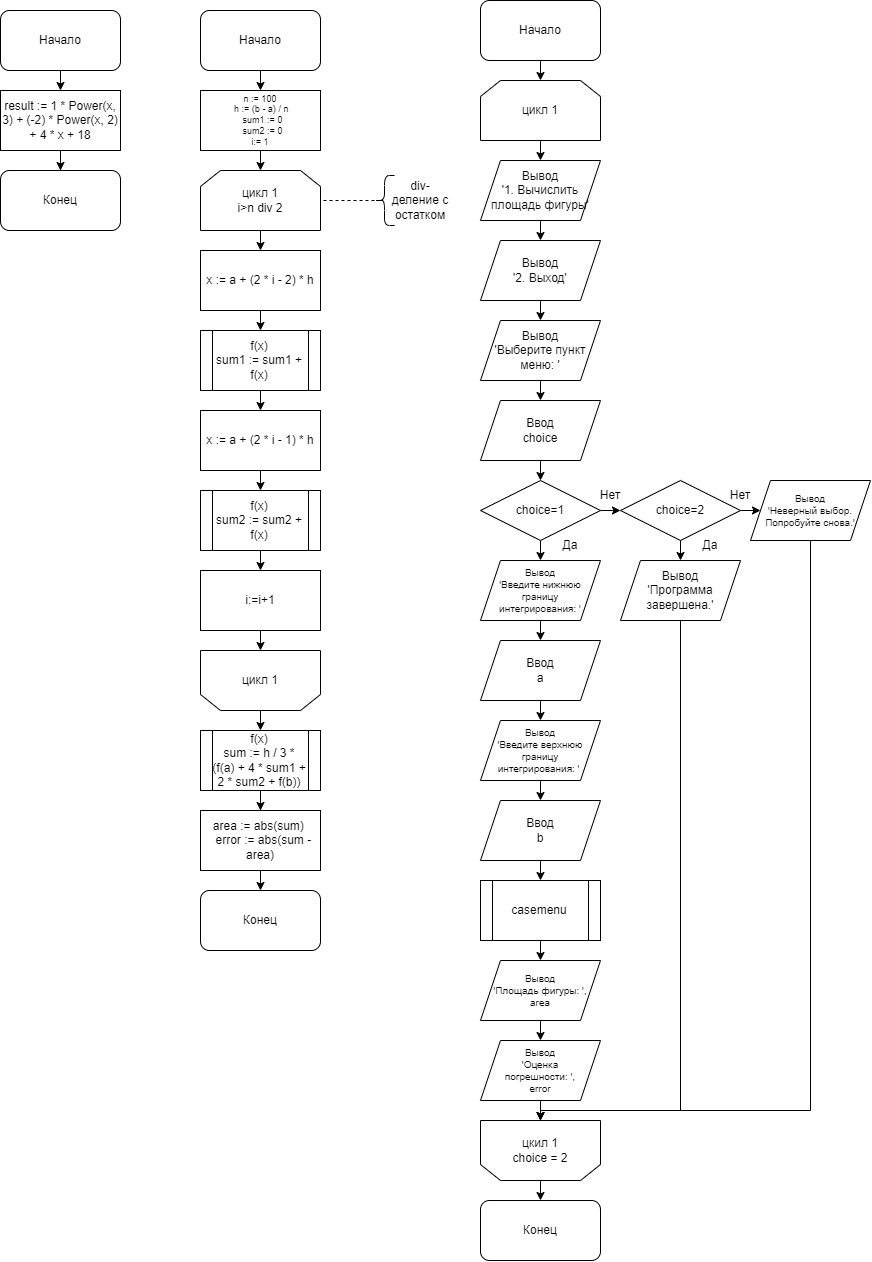


Рис.3 Схема алгоритма задачи.

**Код программы**

**program** dkr3;

**function** f(x: real): real;

**begin**

result := 1 \* Power(x, 3) + (-2) \* Power(x, 2) + 4 \* x + 18;

**end**;

**function** f1(x: real): real;

**begin**

result :=( 1 \* Power(x, 4)/4) + (-2) \* (Power(x, 3)/3) + 4 \* (Power(x, 2)/2) + 18\*x;

**end**;

**function** left(a, b:real):real;

**var**

h, x, sum:real;

**begin**

left:=f1(a)-f1(b);

**end**;

**procedure** casemenu (a, b: real; **var** area, error: real);

**var**

n, i: integer;

x, h: real;

sum1, sum2, sum: real;

**begin**

n := 100;

h := (b - a) / n;

sum1 := 0;

sum2 := 0;

**for** i := 1 **to** n **div** 2 **do**

**begin**

x := a + (2 \* i - 2) \* h;

sum1 := sum1 + f(x);

x := a + (2 \* i - 1) \* h;

sum2 := sum2 + f(x);

**end**;

sum := h / 3 \* (f(a) + 4 \* sum1 + 2 \* sum2 + f(b));

area := sum;

error := abs(sum) - ABS(left(a, b));

**end**;

**var**

a, b, area, error: Real;

choice: Integer;

**begin**

**repeat**

Writeln('1. Вычислить площадь фигуры');

Writeln('2. Выход');

Write('Выберите пункт меню: ');

Readln(choice);

**case** choice **of**

1:

**begin**

Write('Введите нижнюю границу интегрирования: ');

Readln(a);

Write('Введите верхнюю границу интегрирования: ');

Readln(b);

casemenu(a, b, area, error);

Writeln('Площадь фигуры: ', area:0:2);

Writeln('Оценка погрешности: ', error:0:2);

**end**;

2:

**begin**

Writeln('Программа завершена.');

**end**;

**else**

**begin**

Writeln('Неверный выбор. Попробуйте снова.');

**end**;

**end**;

Writeln;

**until** choice = 2;

**end**.

**Результат выполнения программы**

Пример 1. Выбираем пункт 1, вводим 6 и -4.

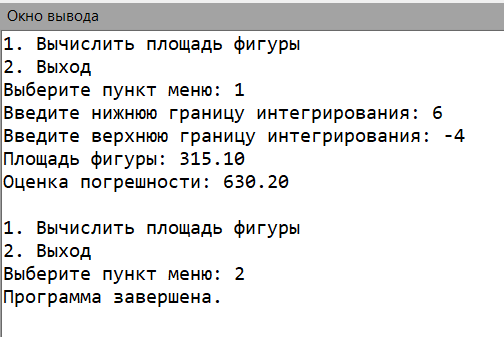


Рис.4 Результат выполнения программы (Пример 1).

Пример 2. Выбираем пункт 2.

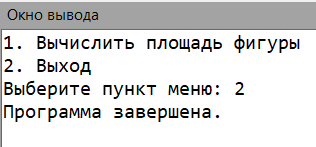


Рис.5 Результат выполнения программы (Пример 2).

Пример 3. Вводим 3.

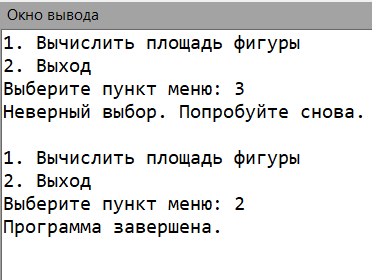


Рис.6 Результат выполнения программы (Пример 3).

Вывод: в результате выполнения домашней контрольной работы был освоен синтаксис построения процедур и функций, изучены способы передачи данных в подпрограммы, а также получены навыки организации минимального пользовательского интерфейса. Благодаря этому, я научился писать case-menu, который позволяет пользователю выбирать различные опции или действия. Более того, я научился оформлять case-menu, чтобы он выглядел структурированным и легким в использовании. Кроме того, я научился рисовать блок-схему для case-menu, что помогает лучше понять его логику и организацию. Эти навыки мне пригодятся при разработке программ и создании удобного пользовательского интерфейса.