Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

Колледж ВятГУ

**ОТЧЕТ**

**ПО ДОМАШНЕЙ КОНТРОЛЬНОЙ РАБОТЕ №4**

**«Работа в графическом режиме»**

**ПО ДИСЦИПЛИНЕ «Основы алгоритмизации и программирования»**

Выполнил: студент учебной группы

ИСПк-101-51-00

Долинин Владислав Александрович

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2024

В отчете должны отображаться:

1. Цель работы
2. Формулировка задания (с вариантом)
3. Описание алгоритма и ответы на вопросы
4. Схема алгоритма с комментариями
5. Код программы
6. Результат выполнения программы
7. Вывод
8. **Цель работы:** освоить принципы работы в графическом режиме; получить базовые навыки взаимодействия с графическими примитивами.
9. **Вариант 1.**

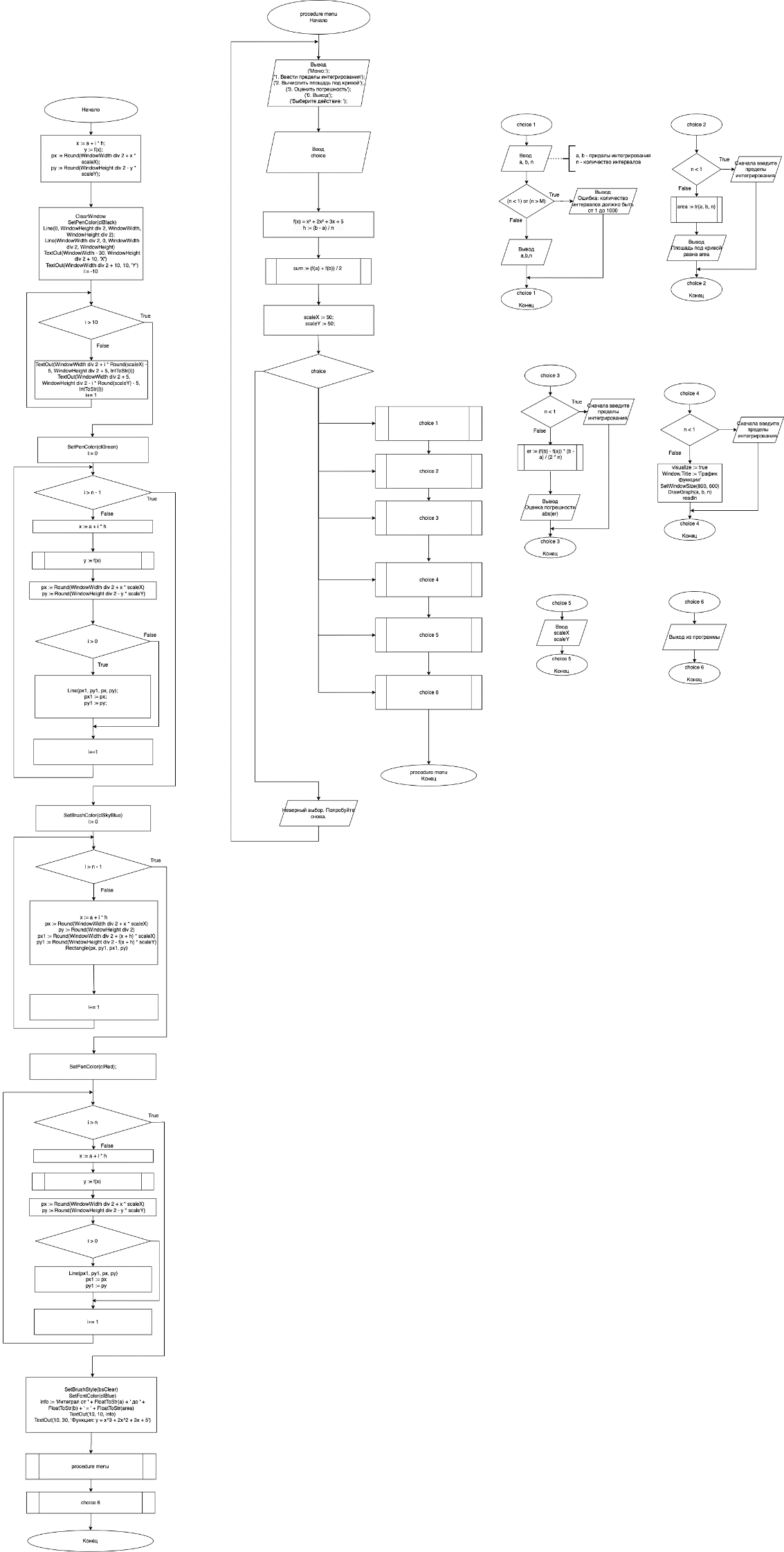
**Формулировка заданий:**

* Дополнить программу, реализованную в ходе предыдущей лабораторной работы, режимом визуализации.
* Предусмотреть возможность вывода кривой, ограничивающей фигуру, на координатную плоскость.
* Реализовать следующие возможности и элементы: масштабирование графика, подписи на осях, вывод информации о задании.
* Реализовать не менее двух возможностей из представленных: независимое масштабирование по осям, штриховка вычисляемой площади, визуализация численного расчета интеграла.

1. **Описание алгоритма.**

* Функция f(x: real): real: Это функция, которую вы интегрируете. Она определена как f(x) = x³ + 2x² + 3x + 5 .
* Функция tr(a, b: real; n: integer): real: Реализует метод трапеций для численного интегрирования. Она делит интервал [a, b] на n частей и вычисляет площадь под кривой.
* Процедура DrawGraph(a, b: real; n: integer): Отвечает за рисование графика функции и визуализацию площади под кривой. Она использует библиотеку GraphABC для рисования.
* Процедура menu: Создает текстовое меню для взаимодействия с пользователем. Пользователь может вводить пределы интегрирования, вычислять площадь, оценивать погрешность, визуализировать график или изменять масштаб.
* Основная программа: Инициализирует переменные и запускает меню.

1. **Схема алгоритма.**

****

1. **Код программы.**

**uses** GraphABC;

**const**

M = 1000;

**var**

a, b: real;

n: integer;

area: real;

er: real;

scaleX, scaleY: real;

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

**begin**

f := 1 \* x \* x \* x + 2 \* x \* x + 3 \* x + 5;

**end**;

**function** tr(a, b: real; n: integer): real;

**var**

h, sum: real;

i: integer;

**begin**

h := (b - a) / n;

sum := (f(a) + f(b)) / 2;

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

sum := sum + f(a + i \* h);

tr := sum \* h;

**end**;

**procedure** DrawGraph(a, b: real; n: integer);

**var**

x, y, h: real;

i: integer;

px, py, px1, py1: integer;

info: string;

**begin**

ClearWindow;

SetPenColor(clBlack);

Line(0, WindowHeight **div** 2, WindowWidth, WindowHeight **div** 2); // Ось X

Line(WindowWidth **div** 2, 0, WindowWidth **div** 2, WindowHeight); // Ось Y

TextOut(WindowWidth - 30, WindowHeight **div** 2 + 10, 'X');

TextOut(WindowWidth **div** 2 + 10, 10, 'Y');

**for** i := -10 **to** 10 **do**

**begin**

TextOut(WindowWidth **div** 2 + i \* Round(scaleX) - 5, WindowHeight **div** 2 + 5, IntToStr(i));

TextOut(WindowWidth **div** 2 + 5, WindowHeight **div** 2 - i \* Round(scaleY) - 5, IntToStr(i));

**end**;

SetPenColor(clGreen);

h := (b - a) / n;

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

**begin**

x := a + i \* h;

y := f(x);

px := Round(WindowWidth **div** 2 + x \* scaleX);

py := Round(WindowHeight **div** 2 - y \* scaleY);

**if** i > 0 **then**

Line(px1, py1, px, py);

px1 := px;

py1 := py;

**end**;

SetBrushColor(clSkyBlue);

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

**begin**

x := a + i \* h;

px := Round(WindowWidth **div** 2 + x \* scaleX);

py := Round(WindowHeight **div** 2);

px1 := Round(WindowWidth **div** 2 + (x + h) \* scaleX);

py1 := Round(WindowHeight **div** 2 - f(x + h) \* scaleY);

Rectangle(px, py1, px1, py);

**end**;

SetPenColor(clRed);

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

**begin**

x := a + i \* h;

y := f(x);

px := Round(WindowWidth **div** 2 + x \* scaleX);

py := Round(WindowHeight **div** 2 - y \* scaleY);

**if** i > 0 **then**

Line(px1, py1, px, py);

px1 := px;

py1 := py;

**end**;

SetBrushStyle(bsClear);

SetFontColor(clBlue);

info := 'Интеграл от ' + FloatToStr(a) + ' до ' + FloatToStr(b) + ' = ' + FloatToStr(area);

TextOut(10, 10, info);

TextOut(10, 30, 'Функция: y = x^3 + 2x^2 + 3x + 5');

**end**;

**procedure** menu;

**var**

choice: integer;

visualize: boolean;

**begin**

scaleX := 50;

scaleY := 50;

**repeat**

writeln('Выберите действие:');

writeln('1. Ввести пределы интегрирования');

writeln('2. Вычислить площадь под кривой');

writeln('3. Оценить погрешность');

writeln('4. Визуализация');

writeln('5. Изменить масштаб');

writeln('6. Выход');

write('Выберите действие: ');

readln(choice);

**case** choice **of**

1:

**begin**

write('Введите нижний предел (A): ');

readln(a);

write('Введите верхний предел (B): ');

readln(b);

write('Введите количество интервалов (n <= ', M, '): ');

readln(n);

**if** (n < 1) **or** (n > M) **then**

writeln('Ошибка: количество интервалов должно быть от 1 до ', M);

**end**;

2:

**begin**

**if** n < 1 **then**

writeln('Сначала введите пределы интегрирования.')

**else**

**begin**

area := tr(a, b, n);

writeln('Площадь под кривой от ', a:0:2, ' до ', b:0:2, ' равна ', area:0:4);

**end**;

**end**;

3:

**begin**

**if** n < 1 **then**

writeln('Сначала введите пределы интегрирования.')

**else**

**begin**

er := (f(b) - f(a)) \* (b - a) / (2 \* n);

writeln('Оценка погрешности: ', abs(er):0:4);

**end**;

**end**;

4:

**begin**

**if** n < 1 **then**

writeln('Сначала введите пределы интегрирования.')

**else**

**begin**

visualize := true;

Window.Title := 'График функции';

SetWindowSize(800, 600);

DrawGraph(a, b, n);

readln;

**end**;

**end**;

5:

**begin**

write('Введите новый масштаб по X: ');

readln(scaleX);

write('Введите новый масштаб по Y: ');

readln(scaleY);

**end**;

6: writeln('Выход из программы.');

**else** writeln('Неверный выбор. Попробуйте снова.');

**end**;

writeln;

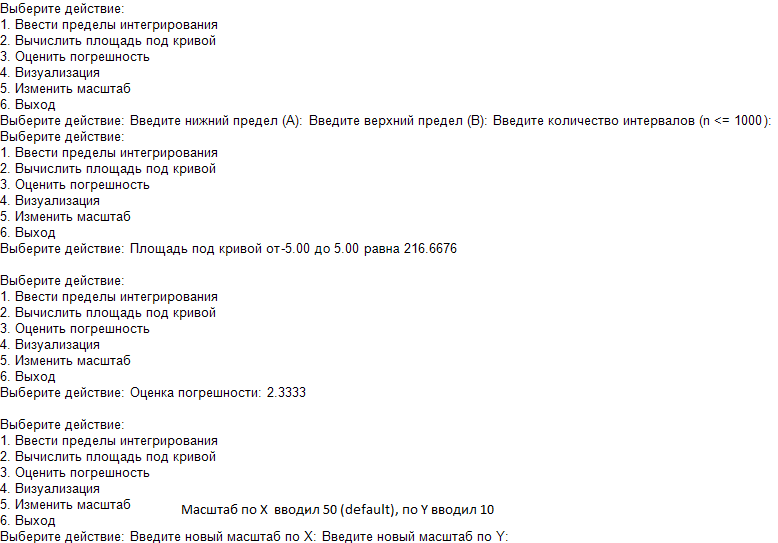
**until** choice = 6;

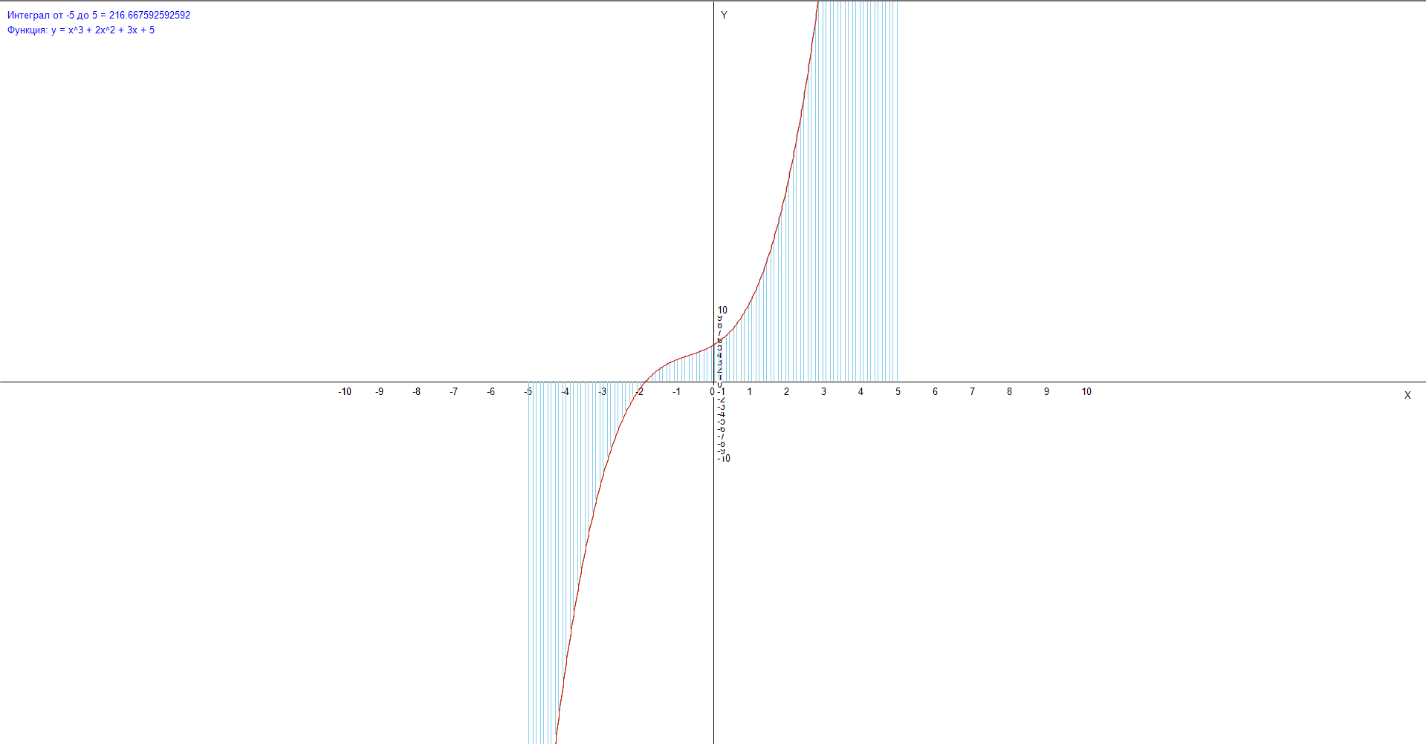
**end**;

**begin**

menu;

**end**.

1. **Результат выполнения программы.**

****

1. **Вывод:** В ходе работы был написан код программы по условиям из пункта 2 (формулировка заданий), построена блок схема, сделано описание алгоритма. Так же выведен результат выполнения программы. Я столкнулся с проблемой в построении блок схемы, но после чего исправил и всё получилось. Освоил принципы работы в графическом режиме; получил базовые навыки взаимодействия с графическими примитивами.