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

Бирюков Павел Сергеевич

Преподаватель:

Сергеева Елизавета Григорьевна

Киров

2023

1. **Цель работы:**

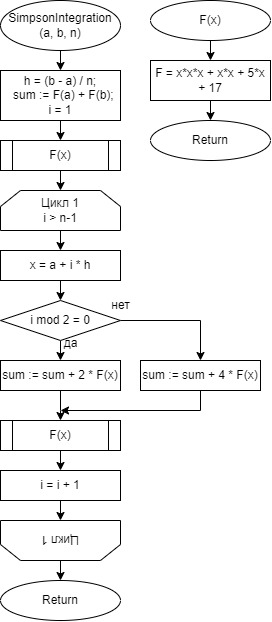
Освоить синтаксис построения процедур и функций, изучить способы передачи данных в подпрограммы, получить навыки организации минимального пользовательского интерфейса.

1. **Формулировка задания:**
2. Реализовать программу вычисления площади фигуры, ограниченной кривой 1 \* x ^ 3 + (1) \* x ^ 2 + (5) \* x + (17) и осью OX (в положительной части по оси OY).
3. Вычисление определенного интеграла должно выполняться численно, с применением метода Симпсона.
4. Пределы интегрирования вводятся пользователем.
5. Взаимодействие с пользователем должно осуществляться посредством case-меню.
6. Требуется реализовать возможность оценки погрешности полученного результата.
7. Необходимо использовать процедуры и функции там, где это целесообразно.
8. **Описание алгоритма:**

В начале дается выбор действия для пользователя (1-3). Вводится предел интеграции (a и b), после вводится кол-во под интервалов (n). Если с начало было введено число 1, то программа будет выводит результат интеграции с помощью численного метода Симпсона. Если было введено число 2, то программа выведет предполагаемую погрешность интеграции. Если будет введено число 3, то программа завершится.

**4. Схема алгоритма:**

|  |  |
| --- | --- |
|  |  |



1. **Код программы:**

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

**begin**

F := x\*x\*x + x\*x + 5\*x + 17;

**end**;

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

**var**

h, x, sum: real;

i: integer;

**begin**

h := (b - a) / n;

sum := F(a) + F(b);

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

**begin**

x := a + i \* h;

**if** i **mod** 2 = 0 **then**

sum := sum + 2 \* F(x)

**else**

sum := sum + 4 \* F(x);

**end**;

SimpsonIntegration := h/3 \* sum;

**end**;

**procedure** CalculateArea;

**var**

a, b: real;

n: integer;

result: real;

**begin**

writeln('Введите пределы интеграции (a и b): ');

readln(a, b);

writeln('Введите количество подинтервалов (n): ');

readln(n);

result := SimpsonIntegration(a, b, n);

writeln('Результатом интеграции Симпсона является: ', result:0:4);

**end**;

**procedure** EvaluateError;

**var**

a, b, n: int64;

result1, result2, error: real;

**begin**

writeln('Введите пределы интеграции (a и b): ');

readln(a, b);

writeln('Введите количество подинтервалов (n): ');

readln(n);

result1 := SimpsonIntegration(a, b, n);

result2 := SimpsonIntegration(a, b, n\*2);

error := (result2 - result1) / 15;

writeln('Предполагаемая погрешность составляет: ', error:0:6);

**end**;

**var**

choice: integer;

**begin**

**repeat**

writeln('1. Вычислить площадь');

writeln('2. Оценить ошибку');

writeln('3. Выход');

write('Выберите вариант: ');

readln(choice);

**case** choice **of**

1: CalculateArea;

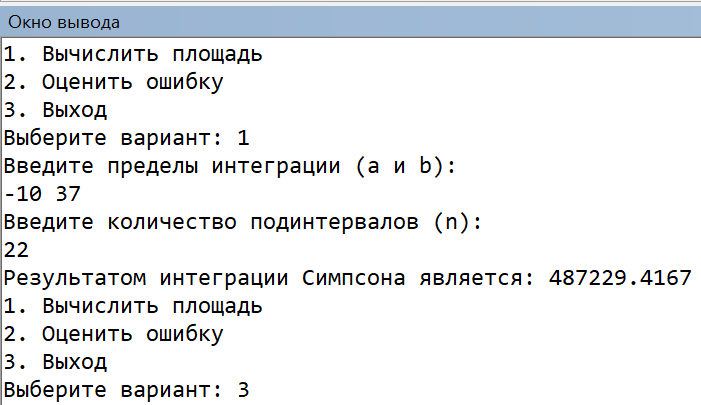
2: EvaluateError;

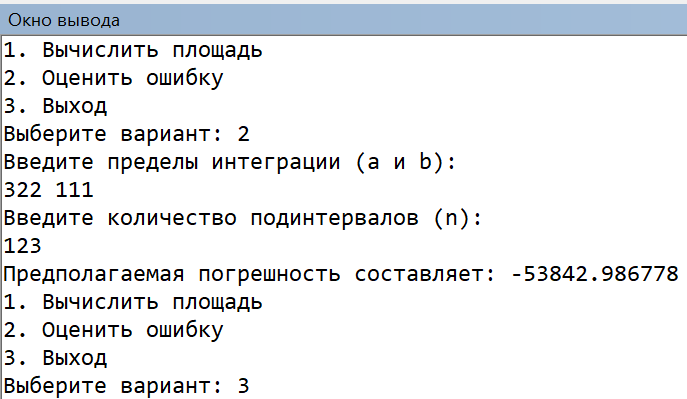
**end**;

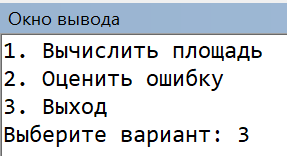
**until** choice = 3;

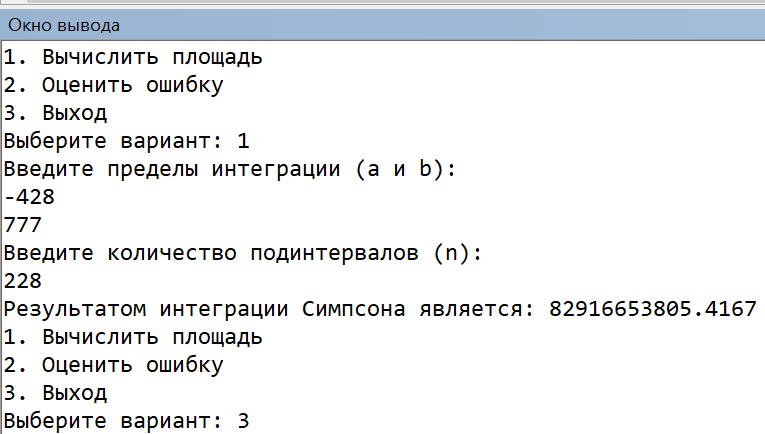
**end**.

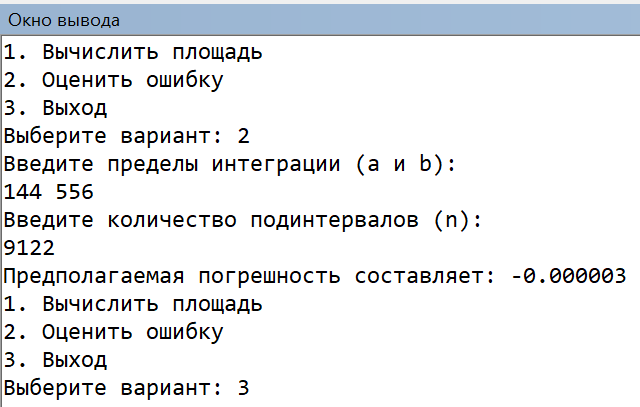
1. **Результат выполнения программы:**





****

****

****

1. **Вывод:**

В завершение скажу что заданную задачу я выполнил. В написание кода были трудности, так как при запуске программы выдавал ошибку в операторе *power*. Решил эту проблему и понял что в *power* при введении выражения с дробным число, нужно ставить «.», а не «,». Еще из-за того что делаю слишком много всего в последние дни, я не могу нормально успокоиться и отдохнуть, за эти полгода я думал что помру скоро от такого, я даже не представляю себе что дальше будет еще намного сложнее, но если бы было по больше времени отдыха (я понимаю что оно у нас по сути есть, но я имел ввиду там хоть несколько дней подряд, чтобы можно было потом собрать с мыслями и освежившейся головой начать снова в полную силу заниматься дальше и выполнять все поставленные для нас задачи). По итогу завершения данной задачи, изучил базовую структуру организации

программы и основные конструкции языка программирования Pascal.