

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных систем  
Факультет автоматики и вычислительной техники  
Кафедра электронных вычислительных машин

Отчёт по лабораторной работе №3  
по дисциплине  
«Программирование»

Выполнил студент гр. ИВТб-1303-06-00	_____ /Гортоломей И.К./
Проверил преподаватель кафедры ЭВМ	_____ /Баташев П.А./

Киров  
2025

## Цель

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

## Задание

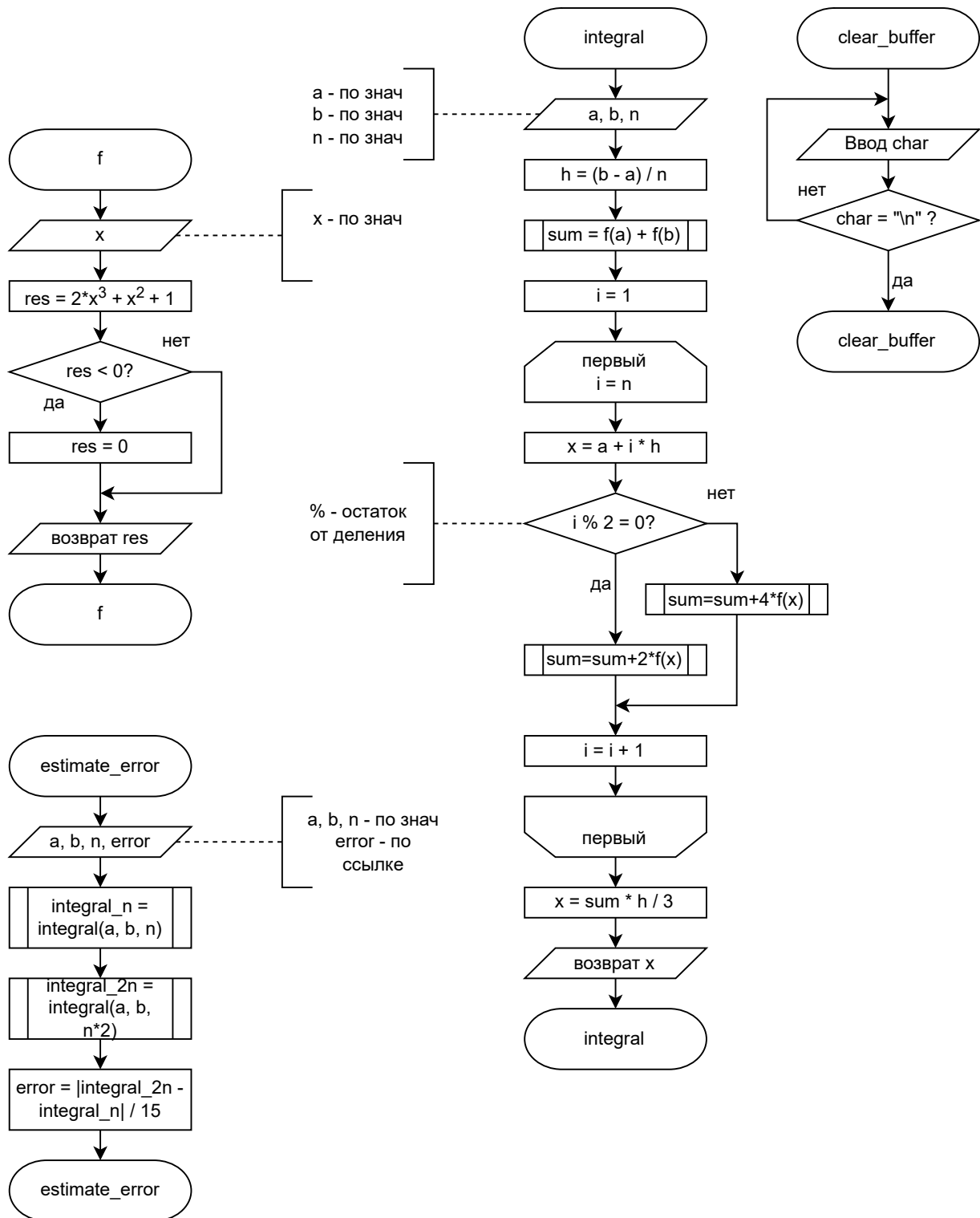
1. Реализовать программу вычисления площади фигуры, ограниченной кривой и осью OX (в положительной части по оси OY).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством case-меню.
5. Требуется реализовать возможность оценки погрешности полученного результата.
6. Необходимо использовать процедуры и функции там, где это целесообразно (программа должна содержать минимум одну процедуру, одну функцию, один пример передачи данных в подпрограммы по ссылке, один пример передачи данных в подпрограммы по значению).

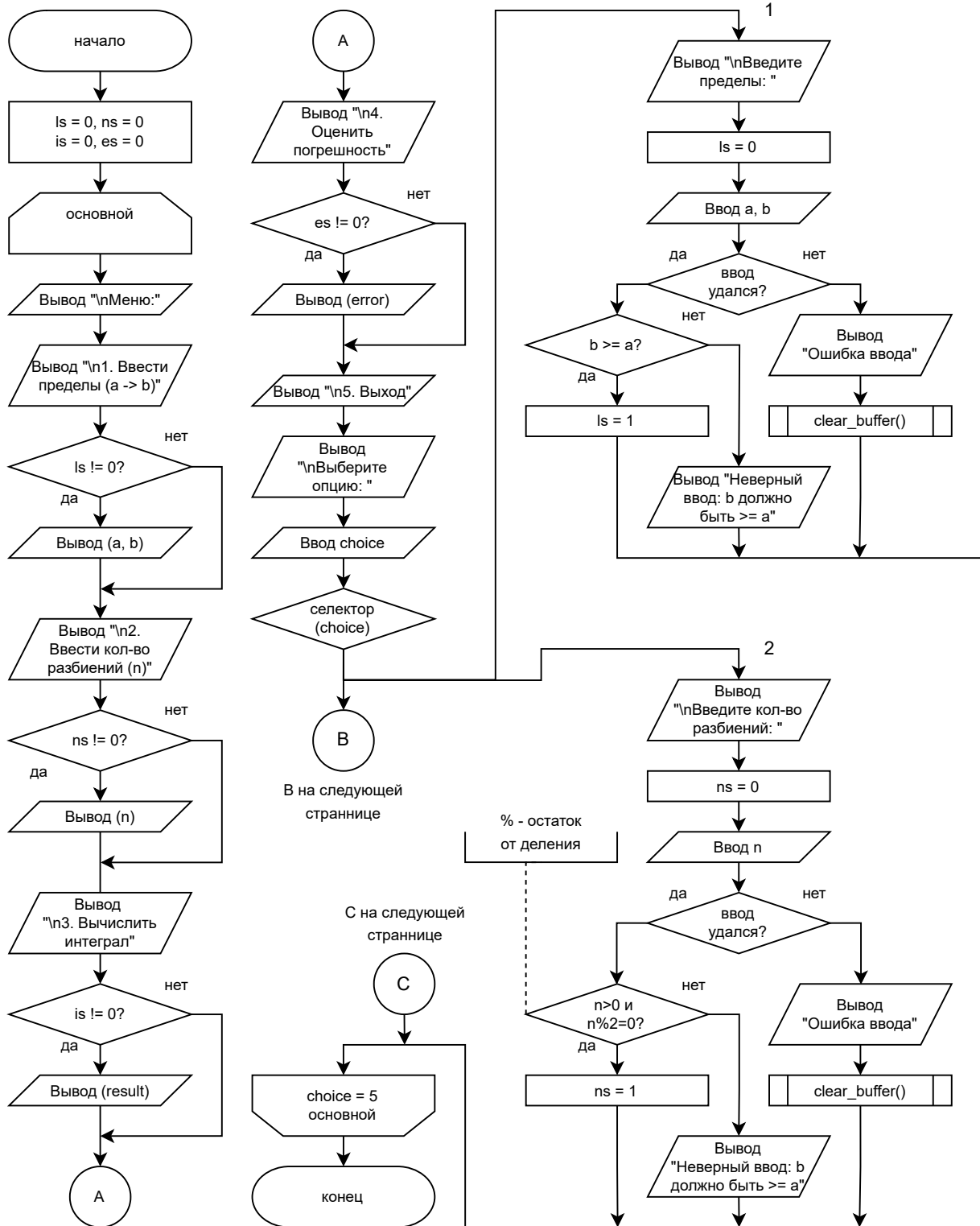
## Дано:

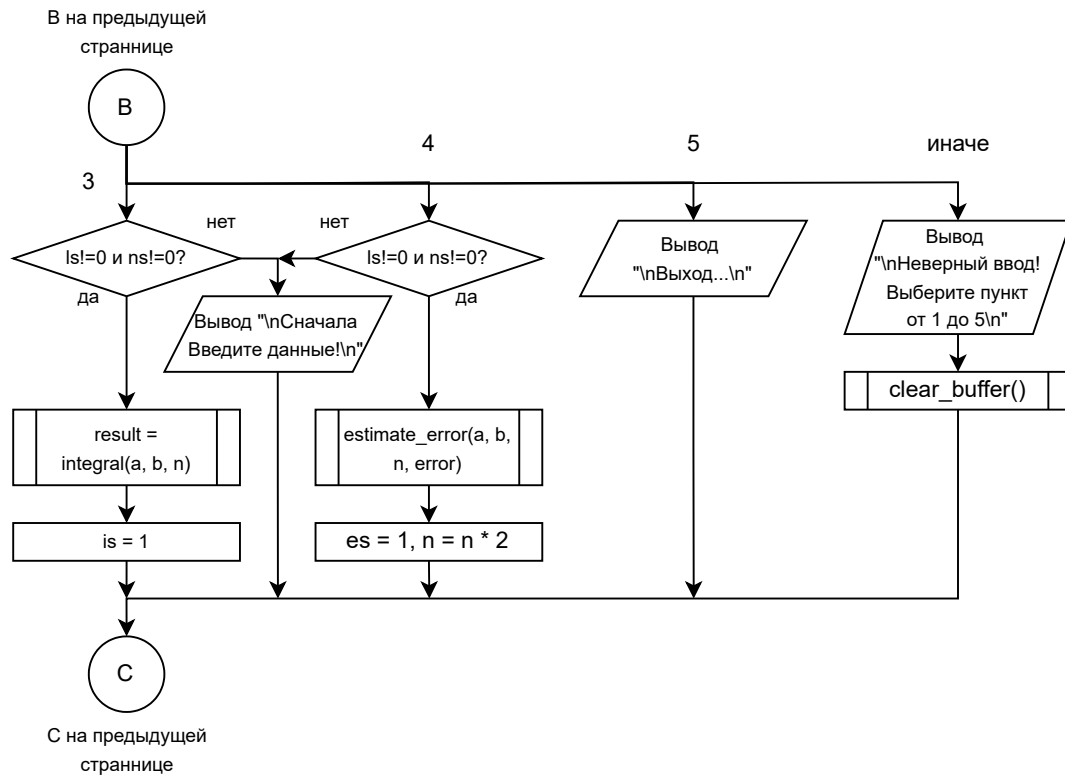
- Уравнение кривой:  $2 * x^3 + 1 * x^2 + 0 * x + 1$
- Метод: Симпсона
- Язык: Си

# Решение

Схема программы:







Код программы на C:

```
#include <math.h>
#include <stdio.h>

// Функция для вычисления значения кривой в точке x
double f(double x) {
    double res = 2 * pow(x, 3) + pow(x, 2) + 1;
    if (res < 0) res = 0; // отбрасывание отрицательного
    return res;
}

// Функция вычисления интеграла методом Симпсона
double integral(double a, double b, int n) {
    double h = (b - a) / n; // шаг
    double sum = f(a) + f(b); // сумма концов

    for (int i = 1; i < n; i++) {
        double x = a + i * h;
        if (i % 2 == 0) sum += 2 * f(x);
        else sum += 4 * f(x);
    } return sum * h / 3;
}

// Процедура для оценки погрешности методом Рунге
void estimate_error(double a, double b, int n, double* error) {
    double integral_n = integral(a, b, n);
    double integral_2n = integral(a, b, n * 2);
    *error = fabs(integral_2n - integral_n) / 15;
}
```

```

void clear_buffer() {
    while (getchar() != '\n');
}

int main() {
    double a, b, result, error;
    int ls = 0, ns = 0, is = 0, es = 0, x, n, choice;
    do {
        printf("\nМеню:");
        printf("\n1. Ввести пределы (a -> b)");
        if (ls) printf(" (%f, %f)", a, b);

        printf("\n2. Ввести кол-во разбиений (n)");
        if (ns) printf(" (%d)", n);

        printf("\n3. Вычислить интеграл");
        if (is) printf(" (%.6f)", result);

        printf("\n4. Оценить погрешность");
        if (es) printf(" (%.6f)", error);

        printf("\n5. Выход");
        printf("\nВыберите опцию: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("\nВведите пределы: ");
                ls = 0;

```

```

if (scanf("%lf %lf", &a, &b) == 2) {
    if (b >= a) ls = 1;
    else printf("Неверный ввод: b должно быть >= a\n");
} else {
    printf("Ошибка ввода\n");
    clear_buffer();
}; break;

case 2:
    printf("\nВведите кол-во разбиений: ");
    ns = 0;
    if (scanf("%d", &n) == 1) {
        if (n > 0 && n % 2 == 0) ns = 1;
        else printf("Неверный ввод: n должно быть >0 и чётным\n");
    } else {
        printf("Ошибка ввода\n");
        clear_buffer();
    }; break;

case 3:
    if (ls && ns) {
        result = integral(a, b, n);
        is = 1;
    } else printf("\nСначала Введите данные!\n");
    break;

case 4:
    if (ls && ns) {
        estimate_error(a, b, n, &error);
        es = 1;
    }

```



```

        n *= 2; // Увеличиваем n для следующего вычисления
    } else printf("\nСначала Введите данные!\n");
    break;

case 5:
    printf("\nВыход...\n");
    break;

default:
    printf("\nНеверный ввод! Выберите пункт от 1 до 5\n");
    clear_buffer();
}
} while (choice != 5);
return 0;
}

```

Примеры работы программы:

```
Меню:  
1. Ввести пределы интегрирования (a, b)  
2. Ввести количество разбиений n  
3. [Недоступно - введите данные]  
4. [Недоступно - введите данные]  
5. Выход  
Выберите опцию: █
```

```
Меню:  
1. Ввести пределы интегрирования (a, b) (-5.000000, 5.000000)  
2. Ввести количество разбиений n (2)  
3. Вычислить интеграл  
4. Оценить погрешность  
5. Выход  
Выберите опцию: █
```

```
Меню:  
1. Ввести пределы интегрирования (a, b) (-5.000000, 5.000000)  
2. Ввести количество разбиений n (4)  
3. Вычислить интеграл (93.333333)  
4. Оценить погрешность (0.000000)  
5. Выход  
Выберите опцию: █
```

## Выводы

Во время выполнения лабораторной работы научился писать функции, передавать в них аргументы по ссылке или значению и возвращать из них результат. В программе был реализован простой пользовательский интерфейс в котором пользователь вводит пределы и разбиение. Также имеется функция для вычисления интеграла и оценки ошибки. Взаимодействие с пользователем реализованно в форме case-меню.