

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 ОХ (в положительной части по оси ОY).
2. Вычисление определенного интеграла должно выполняться численно, с применением метода.
3. Пределы интегрирования вводятся пользователем.
4. Взаимодействие с пользователем должно осуществляться посредством 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

Код программы на C:

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

// Функция для очистки экрана
void clear(void) {
#ifndef _WIN32
    system("cls");
#else
    system("clear");
#endif
}

// Функция для вычисления значения кривой в точке x
double f(double x) {
    return 2 * pow(x, 3) + pow(x, 2) + 1;
}

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

    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;
}
```

```

    sum += 2 * f(x);

else
    sum += 4 * f(x);

}

return sum * h / 3;
}

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

int main() {
    double a = 0, b = 0, result, error;
    int n = 0, choice;
    int limits_set = 0, n_set = 0, integral_set = 0, error_set = 0;

    do {
        printf("\nМеню:\n");
        printf("1. Ввести пределы интегрирования (a, b)");
        limits_set ? printf(" (%f, %f)\n", a, b) : printf("\n");
        printf("2. Ввести количество разбиений n");
        n_set ? printf(" (%d)\n", n) : printf("\n");
        if (limits_set && n_set) {
            printf("3. Вычислить интеграл");
            integral_set ? printf(" (%.6f)\n", result) : printf("\n");
            printf("4. Оценить погрешность");
            error_set ? printf(" (%.6f)\n", error) : printf("\n");
        }
    }
}

```

```

printf("5. Выход\n");
} else {
    printf("3. [Недоступно - введите данные]\n");
    printf("4. [Недоступно - введите данные]\n");
    printf("5. Выход\n");
}

printf("Выберите опцию: ");
scanf("%d", &choice);
clear();

switch (choice) {
    case 1:
        printf("Введите пределы интегрирования (a b): ");
        if (scanf("%lf %lf", &a, &b) == 2) {
            limits_set = 1;
            // printf("Пределы установлены: a=%.2f, b=%.2f\n", a, b);
        } else {
            printf("Ошибка ввода!\n");
            while (getchar() != '\n'); // Очистка буфера
        }
        break;

    case 2:
        printf("Введите количество разбиений n: ");
        if (scanf("%d", &n) == 1 && n > 0) {
            n_set = 1;
        } else {
            printf("Ошибка ввода! n должно быть положительным числом.\n");
            while (getchar() != '\n'); // Очистка буфера
        }
}

```

```

break;

case 3:
    if (limits_set && n_set) {
        result = simpson_integral(a, b, n);
        integral_set = 1;
    } else printf("Сначала введите данные!\n");
    break;

case 4:
    if (limits_set && n_set) {
        estimate_error(a, b, &n, &error);
        error_set = 1;
        n *= 2; // Увеличиваем n для следующего вычисления
    } else printf("Сначала введите данные!\n");
    break;

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

default:
    printf("Неверный ввод! Выберите пункт от 1 до 5.\n");
}

} 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**