|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 2 | | |
| по дисциплине «Основы программирования» | | |
|  | | |
| **Циклы** | | |
|  | | |
|  | Факультет: | ФПМИ |
| Группа: | ПМИ-12 |
|  |  |
| Студент: | Мироненко Алиса |
|  |
|  |
| Преподаватель: | Филиппова Елена Владимировна |
|  |  |
|  | | |
| Новосибирск | | |
| 2021 | | |

**1. Условие задачи**

Даны фиксированные границы интегрирования. Вычислить интеграл по методу Симпсона.

Изображение выглядит как текст

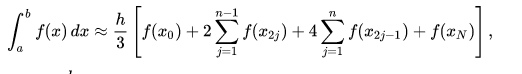
Автоматически созданное описание

**2. Анализ задачи**

**Дано:** Точность интегрирования (eps). Границы интегрирования: a = 0, b = 1. Начальное значение n = 2.

**Результат:** Программа выведет на экране значение интеграла с заданной точностью.

**Решение:** Решение задачи сводилось к преобразованию исходной формулы расчета интеграла, и её записи в программе.



Используем различные функции для расчета:  
1. Функция, задающая сам интеграл: **double f.** Аргументы функции: (double x).

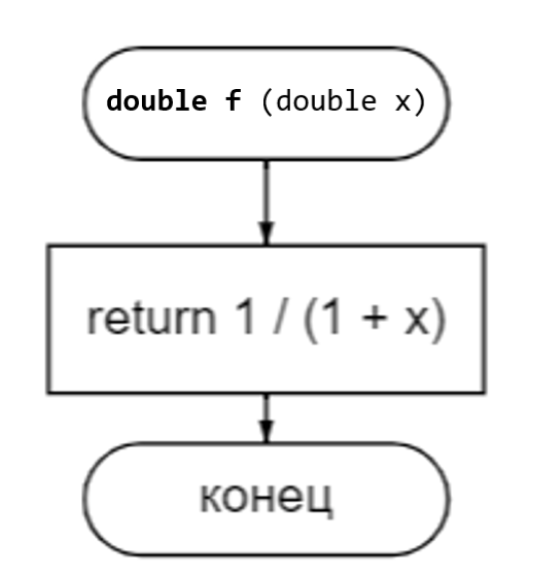
2. Расчет приближенного значения интеграла по формуле Симпсона:   
**double Sympson.** Аргументы функции: (double a, double b, int n).  
Также используем циклы для расчета

1. Цикл для подсчета значений интеграла (кроме sum = f(a) + f(b)): **for** (int i = 0; i <= n - 1; i++).

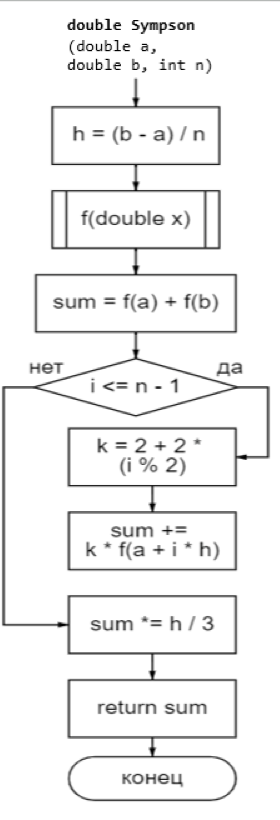
2. Цикл для подсчета значения интеграла с заданной точностью: **do** {…} while (diff > eps).

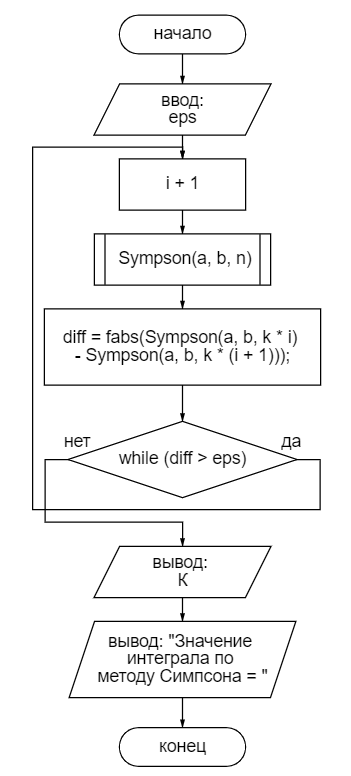
**3. Алгоритм решения задачи**

**«f» -** функция, задающая сам интеграл. Аргументы функции: (double x).

****

**«Sympson» -** расчет приближенного значения интеграла по формуле Симпсона. Аргументы функции: (double a, double b, int n).

****



**4. Тело программы.**

#include <stdio.h>

#include <locale.h>

#include <math.h>

#include <stdlib.h>

double f(double x) { //функция, определяющая сам интеграл

return 1 / (1 + x);

}

double Sympson(double a, double b, int n) { //функция для подсчета площади интеграла

double h = (b - a) / n; //находим шаг h

double sum = f(a) + f(b); //сумма начально и конечного значения

int k; //к нужно как определитель четности/нечетности индекса при f

for (int i = 0; i <= n - 1; i++) { //цикл для подсчета остальных частей интеграла, кроме sum = f(a) + f(b)

k = 2 + 2 \* (i % 2); //проверка четности

sum += k \* f(a + i \* h); //суммируем значения f(x) при всех х

}

sum \*= h / 3; //согласно формуле, домножаем сумму на h / 3

return sum; //возвращаем значение суммы площадей интеграла

}

int main()

{

setlocale(LC\_CTYPE, "Russian");

double a, b; //переменные для начала и конца интеграла

long double eps; //переменная, определяющая допустимую погрешность

printf("Введите значение eps =\n");

scanf\_s("%lf", &eps);

int n;

a = 0;

b = 1;

int k = 2; //начальное число разбиений интеграла на части

int i = 0;

double diff; //переменная для вычисления разницы между соседними измерениями значения интеграла при разных значениях к

do {

i++;

diff = fabs(Sympson(a, b, k \* i) - Sympson(a, b, k \* (i + 1))); //рассчет модуля разности между соседними измерениями значения интеграла при разных значениях к

} while (diff > eps); //пока разность больше допустимой погрешности цикл продолжается

printf("K = %d\n", k \* (i + 1)); //вывод К (числа разбиений)

printf("Значение интеграла по методу Симпсона = %f", Sympson(a, b, k \* (i + 1))); //вывод значения интеграла

return 0;

}

**5. Тесты.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Дано** | **Результат** | **K** | **Примечание** |
| **eps** |
| **1** | 1 | 0.859921 | 4 | Грубая точность |
| **2** | 0.0001 | 0.698797 | 118 | Средняя точность |
| **3** | 0.00000001 | 0.693205 | 11550 | Высокая точность |

**6. Результат работы.**

На всех тестах программа выдала ожидаемый результат, следовательно, задача решена верно.