**Московский государственный технический**

**университет им. Н. Э. Баумана.**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Основы программирования»

Отчет по лабораторной работе № 6

«Численное интегрирование функции»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-12Б |  | преподаватель каф. ИУ5 |
| Поляков Леонид |  | Правдина А. Н. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2023 г.

**Постановка задачи**

Вычислить определённый интеграл \int_{a}^{b} f(x) \,dx в пределах от a до b для четырех функций:

* f_1 = x
* f_2 = \sin(22 x)
* f_3 = x^4
* f_4 = \arctan(x)

Численное интегрирование функции с заданной точностью выполнить двумя методами:

* метод прямоугольников. Вычисление интеграла оформить в виде функции IntRect;
* метод трапеций. Вычисление интеграла оформить в виде функции IntTrap.

Вычисления выполнить для пяти значений точности: 0.01, 0.001, 0.0001, 0.00001 и 0.000001.

Исследовать быстродействие алгоритма в зависимости от подынтегральной функции и требуемой точности (быстродействие алгоритма можно оценить числом элементарных прямоугольников n).

Для каждого метода представить результаты вычислений в виде 5 таблиц, по одной таблице для каждого значения точности. В каждой таблице выводить данные для всех четырех функций.

Для печати таблицы результатов использовать функцию

void PrintTabl(I\_print i\_prn[], int k)

Здесь i\_prn[] – массив структур типа I\_print размерностью k

**Разработка алгоритма**

Подсчёт интеграла производят функции

double IntRect(func\_ref\_1, double, double, double, unsigned&);

double IntTrap(func\_ref\_1, double, double, double, unsigned&);

они принимают на вход указатель на функцию, интеграл которой надо посчитать, нижнюю и верхнюю границы интегрирования, требуемую точность подсчёта интеграла и начальное количество разбиений (выходной параметр). Указатели на функции, считающие интеграл содержаться в массиве const func\_ref\_3 IntСalculationFuncs[2]. Указатели на функции, считающие точное значение интеграла и значение интегрируемой функции в точке находятся соответственно в массивах const func\_ref\_2 functions\_exact[4] и const func\_ref\_1 functions[4]. Границы интегрирования запрашиваются у пользователя с клавиатуры. Затем во внешнем цикле программа итерируется по массиву с методами подсчёта интеграла, считая для каждого метода интегралы функций, ссылки на которые находятся в массиве functions[4] для пяти разных значений точности. Результаты вычислений хранятся в массиве структур I\_print i\_pr[4], каждый элемент которого представляет собой структуру с полями хранящими название функции, точное значение интеграла этой функции, значение интеграла вычисленное суммированием, и число понадобившихся разбиений для достижения требуемой точности.

**Struct I\_pnint{}**

struct I\_print { //данные для печати результатов интегрирования

const char\* function\_name; //название функции

double integral\_sum; //значение интегральной суммы

double exact\_value; //точное значение интеграла

unsigned n; //число разбиений области интегрирования, при котором достигнута требуемая точность

};

functions[4]

const func\_ref\_1 functions[4]{ F\_x, F\_sin, F\_pow\_x, F\_arctan };

//Хранит указатели на функции, которые интегрируются

functions\_exact[4]

const func\_ref\_2 functions\_exact[4]{ X\_exact, Sin\_exact, PowX\_exact, Arctan\_exact };

//Хранит указатели на функции, которые считают точное значение интеграла

IntСalculationFuncs[2]

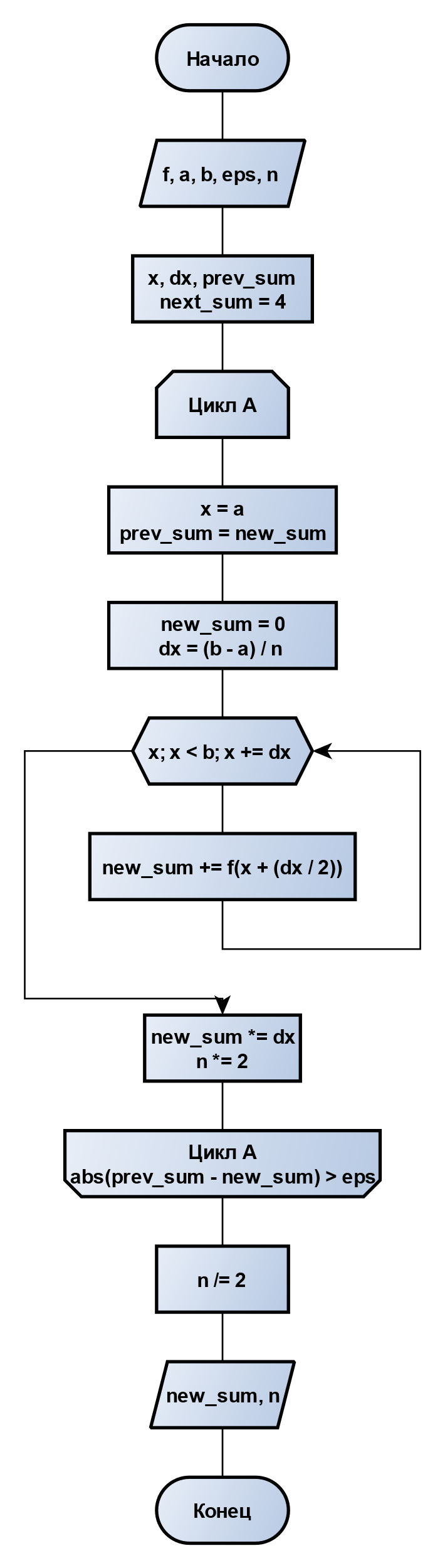
const func\_ref\_3 IntСalculationFuncs[2]{ IntRect, IntTrap };

//Хранит указатели на функции, считающие интеграл

void PrintTabl(I\_print i\_prn[], int arr\_size)

* i\_prn[] //массив структур типа I\_print
* arr\_size // размер массива I\_print (количество функций, которые интегрируем)

Функция double IntRect(…)

double IntRect(func\_ref\_1 f, double a, double b, double eps, unsigned& n)

f – ссылка на интегрируемую функцию

a – нижний предел интегрирования

b – верхний предел интегрирования

eps – требуемая точность, с которой надо вычислить интеграл

&n – начальное число разбиений отрезка [a, b]

x – текущее значение x на отрезке [a, b]

dx – приращение x на каждом шаге

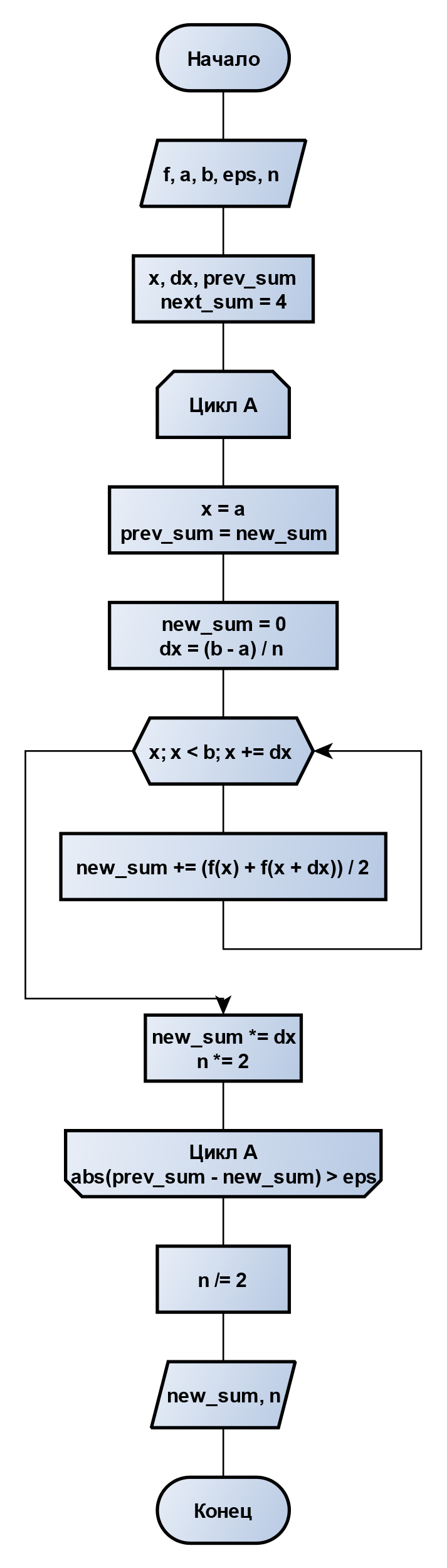
prev\_sum – старое значение интеграла

new\_sum – новое значение интеграла на каждой итерации

Функция IntRect считает на каждом шаге новое значение интеграла методом прямоугольников для уменьшенного вдвое dx;

если разность старого и нового значения интеграла меньше eps, то вычисления прекращаются и функция возвращает последнее посчитанное значение интеграла, в противном случае последнее вычисленное значение записывается в prev\_sum, а новое значение считается для удвоенного числа разбинений.

Функция double IntTrap(…)

double IntTrap(func\_ref\_1 f, double a, double b, double eps, unsigned& n)

f – ссылка на интегрируемую функцию

a – нижний предел интегрирования

b – верхний предел интегрирования

eps – требуемая точность, с которой надо вычислить интеграл

&n – начальное число разбиений отрезка [a, b]

x – текущее значение x на отрезке [a, b]

dx – приращение x на каждом шаге

prev\_sum – старое значение интеграла

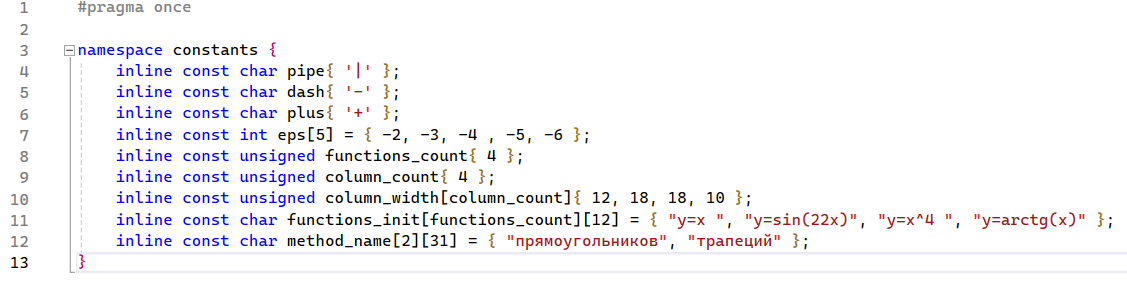
new\_sum – новое значение интеграла на каждой итерации

Функция IntTrap считает на каждом шаге новое значение интеграла методом прямоугольников для уменьшенного вдвое dx;

если разность старого и нового значения интеграла меньше eps, то вычисления прекращаются и функция возвращает последнее посчитанное значение интеграла, в противном случае последнее вычисленное значение записывается в prev\_sum, а новое значение считается для удвоенного числа разбинений.

**Текст программы**

**constants.h**



**functions.hИзображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

**functions.cpp** **Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание** Изображение выглядит как текст, снимок экрана, Шрифт, число

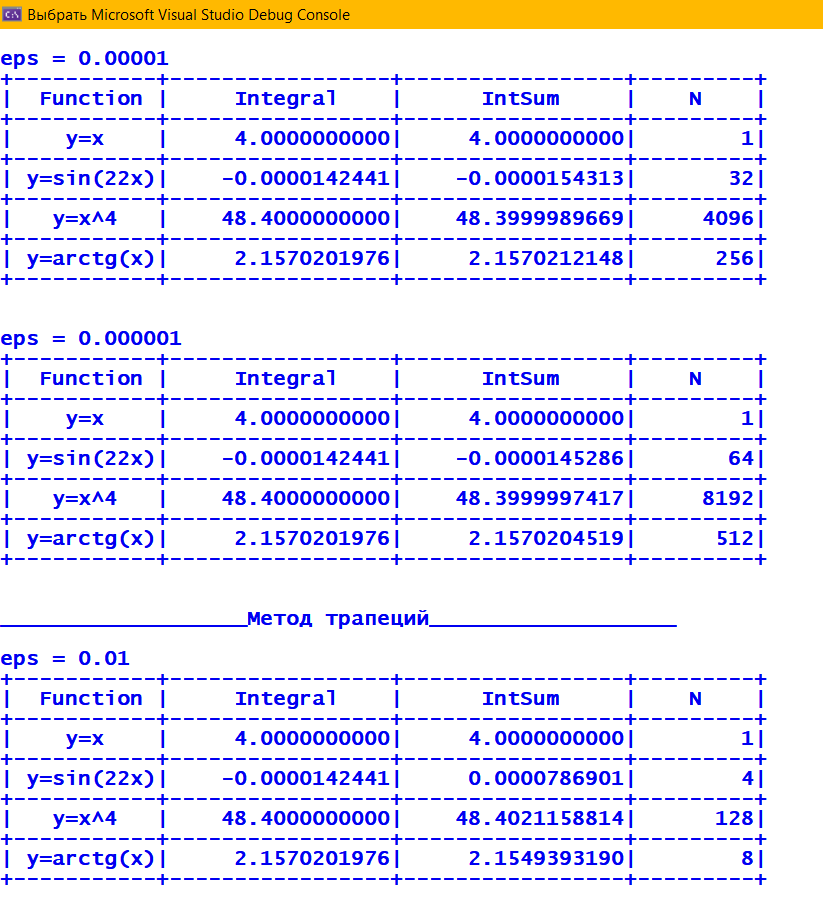
Автоматически созданное описание

**main.cpp**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Анализ результатов**

Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание****

**Изображение выглядит как текст, снимок экрана, число, Шрифт

Автоматически созданное описание**

**Результат работы программы соответствует ожиданиям и полностью удовлетворяет требованиям ТЗ**

**Использованные источники**

* **Веб-сайт кафедры ИУ5, курс «основы программирования»**

[**https://cpp1.wiki.iu5edu.ru**](https://cpp1.wiki.iu5edu.ru)

* **Веб-сайт кафедры ИУ5 «автоматизированные системы обработки информации и управления»**

**https://e-learning.bmstu.ru/iu5/mod/folder/view.php?id=277**