**Московский авиационный институт (национальный исследовательский университет)**

Институт №3. Факультет «Системы управления, информатика и электроэнергетика». Кафедра №304

Практическое задание

Группа М30-107Б-18

Выполнил:

Кривонос Александр

Принял:

Чечиков Юрий Борисович

Москва 2018

**Содержание**

1. Задание………………………………………………………………………………….2

2. Описание метода……………………………………………………………………….3

2. Блок-схема алгоритма…………………………………………………………………4

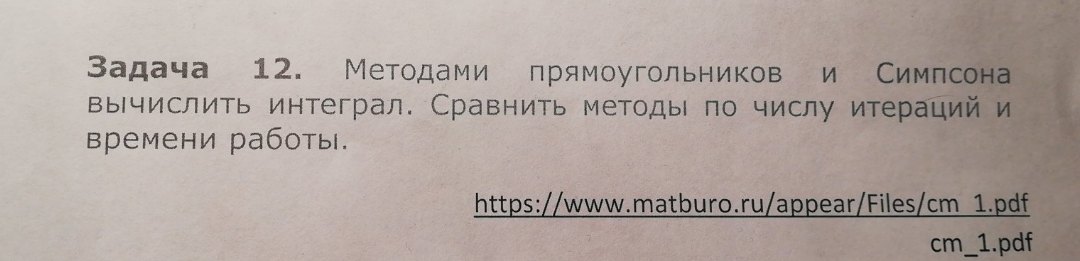
3. Псевдокод……………………………………………………………………………....5

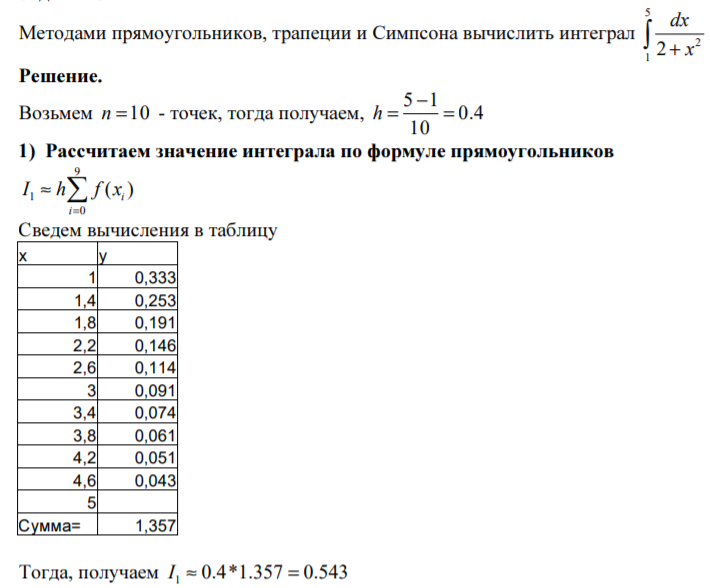
4. Текст программы………………………………………………………………………10

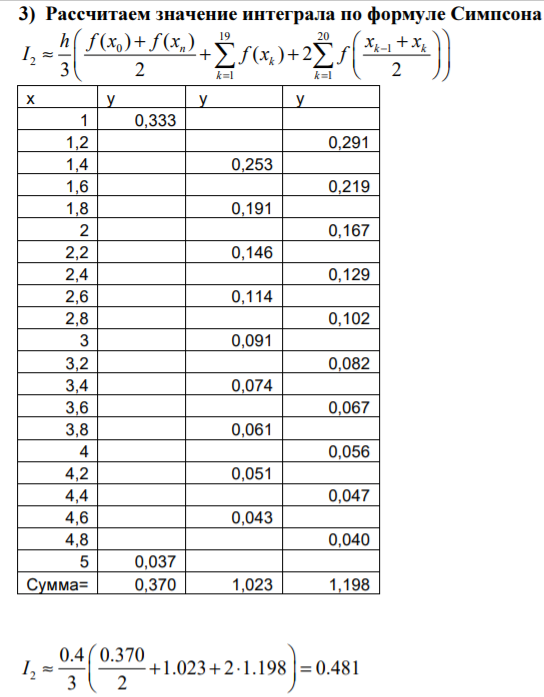
5. Тест (некорректный и корректный)………………………………………………….14

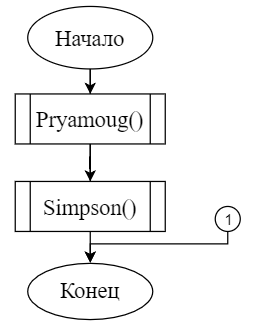
6. Вывод…………………………………………………………………………………..16

**Задание**



**Описание методов (с сайта)**

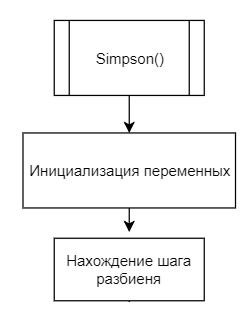


**Блок-схема алгоритма**

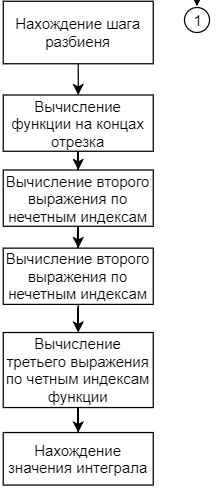


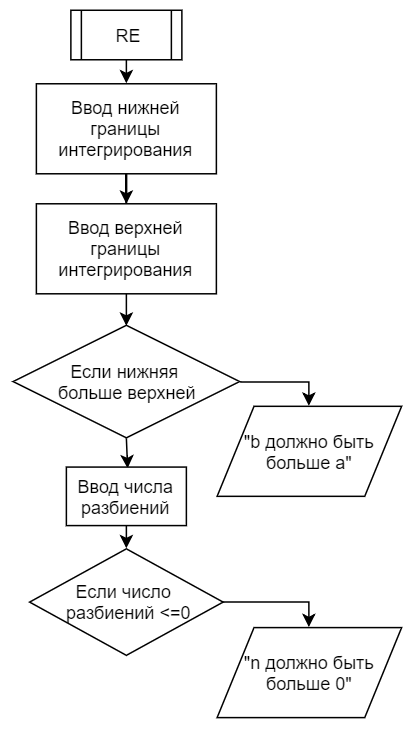
Описание функции Pryamoug:

1. Назначение: Вычисляет приближенное значение интеграла по методу прямоугольников
2. Прототип: double Pryamoug()
3. Обращение: Pryamoug()
4. Блок-схема

Описание функции Simpson:

1. Назначение: Вычисляет приближенное значение интеграла по методу прямоугольников
2. Прототип: double Simpson()
3. Обращение: Simpson()
4. Блок-схема



Описание функции RE:

1. Назначение: ввод границ интегрирования и их проверка
2. Прототип: int RE(double &a, double &b, int &n)
3. Обращение: RE(a, b, n)
4. Блок-схема

**Текст программы**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Информатика и Вычислительная техника \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Language: cpp, MSVS 2010 and above \*

\* Programmers: M3O-107Б-18 \*

\* Кривонос Александр Александрович \*

\* \*

\* Created: 17.03.2019 Last revision: 20.03.2019 \*

\* Comment: Вычисление интеграла разными методами \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <math.h>

#include <ctime>

using namespace std;

int RE(double &a, double &b, int &n)

{

cout << "Введите нижнюю граница интегрирования а: ";

cin >> a;

cout << "Введите верхнюю граница интегрирования b: ";

cin >> b;

if (a >= b)

{

cout << "b должно быть болше a\n";

system("Pause");

return 1;

}

cout << "Введите число разбиений: ";

cin >> n;

cout << endl;

if (n <= 0)

{

cout << "n должно быть больше 0\n";

system("Pause");

return 2;

}

return 0;

}

double Pryamoug(int RE(double &a, double &b, int &n))

{

double a; //нижняя граница интегрирования

double b; //верхняя граница интеграла

double h; //шаг разбиения

int n; //число разбиений

double func;// функция

double x = 0.0; //переменная

int i=0; //число итераций

double Sum=0;//Сумма значений функций

double I; //значение интеграла

cout << "Вычисление по методу прямоугольников" << endl;

RE(a, b, n);

int start = clock();

h = (b - a) / n; //нахождение шага разбиения

cout << "Шаг разбиения: " << h << endl;

/\*Вычисление функции на отрезке\*/

for (x = a; x <= b-h; x = x + h)

{

i++;

func = 1 / (2 + x \* x);

cout << i << ". " << func << endl;

Sum = Sum + func;

}

I = h \* Sum;

cout << "Значение интеграла: " << I << endl;

int end = clock();

cout << "Время работы: " << end - start <<" милисекунд"<< endl;

return I;

}//Pryamoug

double Simpson(int RE (double &a, double &b, int &n))

{

double a; //нижняя граница интегрирования

double b; //верхняя граница интеграла

double h; //шаг разбиения

int n; //число разбиений

double func;// функция

double x = 0.0; //переменная

int i = 1; //число итераций

double a1=0; //первое выражение

double a2=0; //второе выражение

double a3=0; //третье выражение

double x1 = 0; //первая переменная

double xn = 0; //последняя переменная

double Sum = 0;//Сумма значений функций

double I; //значение интеграла

cout << "Вычисление по методу Симпсона" << endl;

RE(a, b, n);

int start = clock();

h = (b - a) / (2\*n); //нахождение шага разбиения

cout << "Шаг разбиения: " << h << endl;

x1 = 1 / (2 + a \* a);

xn = 1 / (2 + b \* b);

a1 = (x1 + xn)/2;

cout << i << "." << x1 << endl;

i++;

cout << "Вычисление второго выражения по нечетным индексам:" << endl;

/\*Вычисление второго выражения по нечетным индексам\*/

for (x = (a + 2 \* h); x <= (b - 2\*h); x = (x + 2 \* h))

{

func = 1 / (2 + x \* x);

cout << i << "." << func << " " << endl;

i++;

a2 = a2 + func;

}

cout << "Вычисление третьего выражения по нечетным индексам:" << endl;

/\*Вычисление третьего выражения по четным индексам функции\*/

for (x = (a + h); x <= (b - h); x = x + 2 \* h)

{

func = 1 / (2 + ((x \* x - h) + x \* x)/2);

cout << i << "." << func << endl;

i++;

a3 = a3 + func;

}

cout << i << "." << xn << endl;

Sum = (a1 + a2 + 2\*a3);

I = (2 \* h \* Sum) / 3;

cout << "Значение интеграла: " << I << endl;

int end = clock();

cout << "Время работы: " << end - start << " милисекунд" << endl;

return I;

}//Simpson

int main()

{

setlocale(LC\_ALL, "Russian");

Pryamoug(RE);

Simpson(RE);

system("Pause");

return 0;

}// main()

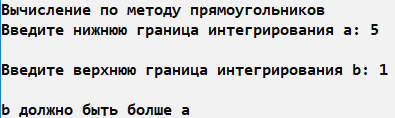
**Тесты (корректные и некорректные)**

**1.Некорректные**

1. Цель: проверить работу программы при некорректных значениях границ интегрирования

Исходные данные: a = 5, b = 1

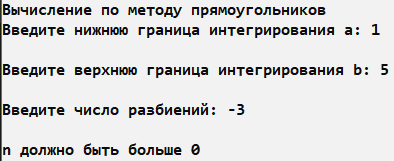
Ожидаемый результат: “b должно быть больше a”



1. Цель: проверить работу программы при некорректном числе разбиений

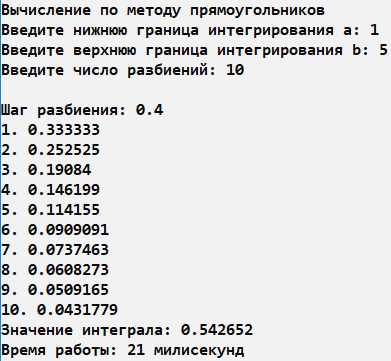
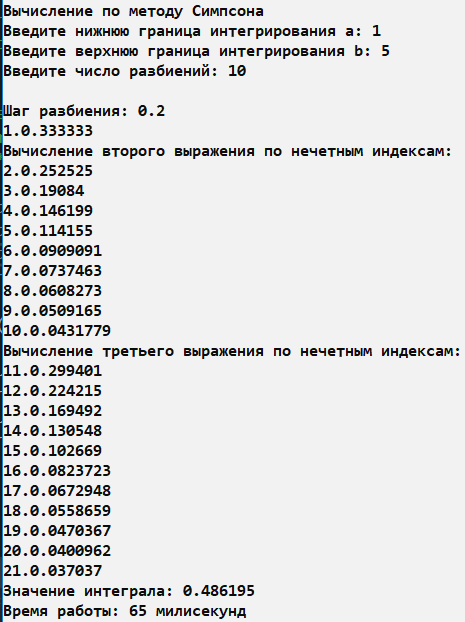
Исходные данные: n = -3

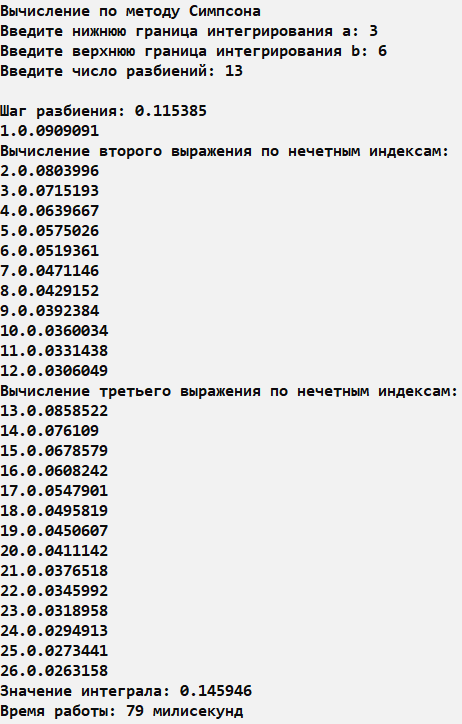
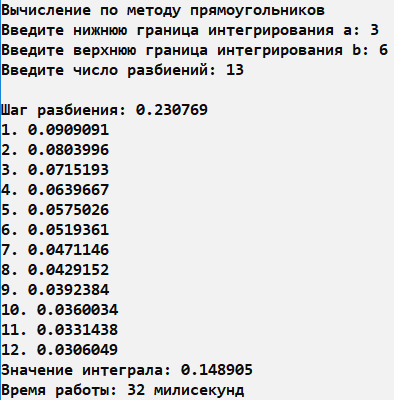
Ожидаемый результат: “n должно быть больше 0”



1. **Корректные**
2. Цель: проверить работу программы при корректных значениях

Исходные данные: a = 1, b = 5, n = 10



Исходные данные: a = 3, b = 6, n = 13

**Анализ**

Таким образом, видно, что алгоритм прямоугольников легче (по количеству итераций) и работает быстрее, но алгоритм Симпсона вычисляет более точное значение интеграла

**Вывод**

Разработка программы завершена на том основании, что

1) Полученные результаты совпали с ожидаемыми.

2) Считаем, что набор тестов полный.