ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ

УНІВЕРСИТЕТ РАДІОЕЛЕКТРОНІКИ

Кафедра Інформатики

**Звіт**

з лабораторної роботи № 5

на тему «інтерполяція»

Виконав Перевірив

ст.гр.ІТІНФ-20-1 ас.каф. Інформатики

Самченко С.О. Пономаренко Р.П.

Харків 2021

**Мета роботи:** Застосування чисельних методів обчислення інтегралів у різноманітних практичних задачах. Аналіз та порівняння результатів.

ВАРІАНТ 21

Метод трапецій:

#include <iostream>

#include <cmath>

using namespace std;

// Метод трапеций

double f(double x) { return (-6 \* pow(x, 4) + 5 \* pow(x, 3) + 7 \* pow(x, 2) + 4 \* x - 2); } // основное уравнение

double I(double a, double b, int n, double y) { return ((b - a) / (2 \* n) \* y); } // взятие интеграла

double dx(double x) { return (-72 \* pow(x,2) + 30 \* x + 14); } // вторая производная

double R(double a, double b, double h) { return (-1 \* ((b - a) / 24) \* pow(h, 2) \* dx(b)); } // остаточный член квадратурной формулы

int main() {

setlocale(LC\_ALL, "");

int n; double a(0), b(2), y(0), h, In, Rn;

cout << "границы интегрирования: " << "[ " << a << " ; " << b << " ]\n";

cout << "введите количество отрезков (n): ";

cin >> n;

if (n > 1) {

h = (b - a) / n;

y += f(a) + f(b);

for (int i = 1; i < n; i++) {

y += 2 \* (f(a + h \* i));

}

In = I(a, b, n, y);

Rn = R(a, b, h);

cout << "интеграл от [ " << a << " ; " << b << " ] = " << In << endl;

cout << "остаточный член квадратурной формулы = " << Rn << endl;

cout << "ответ: " << In + Rn << endl;

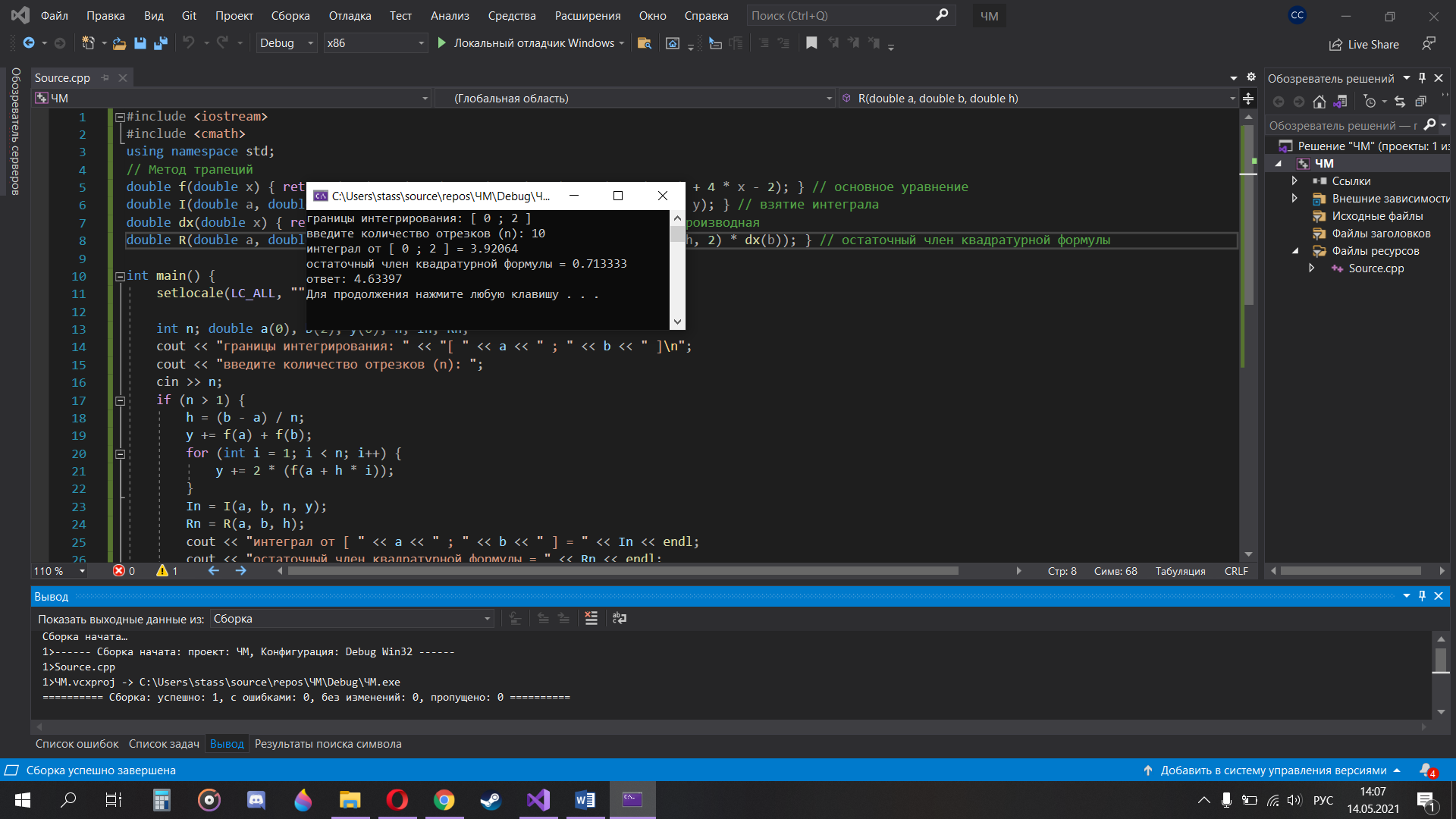
}

else { cout << "!ОШИБКА! не сущестует решения с таким кол-вом отрезков\n\n"; }

system("pause");

return 0;

}



Метод Симпсона:

#include <iostream>

#include <cmath>

using namespace std;

typedef double(\*pointFunc)(double);

// Метод Симпсона

double f(double x) { return (-6 \* pow(x, 4) + 5 \* pow(x, 3) + 7 \* pow(x, 2) + 4 \* x - 2); } // основное уравнение

double R(double a, double b, double h) { return (-1 \* ((b - a) / 180) \* pow(h, 4) \* -144); } // остаточный член квадратурной формулы

double I(pointFunc f, double a, double b, int n) { // взятие интеграла

double k1 = 0, k2 = 0;

double h = (b - a) / n;

for (int i = 1; i < n; i += 2) {

k1 += f(a + i \* h);

k2 += f(a + (i + 1) \* h);

}

return h / 3 \* (f(a) + 4 \* k1 + 2 \* k2);

}

int main() {

setlocale(LC\_ALL, "");

double a(0), b(2), eps(0.00001), In, I1, Rn;

int n;

cout << "границы интегрирования: " << "[ " << a << " ; " << b << " ]\n";

cout << "введите начальное число шагов (n): ";

cin >> n;

double h = (b - a) / n;

In = I(f, a, b, n); // первое приближение для интеграла

do { // второе приближение

I1 = In;

n = 2 \* n;

In = I(f, a, b, n);

} while (fabs(In - I1) > eps);

Rn = R(a, b, h);

cout << "интеграл от [ " << a << " ; " << b << " ] = " << In << endl;

cout << "остаточный член квадратурной формулы = " << Rn << endl;

cout << "ответ: " << In + Rn << endl;

system("pause");

return 0;

}

