МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 2

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Доцент                                                               С.Л. Козенко

должность, уч. Степень, звание           подпись, дата                     инициалы, фамилия

ОТЧЕТ О ПРАКТИЧЕСКОЙ РАБОТЕ №3

ИНТЕРПОЛЯЦИЯ

по курсу: ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 4136                                                                               Бобрович Н. С.

                                                                        подпись, дата                      инициалы, фамилия

Санкт-Петербург 2023

**Цель работы:**

а) освоение методов интерполяции функций;

б) совершенствование навыков по алгоритмизации и программированию вычислительных задач.

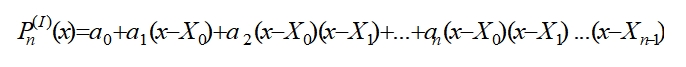
**Задание:**

Скриншот 22-05-2023 205633

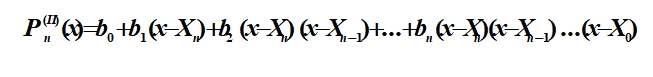
**Математическая часть:**

#### 1.5.2. Интерполяционные формулы Ньютона

Для функций, заданных таблицами с постоянным шагом изменения аргумента, наиболее часто используются первая или вторая формулы Ньютона, в которых интерполяционная функция определяется как многочлен вида:



при интерполяции от нулевого узла ***Х0*** или



при интерполяции от узла ***Хn***.

Значения коэффициентов ***ai*** и ***bi*** в формулах (21) или (22) находятся из условий Лагранжа, определяющих в узлах интерполяции совпадение значений интерполирующей функции со значением табличнозаданной функции

Скриншот 30-05-2023 223707

(см. также формулу (2) в общей постановке задачи интерполяции).

Полагая ***Х=Х0*** , в формуле (21) получим

***Pn(Х0)=a0=Y0 .***

Аналогично для ***Х=Х1***

***Pn(Х1)=a0+a1(X1-X0)=Y1 ,***

и далее

***a1=(Y1-Y0)/(X1-X0)***

или, используя введённые обозначения,

***a1= Δy0/(1!h).***

Продолжая подстановки значений ***Хi ,*** получим

***Pn(Х2)=a0+a1(X2-X0)+ a2 (X2-X0)(X2-X1) =Y2 ,***

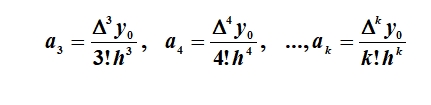
и далее

***a2\*2h2=Y2 - a0 - a1\*2h* = *Y2 - Y0 - Δy0/h\*2h* = *Y2 - 2 Y1 + Y0 = Δ2y0***

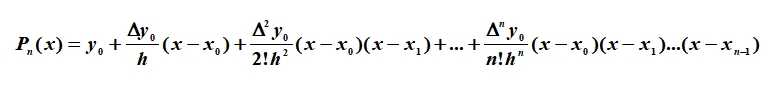
откуда



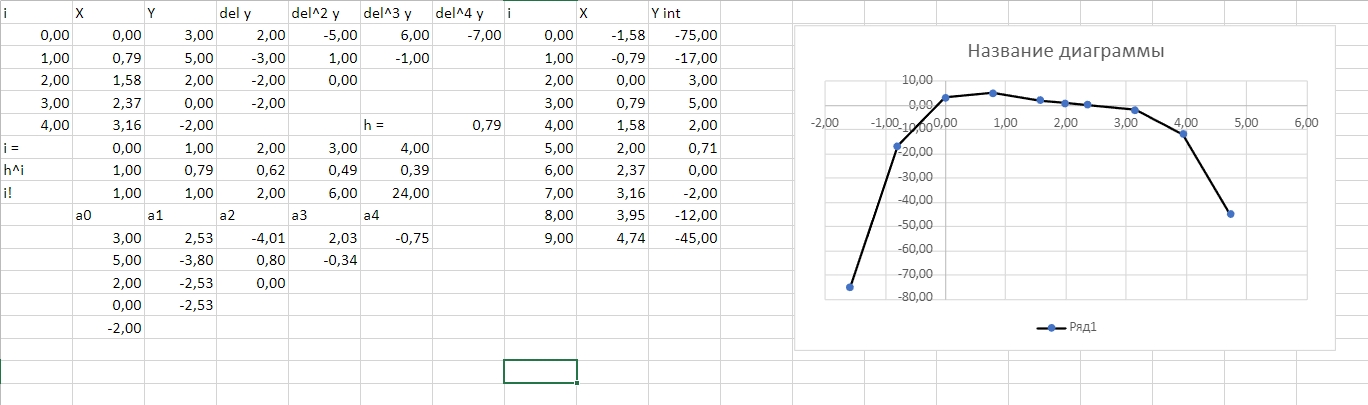
Проведя аналогичные преобразования для ***Х=Х3*** и ***Х=Х4***, получим

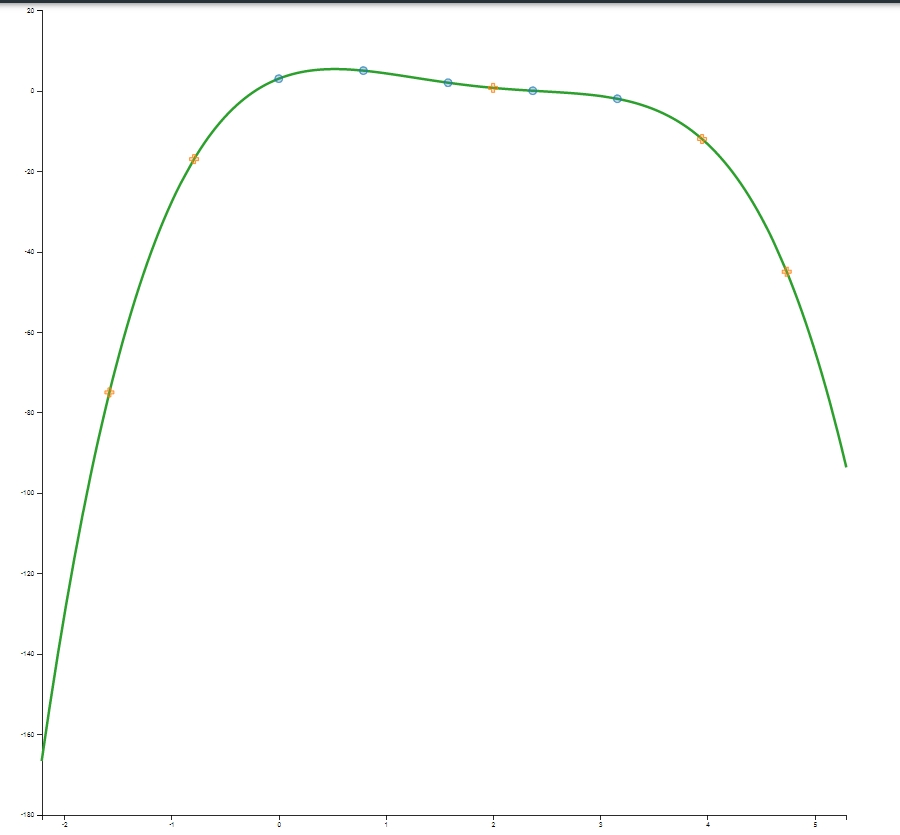
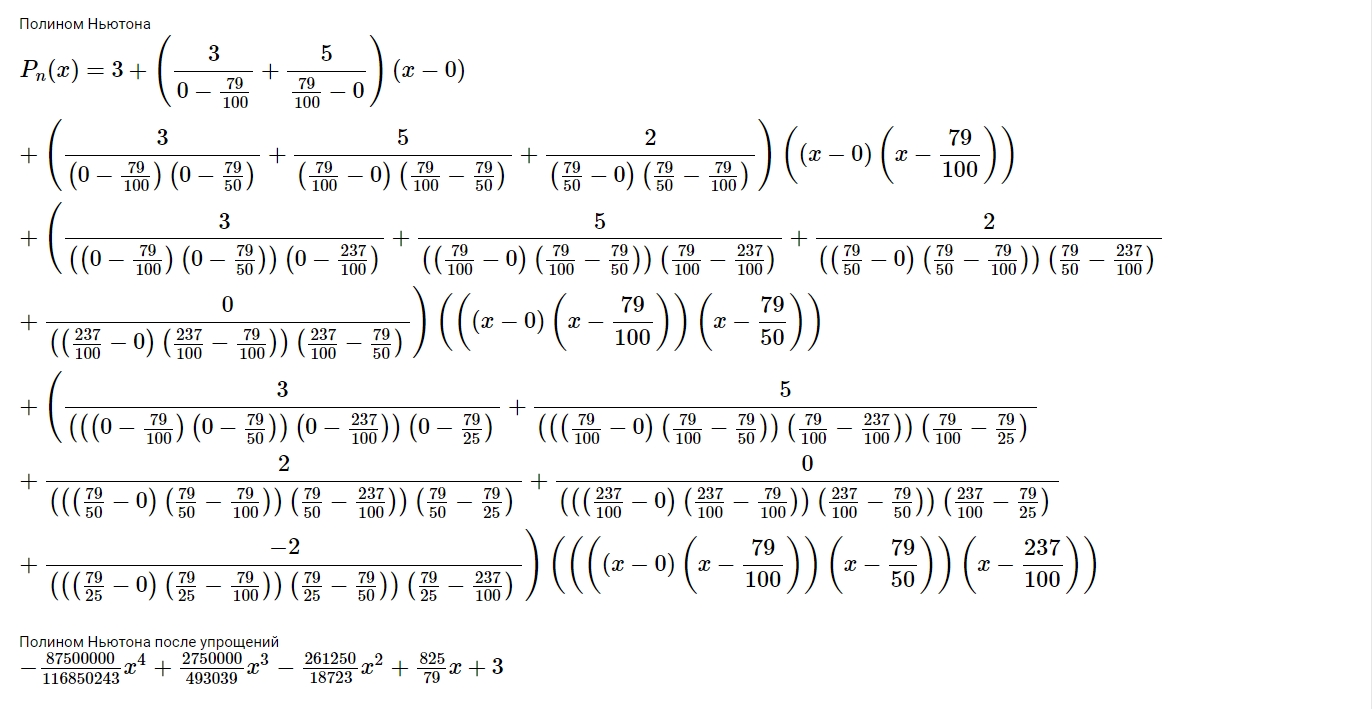
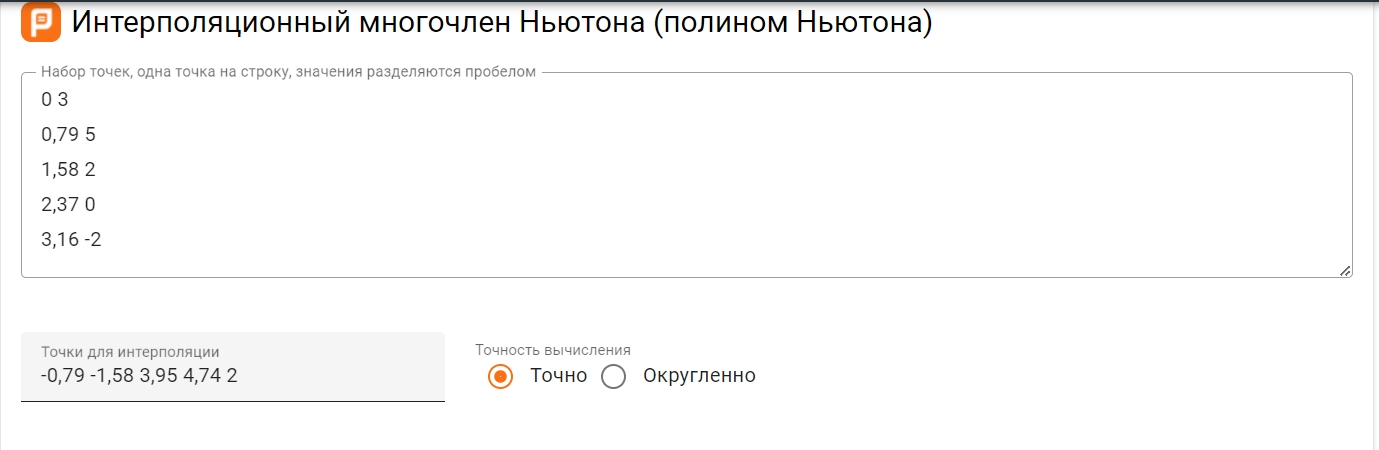


Подставив (24) в (21), получим



**Аналитические расчеты:**

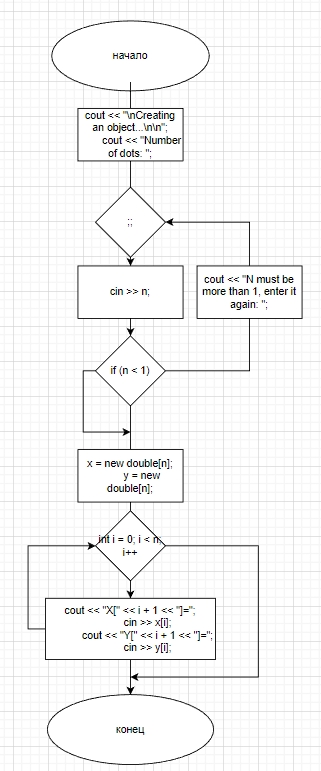
****



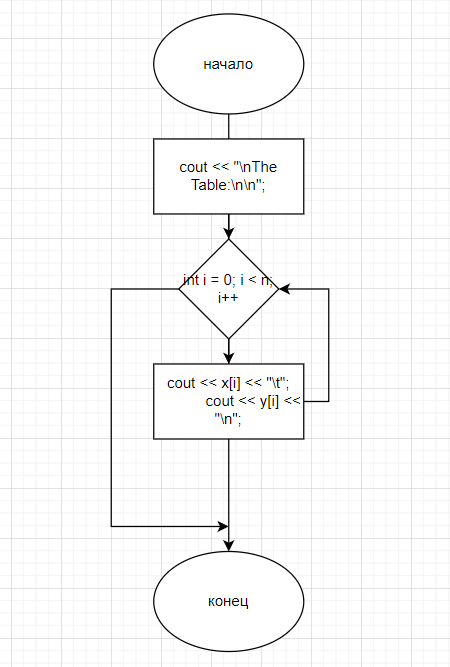
**Схема алгоритма:**

Class newton:

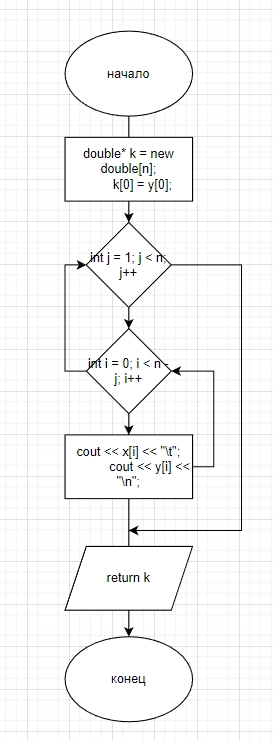
newton:



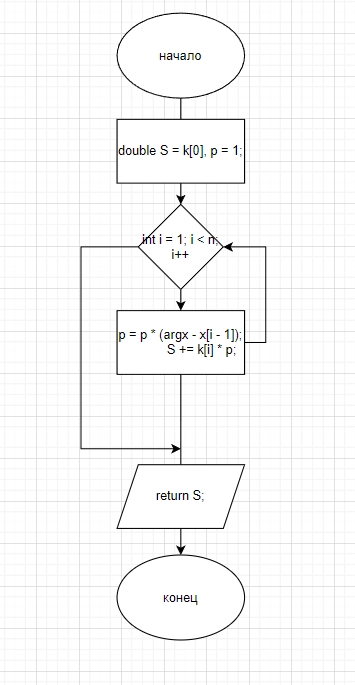
show:



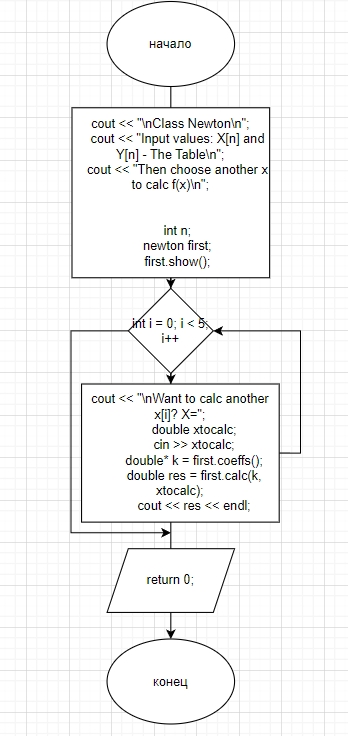
coeffs:



calc:



main:



**Листинг кода программы:**

#include <iostream>

#include <string>

#include <math.h>

#include <stdlib.h>

using namespace std;

class newton

{

double\* x;

double\* y;

int n;

public:

newton()

{

cout << "\nCreating an object...\n\n";

cout << "Number of dots: ";

for (;;)

{

cin >> n;

if (n < 1)

cout << "N must be more than 1, enter it again: ";

else break;

}

cout << endl;

x = new double[n];

y = new double[n];

for (int i = 0; i < n; i++)

{

cout << "X[" << i + 1 << "]=";

cin >> x[i];

cout << "Y[" << i + 1 << "]=";

cin >> y[i];

}

}

void show()

{

cout << "\nThe Table:\n\n";

for (int i = 0; i < n; i++)

{

cout << x[i] << "\t";

cout << y[i] << "\n";

}

}

~newton()

{

delete[] x;

delete[] y;

}

double\* coeffs()

{

double\* k = new double[n];

k[0] = y[0];

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

for (int i = 0; i < n - j; i++)

{

y[i] = (y[i + 1] - y[i]) / (x[i + j] - x[i]);

//cout << y[i] << endl;

k[j] = y[0];

}

cout << k[j] << endl;

}

return k;

}

double calc(double\* k, double argx)

{

double S = k[0], p = 1;

S += k[1] \* (argx - x[1]) + k[2] \* (argx - x[1]) \* (argx - x[2]) + k[3] \* (argx - x[1]) \* (argx - x[2]) \* (argx - x[3]) + k[4] \* (argx - x[1]) \* (argx - x[2]) \* (argx - x[3]) \* (argx - x[4]);

//cout << "S = " << S << endl;

return S;

}

};

int main()

{

cout << "\nClass Newton\n";

cout << "Input values: X[n] and Y[n] - The Table\n";

cout << "Then choose another x to calc f(x)\n";

int n;

double h = 0.79;

newton first;

first.show();

double\* k = first.coeffs();

for (int i = 0; i < 5; i++) {

cout << "\nWant to calc another x[i]? X=";

double xtocalc;

cin >> xtocalc;

double res = first.calc(k, xtocalc + h);

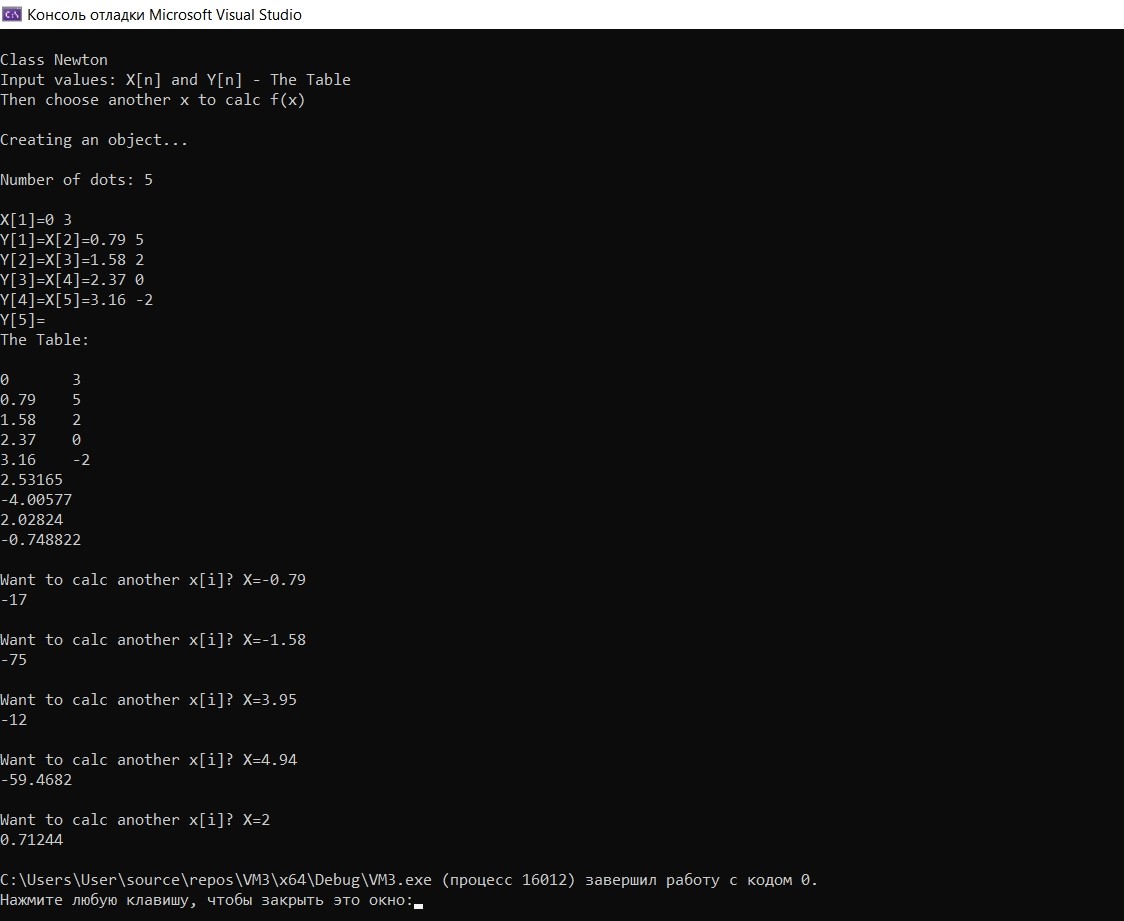
cout << res << endl;

}

return 0;

}

**Результаты программных расчетов:**



**Сравнение результатов программных и аналитических расчетов:**

Исходя из результатов мы видим, что результаты сходятся с допустимой разницей.

**Вывод**

В ходе выполнения практической работы №3 была освоена интерполяция методом Ньютона. Также были улучшены навыки по программированию задачи на языке C++ в программе Microsoft Visual Studio.