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:

3	0	0.79	1.58	2.37	3.16	3.0	5.0	2.0	0.0	-2.0	Ньютона

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:

$$P_n^{(I)}(x)=a_0+a_1(x-X_0)+a_2(x-X_0)(x-X_1)+\ldots+a_n(x-X_0)(x-X_1)\ldots(x-X_{n-1})$$

при интерполяции от нулевого узла X_0 или

$$P_n^{(II)}(x)=b_0+b_1(x-X_n)+b_2(x-X_n)(x-X_{n-1})+\ldots+b_n(x-X_n)(x-X_{n-1})\ldots(x-X_0)$$

при интерполяции от узла X_n.

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

$$P_n(x_i)=Y_i$$

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

Полагая $X=X_0$, в формуле (21) получим

$$P_n(X_0)=a_0=Y_0 .$$

Аналогично для $X=X_1$

$$P_n(X_1)=a_0+a_1(X_1-X_0)=Y_1 ,$$

и далее

$$a_1=(Y_1-Y_0)/(X_1-X_0)$$

или, используя введённые обозначения,

$$a_1 = \Delta y_0/(1!h).$$

Продолжая подстановки значений X_i, получим

$$P_n(X_2)=a_0+a_1(X_2-X_0)+a_2(X_2-X_0)(X_2-X_1)=Y_2,$$

и далее

$$a_2*2h^2=Y_2-a_0-a_1*2h=Y_2-Y_0-\Delta y_0/h*2h=Y_2-2Y_1+Y_0=\Delta^2 y_0$$

откуда

$$a_2=\frac{\Delta^2 y_0}{2h^2}.$$

Проведя аналогичные преобразования для $X=X_3$ и $X=X_4$, получим

$$a_3=\frac{\Delta^3 y_0}{3!h^3},\quad a_4=\frac{\Delta^4 y_0}{4!h^4},\quad \ldots, a_k=\frac{\Delta^k y_0}{k!h^k}$$

Подставив (24) в (21), получим

$$P_n(x)=y_0+\frac{\Delta y_0}{h}(x-x_0)+\frac{\Delta^2 y_0}{2!h^2}(x-x_0)(x-x_1)+\ldots+\frac{\Delta^n y_0}{n!h^n}(x-x_0)(x-x_1)\ldots(x-x_{n-1})$$

Аналитические расчеты:

i	X	Y	del y	del^2 y	del^3 y	del^4 y	i	X	Y int
0,00	0,00	3,00	2,00	-5,00	6,00	-7,00	0,00	-1,58	-75,00
1,00	0,79	5,00	-3,00	1,00	-1,00		1,00	-0,79	-17,00
2,00	1,58	2,00	-2,00	0,00			2,00	0,00	3,00
3,00	2,37	0,00	-2,00				3,00	0,79	5,00
4,00	3,16	-2,00			h =	0,79	4,00	1,58	2,00
i =	0,00	1,00	2,00	3,00	4,00		5,00	2,00	0,71
h^i	1,00	0,79	0,62	0,49	0,39		6,00	2,37	0,00
i!	1,00	1,00	2,00	6,00	24,00		7,00	3,16	-2,00
	a0	a1	a2	a3	a4		8,00	3,95	-12,00
	3,00	2,53	-4,01	2,03	-0,75		9,00	4,74	-45,00
	5,00	-3,80	0,80	-0,34					
	2,00	-2,53	0,00						
	0,00	-2,53							
	-2,00								

Интерполяционный многочлен Ньютона (полином Ньютона)

Набор точек, одна точка на строку, значения разделяются пробелом

0 3
0,79 5
1,58 2
2,37 0
3,16 -2

Точки для интерполяции

-0,79 -1,58 3,95 4,74 2

Точность вычисления

Точно Округленно

Полином Ньютона

$$
\begin{aligned}
P_n(x) = 3 &+ \left(\frac{3}{0-\frac{79}{100}} + \frac{5}{\frac{79}{100}-0}\right)(x-0) \\
&+ \left(\frac{3}{\left(0-\frac{79}{100}\right)\left(0-\frac{79}{50}\right)} + \frac{5}{\left(\frac{79}{100}-0\right)\left(\frac{79}{100}-\frac{79}{50}\right)} + \frac{2}{\left(\frac{79}{50}-0\right)\left(\frac{79}{50}-\frac{79}{100}\right)}\right)\left((x-0)\left(x-\frac{79}{100}\right)\right) \\
&+ \left(\frac{3}{\left(\left(0-\frac{79}{100}\right)\left(0-\frac{79}{50}\right)\right)\left(0-\frac{237}{100}\right)} + \frac{5}{\left(\left(\frac{79}{100}-0\right)\left(\frac{79}{100}-\frac{79}{50}\right)\right)\left(\frac{79}{100}-\frac{237}{100}\right)} + \frac{2}{\left(\left(\frac{79}{50}-0\right)\left(\frac{79}{50}-\frac{79}{100}\right)\right)\left(\frac{79}{50}-\frac{237}{100}\right)}\right. \\
&+ \left.\frac{0}{\left(\left(\frac{237}{100}-0\right)\left(\frac{237}{100}-\frac{79}{100}\right)\right)\left(\frac{237}{100}-\frac{79}{50}\right)}\right)\left(\left((x-0)\left(x-\frac{79}{100}\right)\right)\left(x-\frac{79}{50}\right)\right) \\
&+ \left(\frac{3}{\left(\left(\left(0-\frac{79}{100}\right)\left(0-\frac{79}{50}\right)\right)\left(0-\frac{237}{100}\right)\right)\left(0-\frac{79}{25}\right)} + \frac{5}{\left(\left(\left(\frac{79}{100}-0\right)\left(\frac{79}{100}-\frac{79}{50}\right)\right)\left(\frac{79}{100}-\frac{237}{100}\right)\right)\left(\frac{79}{100}-\frac{79}{25}\right)}\right. \\
&+ \frac{2}{\left(\left(\left(\frac{79}{50}-0\right)\left(\frac{79}{50}-\frac{79}{100}\right)\right)\left(\frac{79}{50}-\frac{237}{100}\right)\right)\left(\frac{79}{50}-\frac{79}{25}\right)} + \frac{0}{\left(\left(\left(\frac{237}{100}-0\right)\left(\frac{237}{100}-\frac{79}{100}\right)\right)\left(\frac{237}{100}-\frac{79}{50}\right)\right)\left(\frac{237}{100}-\frac{79}{25}\right)} \\
&+ \left.\frac{-2}{\left(\left(\left(\frac{79}{25}-0\right)\left(\frac{79}{25}-\frac{79}{100}\right)\right)\left(\frac{79}{25}-\frac{79}{50}\right)\right)\left(\frac{79}{25}-\frac{237}{100}\right)}\right)\left(\left(\left((x-0)\left(x-\frac{79}{100}\right)\right)\left(x-\frac{79}{50}\right)\right)\left(x-\frac{237}{100}\right)\right)
\end{aligned}
$$

Полином Ньютона после упрощений

$$-\frac{87500000}{116850243}x^4 + \frac{2750000}{493039}x^3 - \frac{261250}{18723}x^2 + \frac{825}{79}x + 3$$

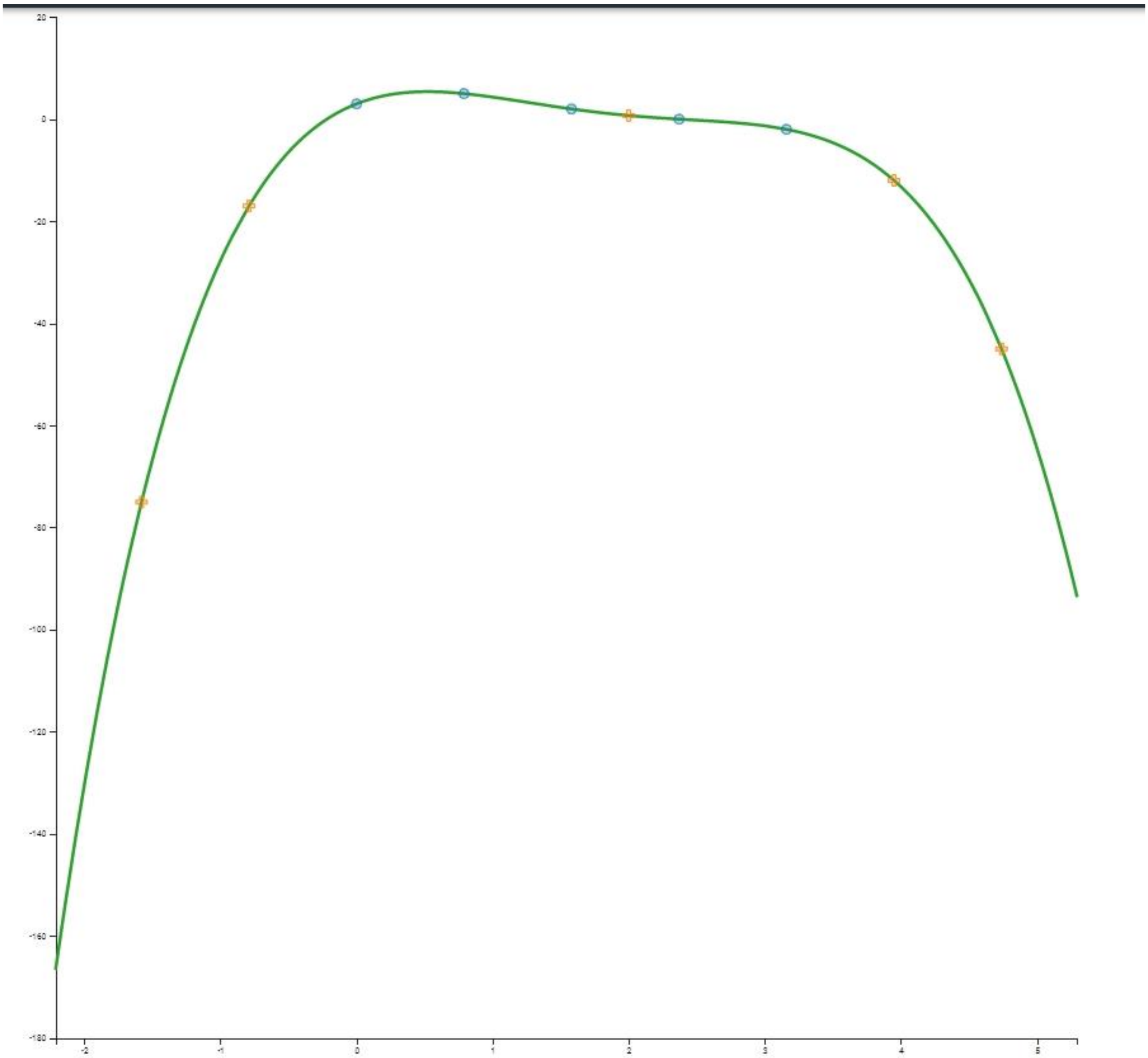
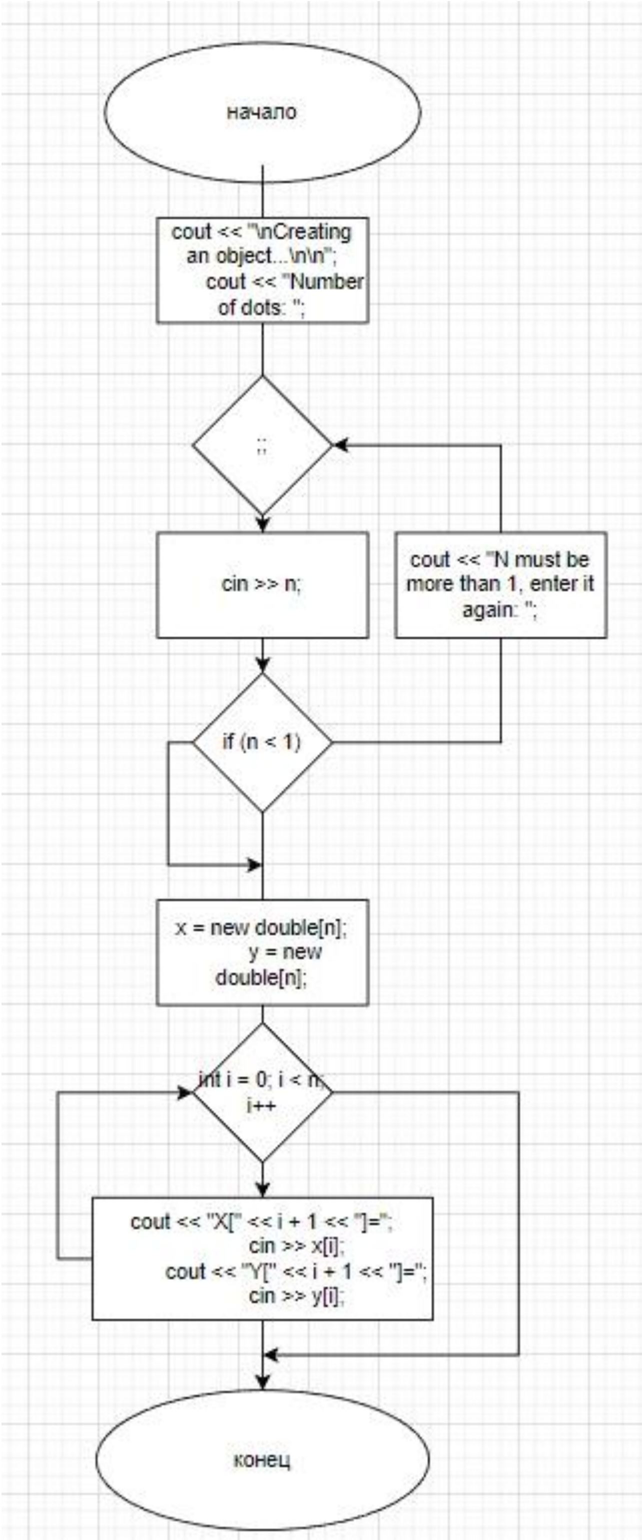


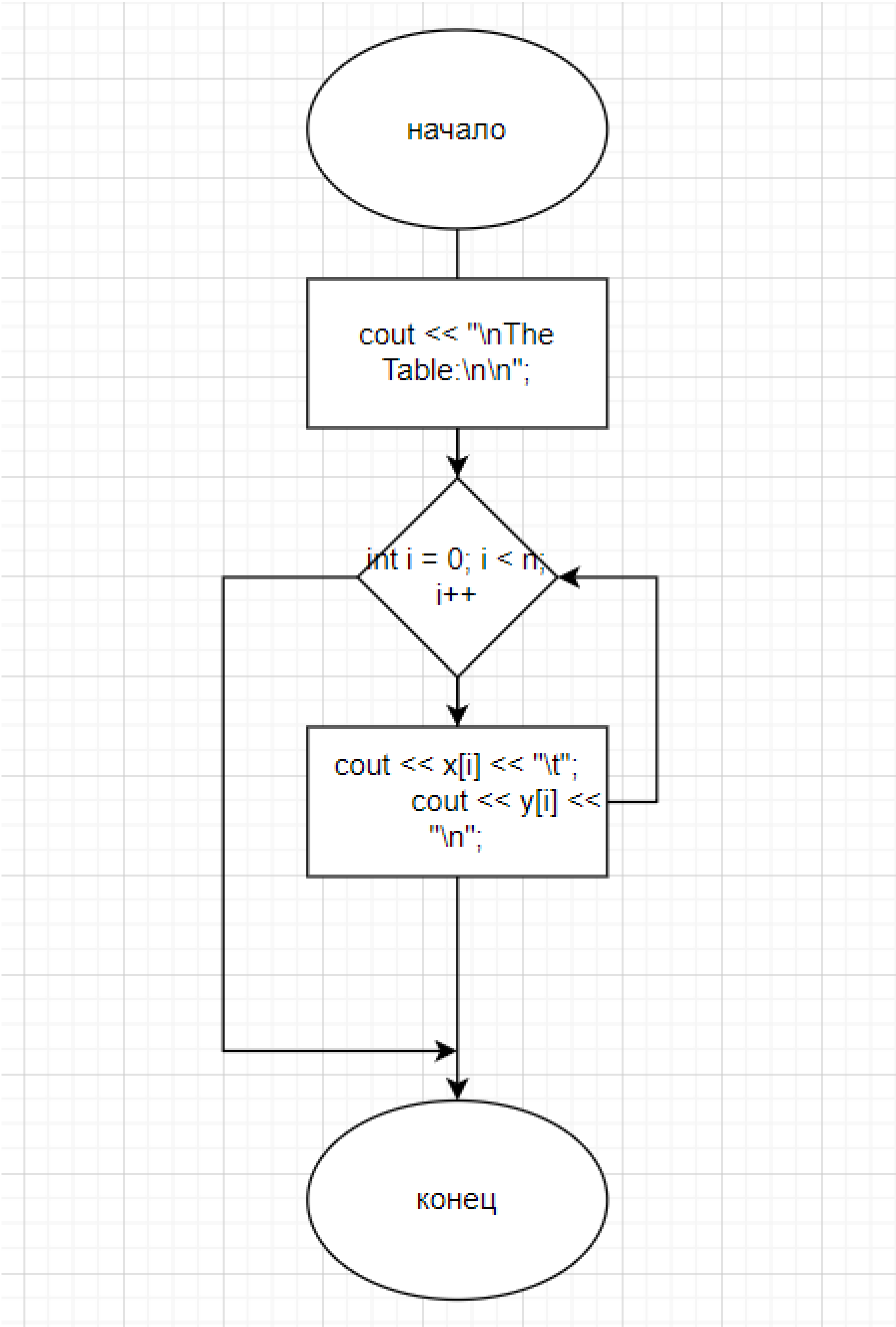
Схема алгоритма:

Class newton:

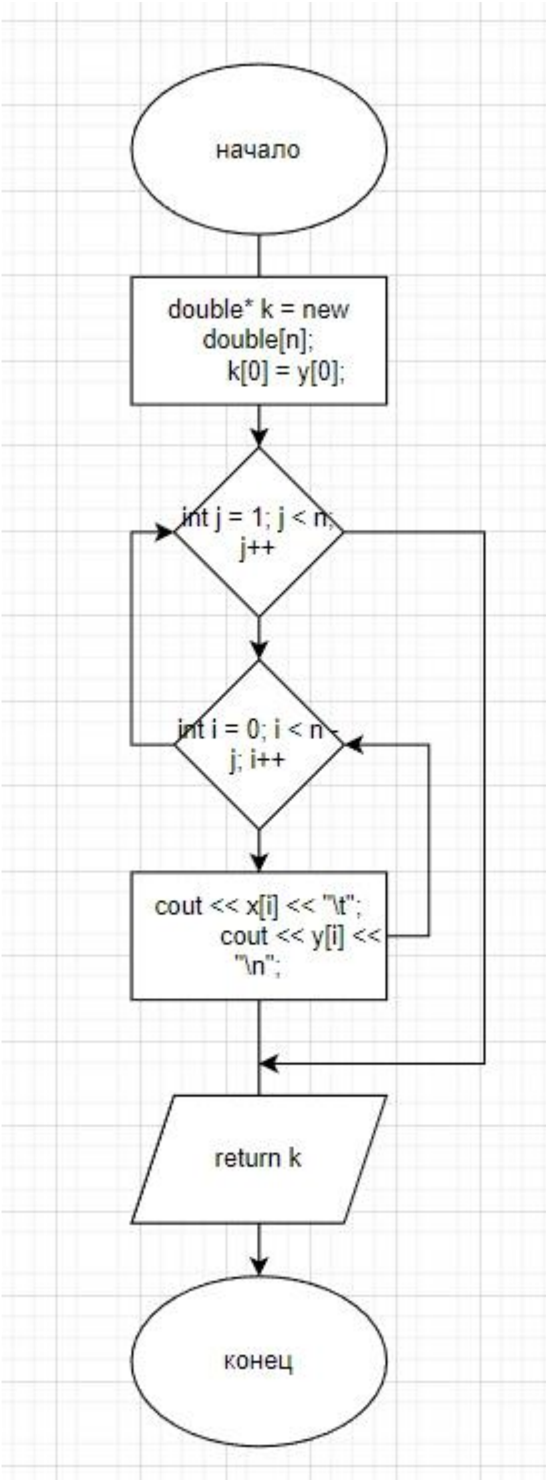
newton:


начало
cout << "\nCreating an object...\n\n"; cout << "Number of dots: ";
;;
cin >> n;
cout << "N must be more than 1, enter it again: ";
if (n < 1)
x = new double[n]; y = new double[n];
int i = 0; i < n; i++
cout << "X[" << i + 1 << "]="; cin >> x[i]; cout << "Y[" << i + 1 << "]="; cin >> y[i];
конец

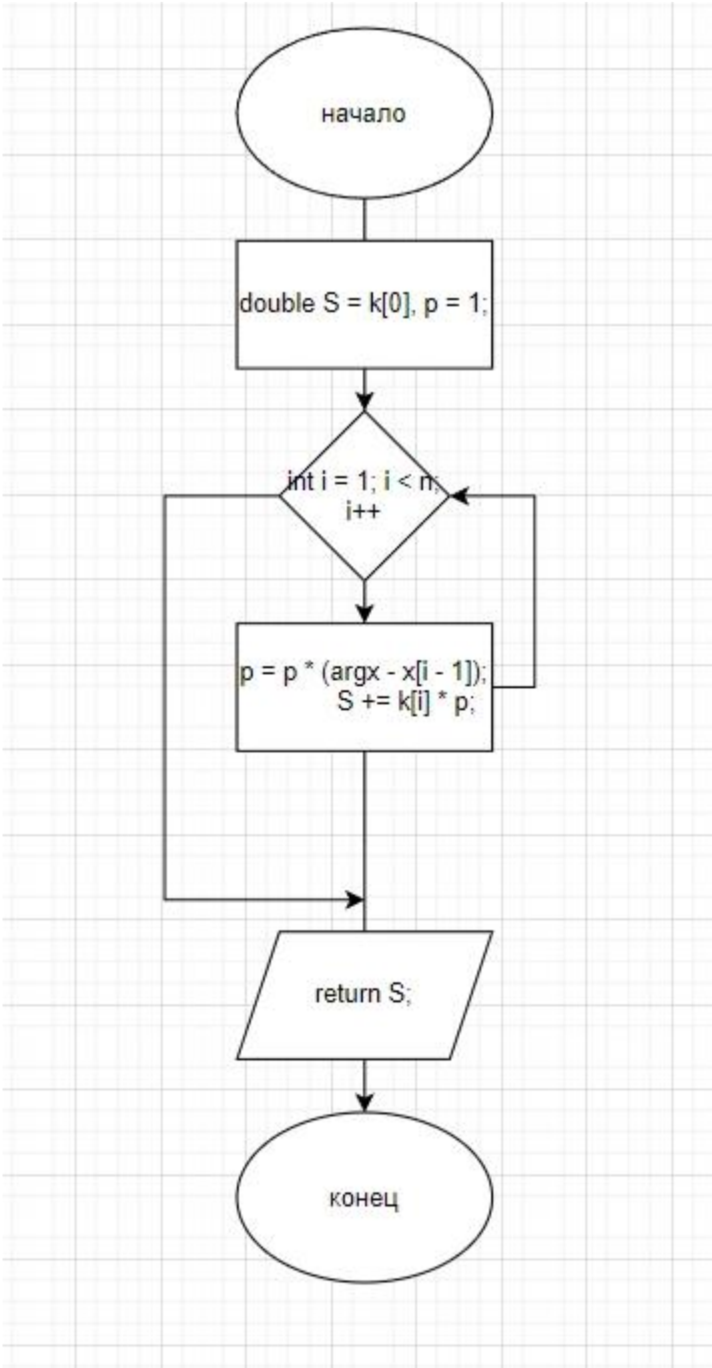
show:



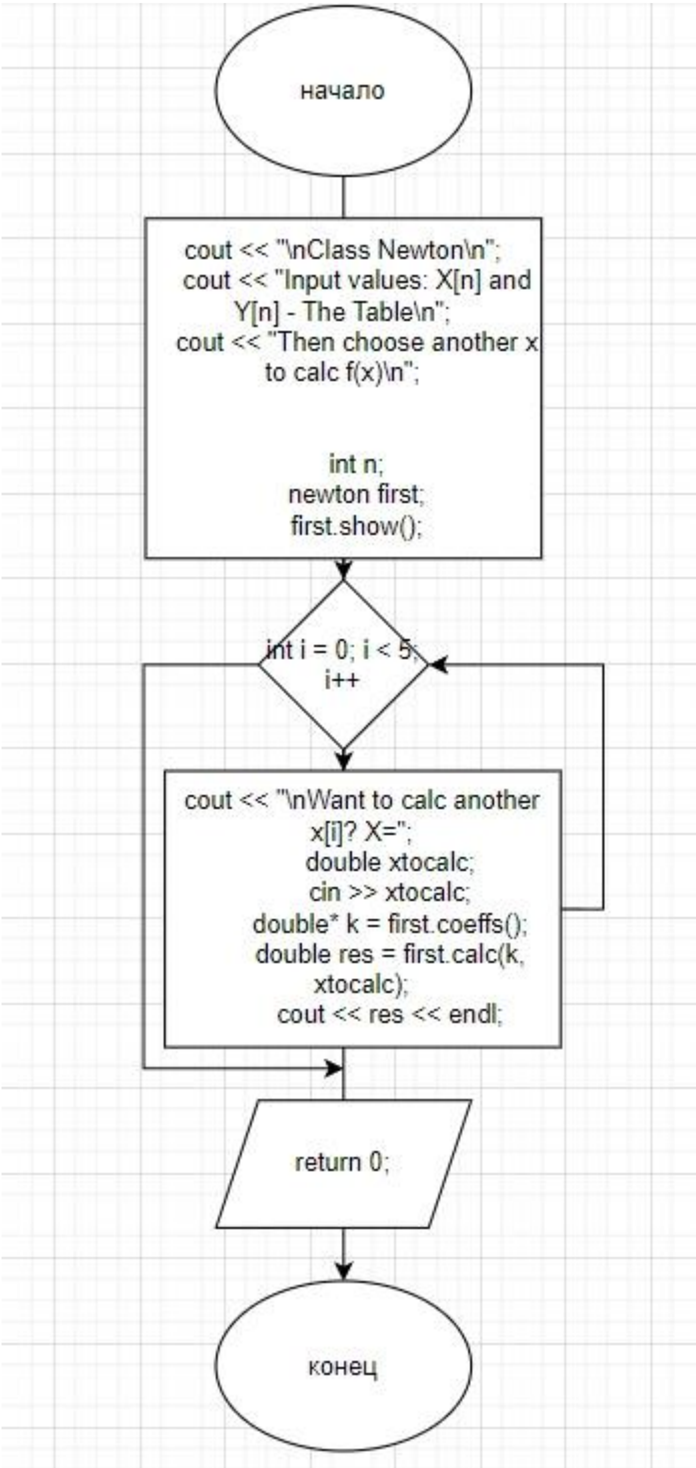
coeffs:


начало
double* k = new
double[n];
k[0] = y[0];
int j = 1; j < n;
j++
int i = 0; i < n -
j; i++
cout << x[i] << "\t";
cout << y[i] <<
"\n";
return k
конец

calc:


начало
double S = k[0], p = 1;
int i = 1; i < n;
i++
p = p * (argx - x[i - 1]);
S += k[i] * p;
return S;
конец

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:

Консоль отладки Microsoft Visual Studio

```
Class Newton
Input values: X[n] and Y[n] - The Table
Then choose another x to calc f(x)

Creating an object...

Number of dots: 5

X[1]=0 3
Y[1]=X[2]=0.79 5
Y[2]=X[3]=1.58 2
Y[3]=X[4]=2.37 0
Y[4]=X[5]=3.16 -2
Y[5]=
The Table:

0       3
0.79    5
1.58    2
2.37    0
3.16    -2
2.53165
-4.00577
2.02824
-0.748822

Want to calc another x[i]? X=-0.79
-17

Want to calc another x[i]? X=-1.58
-75

Want to calc another x[i]? X=3.95
-12

Want to calc another x[i]? X=4.94
-59.4682

Want to calc another x[i]? X=2
0.71244

C:\Users\User\source\repos\VM3\x64\Debug\VM3.exe (процесс 16012) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно:
```

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.