Задание 3-4

class Program

{

static double Lagrange(double[] x, double[] y, double xValue)

{

double result = 0;

int n = x.Length;

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

{

double term = y[i];

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

{

if (i != j)

{

term \*= (xValue - x[j]) / (x[i] - x[j]);

}

}

result += term;

}

return result;

}

static void Main(string[] args)

{

// Заданные точки (x, y)

double[] x = { 0.27, 0.93, 1.46, 2.11, 2.87 };

double[] y = { 2.60, 2.43, 2.06, 0.25, -2.60 };

double[] xValues = { 1.02, 0.65, 1.28 };

foreach (double xValue in xValues)

{

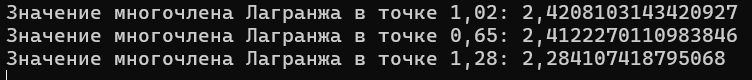
double interpolatedValue = Lagrange(x, y, xValue);

Console.WriteLine($"Значение многочлена Лагранжа в точке {xValue}: {interpolatedValue}");

}

Console.ReadLine();

}



}

Задание 6

using System;

class Program

{

static void Main()

{

double[] x = { 1.05, 1.06, 1.07, 1.08, 1.09, 1.10, 1.11 };

double[] y = { 0.04879, 0.058269, 0.067659, 0.076961, 0.086178, 0.09531, 0.10436 };

double valueToInterpolate = 1.083;

double result = NewtonInterpolation(x, y, valueToInterpolate);

Console.WriteLine($"Интерполированное значение в x = {valueToInterpolate}: y = {result}");

Console.ReadLine();

}

static double NewtonInterpolation(double[] x, double[] y, double value)

{

int n = x.Length;

double[,] dividedDifferences = new double[n, n];

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

{

dividedDifferences[i, 0] = y[i];

}

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

{

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

{

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

}

}

double interpolatedValue = dividedDifferences[0, 0];

double productTerm = 1.0;

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

{

productTerm \*= (value - x[j - 1]);

interpolatedValue += dividedDifferences[0, j] \* productTerm;

}

return interpolatedValue;

}

}

