**2.Классы**

Задание 1. Создать класс А с целочисленными полями а и b и двумя методами согласно варианту. Внутри класса реализовать конструктор для инициализации a и b. Создать объект класса и продемонстрировать работу со всеми элементами класса.

namespace task1

{

class ClassA

{

private int \_a;

private int \_b;

public int a

{

get { return \_a; }

set { \_a = value; }

}

public int b

{

get { return \_b; }

set { \_b = value; }

}

public ClassA(int a, int b)

{

\_a = a;

\_b = b;

}

public double Сalculation(int a, int b)

{

double rezult = (Math.Sin(b) + 4 )/(2\*a);

return rezult;

}

public double SquaringSum(int a, int b)

{

double rezult = Math.Pow(a + b, 2);

return rezult;

}

public void Print(int a,int b)

{

Console.WriteLine("Вычесление равно: " + Сalculation(a, b));

Console.WriteLine("Сумма квадратов равна:" + SquaringSum(a,b));

}

}

}

namespace task1

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите A: ");

int a = int.Parse(Console.ReadLine());

Console.Write("Введите B: ");

int b = int.Parse(Console.ReadLine());

ClassA A = new ClassA(a, b);

A.Print(a, b);

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4, 5 | 0,3801344656671077 и 81 |

Анализ результатов:

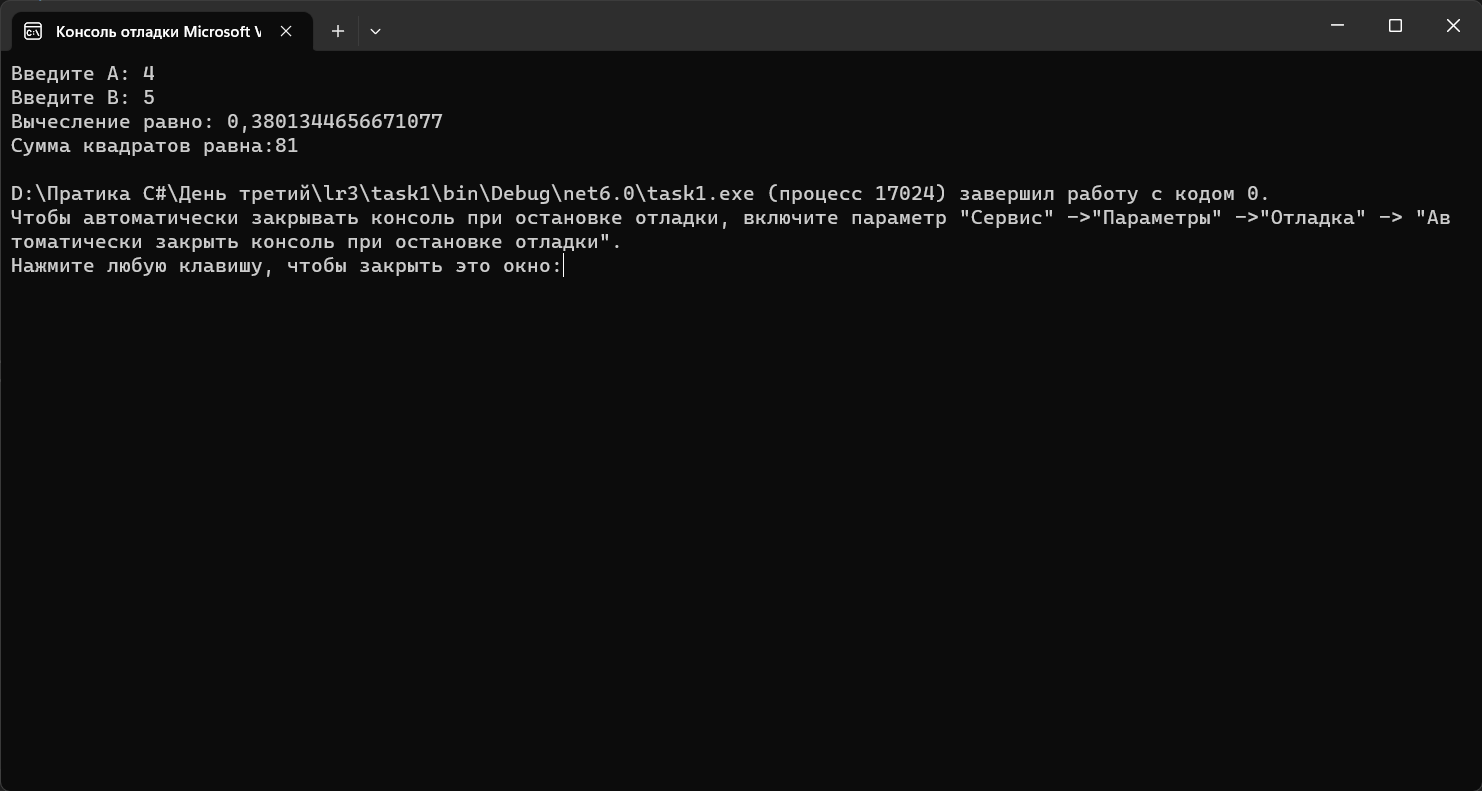


Рисунок 1.1 – Результат работы программы

Задание 2. Составить описание класса многочлена вида ах 2 + bх + с. Предусмотреть

методы, реализующие:

вычисление значения многочлена для заданного аргумента;

операцию сложения, вычитания и умножения многочленов с получением

нового объекта-многочлена;

вывод на экран описания многочлена.

Написать программу, демонстрирующую все разработанные элементы класса.

namespace task\_3

{

class Polynomial

{

private double \_a, \_b, \_c;

public double A

{

get { return \_a; }

set { \_a = value; }

}

public double B

{

get { return \_b; }

set { \_b = value; }

}

public double C

{

get { return \_c; }

set { \_c = value; }

}

public Polynomial(double a, double b, double c)

{

\_a = a;

\_b = b;

\_c = c;

}

public double Evaluate(double x)

{

// Вычисление значения многочлена для заданного аргумента

return A \* x \* x + B \* x + C;

}

public static Polynomial operator +(Polynomial p1, Polynomial p2)

{

// Сложение многочленов

return new Polynomial(p1.A + p2.A, p1.B + p2.B, p1.C + p2.C);

}

public static Polynomial operator -(Polynomial p1, Polynomial p2)

{

// Вычитание многочленов

return new Polynomial(p1.A - p2.A, p1.B - p2.B, p1.C - p2.C);

}

public static Polynomial operator \*(Polynomial p1, Polynomial p2)

{

// Умножение многочленов

double a = p1.A \* p2.A;

double b = p1.A \* p2.B + p1.B \* p2.A;

double c = p1.A \* p2.C + p1.B \* p2.B + p1.C \* p2.A;

return new Polynomial(a, b, c);

}

public void Print()

{

Console.WriteLine("{0}x^2 + {1}x + {2}", A, B, C);

}

}

}

namespace task\_3

{

internal class Program

{

static void Main()

{

Polynomial p1 = new Polynomial(2, 3, 1);

Polynomial p2 = new Polynomial(-1, 5, 2);

// Вычисление значения многочлена для заданного аргумента

Console.WriteLine("Значение многочлена p1 при x=2: {0}", p1.Evaluate(2));

// Сложение многочленов

Polynomial p3 = p1 + p2;

Console.Write("p1 + p2 = ");

p3.Print();

// Вычитание многочленов

Polynomial p4 = p1 - p2;

Console.Write("p1 - p2 = ");

p4.Print();

// Умножение многочленов

Polynomial p5 = p1 \* p2;

Console.Write("p1 \* p2 = ");

p5.Print();

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

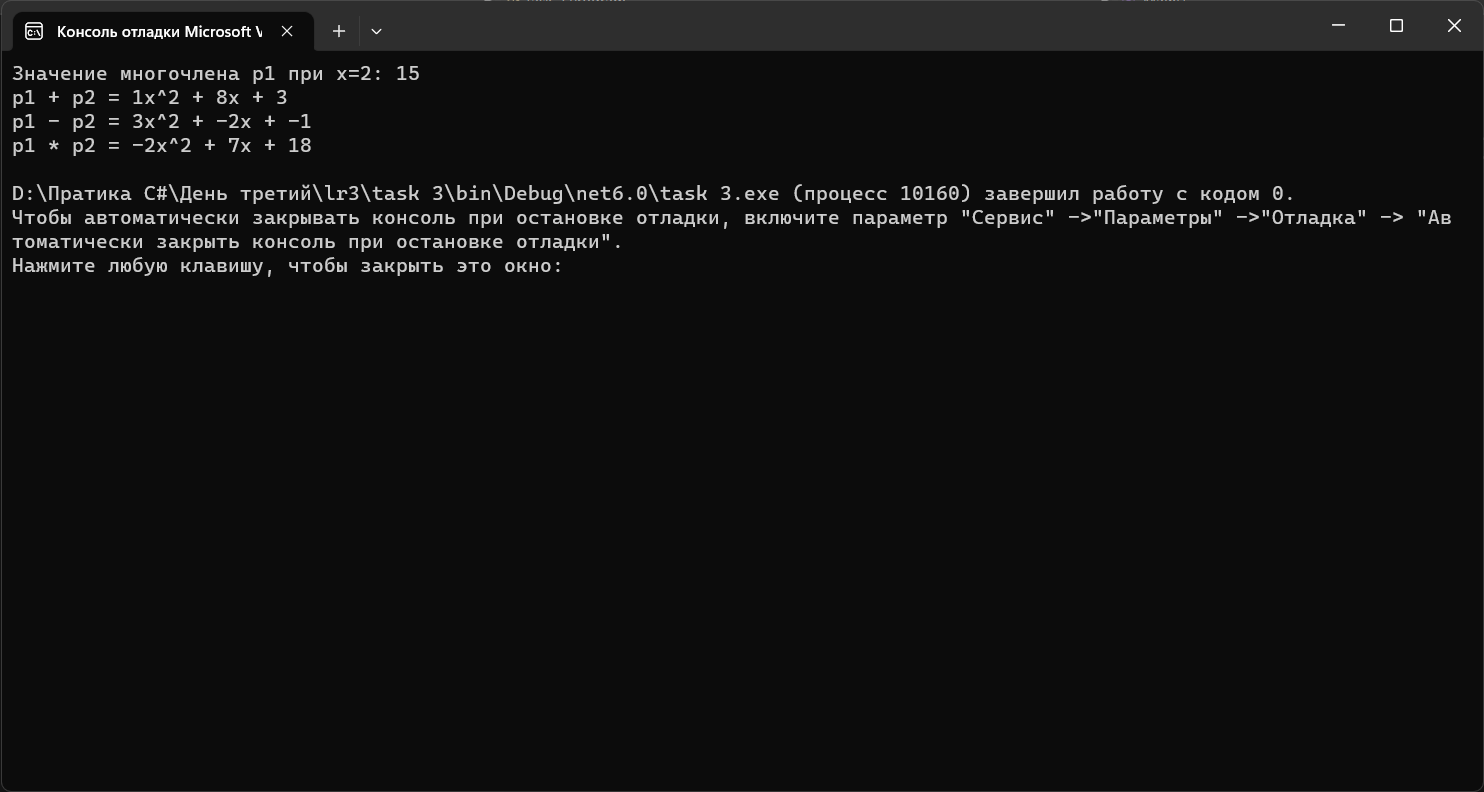


Рисунок 1.2 – Результат работы программы

работы программы