**14.** Сборки, Библиотеки, Атрибуты, Директивы

Задание 1. Создать библиотеку с двумя классами: Треугольник (методы ввода сторон, проверки на существование, вычисления периметра, вычисления площади, определения вида треугольника (разносторонний, равнобедренный, равносторонний)); Прямоугольник (методы ввода сторон, вычисления периметра, вычисления площади).

using GeometryLibrary;

namespace task1

{

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter triangle sides: ");

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

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

double c = double.Parse(Console.ReadLine());

try

{

Triangle t = new Triangle(a, b, c);

Console.WriteLine($"Perimeter: {t.Perimeter()}");

Console.WriteLine($"Area: {t.Area()}");

Console.WriteLine($"Type: {t.Type()}");

}

catch (ArgumentException e)

{

Console.WriteLine(e.Message);

}

Console.WriteLine("\nEnter rectangle sides: ");

double l = double.Parse(Console.ReadLine());

double w = double.Parse(Console.ReadLine());

try

{

Rectangle r = new Rectangle(l, w);

Console.WriteLine($"Perimeter: {r.Perimeter()}");

Console.WriteLine($"Area: {r.Area()}");

}

catch (ArgumentException e)

{

Console.WriteLine(e.Message);

}

Console.ReadLine();

}

}

}

Анализ результатов:

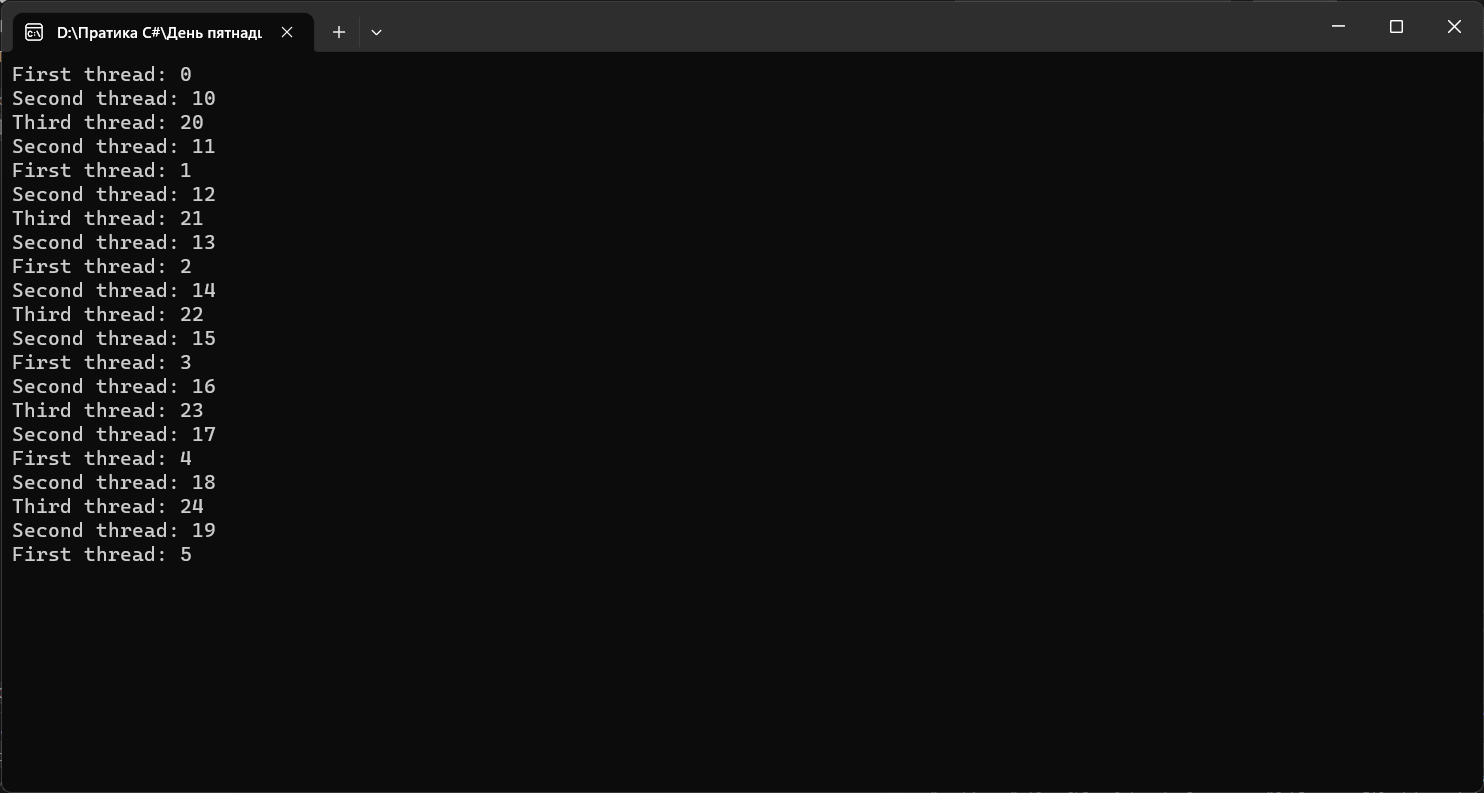


Рисунок 1.1 –работы программы

Задание 2. Создать консольное приложение, в котором будут два метода, один метод будет выполняться двумя потоками одновременно, а другой метод в каждый момент времени будет выполняться одним потоком. (Количество потоков: 2, Метод: 1. A+A 1 +A 2 +A 3 +..+A N , А и N вводятся с клавиатуры; 2. A\*A 1 \*A 2 \*A 3 \*…\*A N , А и N вводятся с клавиатуры ).

namespace task3

{

using System;

using System.Threading;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("Enter A:");

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

Console.WriteLine("Enter N:");

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

Thread t1 = new Thread(() => Method1(A, N));

Thread t2 = new Thread(() => Method1(A, N));

Thread t3 = new Thread(() => Method2(A, N));

t1.Start();

t2.Start();

t3.Start();

t1.Join();

t2.Join();

t3.Join();

}

static void Method1(int A, int N)

{

int sum = A;

for (int i = 1; i <= N; i++)

{

sum += A + i;

}

Console.WriteLine("Method1: Sum = " + sum + " (Thread ID: " + Thread.CurrentThread.ManagedThreadId + ")");

}

static void Method2(int A, int N)

{

int prod = A;

for (int i = 1; i <= N; i++)

{

prod \*= A + i;

}

Console.WriteLine("Method2: Product = " + prod + " (Thread ID: " + Thread.CurrentThread.ManagedThreadId + ")");

}

}

}Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 5, 8 | 81 |

Анализ результатов:

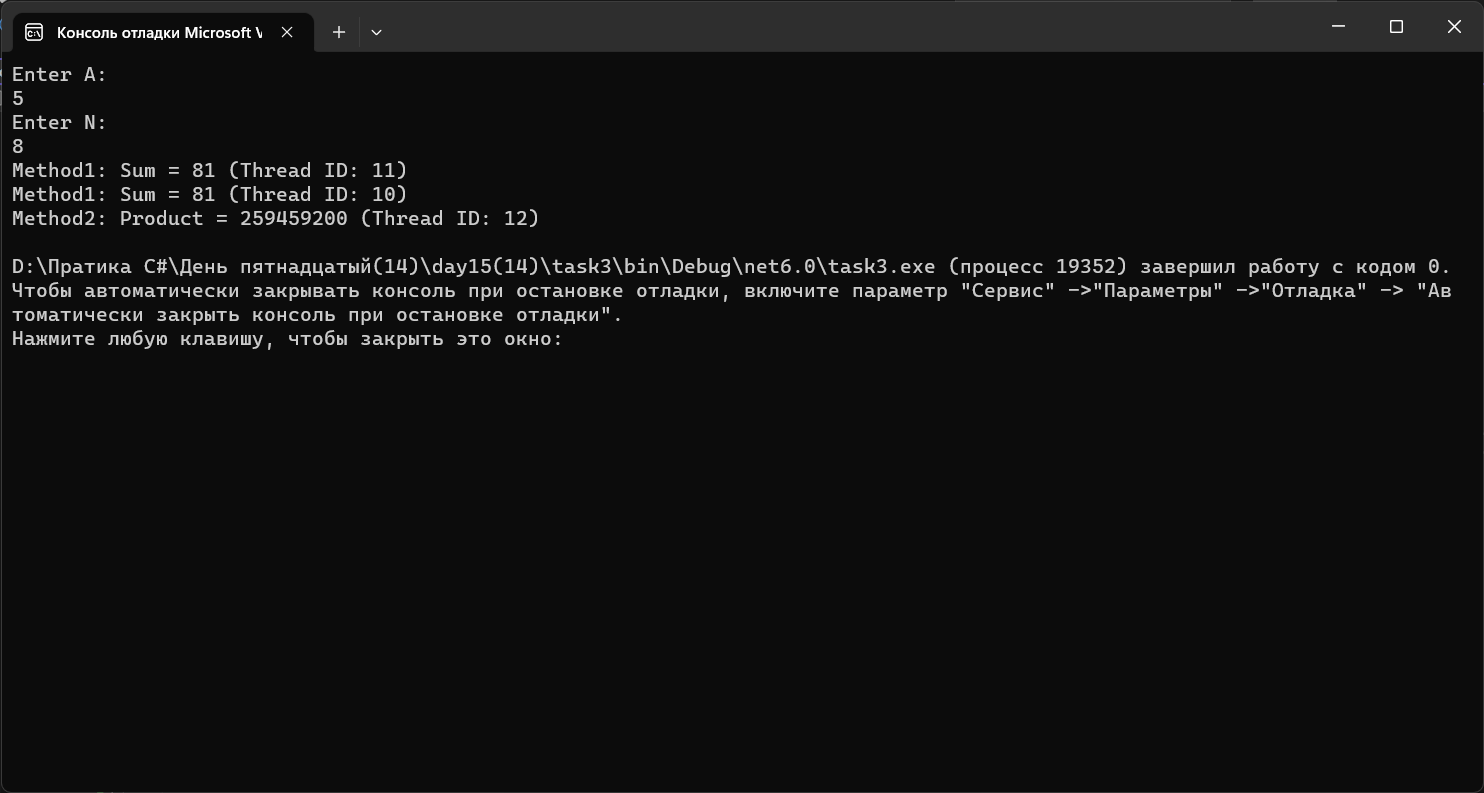
**

Рисунок 1.2 – Результат работы программы

работы программы