**Принципы объектно-ориентированного программирования и платформа Microsoft .Net Framework**

**Задание 1**

**Дано расстояние в сантиметрах. Найти число полных**

**метров в нем.**

**Код**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task1

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите дистанцию в сантиметрах :");

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

int fullMeters = ConvertToFullMeters(distanceCm);

Console.WriteLine($"Полные метры: {fullMeters}");

}

static int ConvertToFullMeters(int centimeters)

{

return centimeters / 100;

}

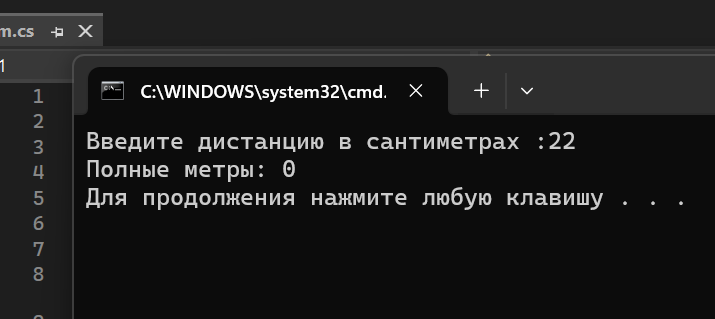
}

}

**Таблица 3.1- Входные и выходные данные**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| 22 | 0 |

**Анализ результата**

****

**Задание 2**

**Проверить истинность высказывания: &quot;Квадратное**

**уравнение A·x2 + B·x + C = 0 с данными коэффициентами A, B, C имеет**

**вещественные корни&quot;.**

**Код**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Task2

{

internal class Program

{

static void Main(string[] args)

{

Console.Write("Введите коэффициент A: ");

double A = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите коэффициент B: ");

double B = Convert.ToDouble(Console.ReadLine());

Console.Write("Введите коэффициент C: ");

double C = Convert.ToDouble(Console.ReadLine());

bool hasRealRoots = CheckForRealRoots(A, B, C);

Console.WriteLine($"Квадратное уравнение имеет вещественные корни: {hasRealRoots}");

}

static bool CheckForRealRoots(double A, double B, double C)

{

if (A == 0)

{

throw new ArgumentException("Коэффициент A не должен быть равен нулю.");

}

double discriminant = B \* B - 4 \* A \* C;

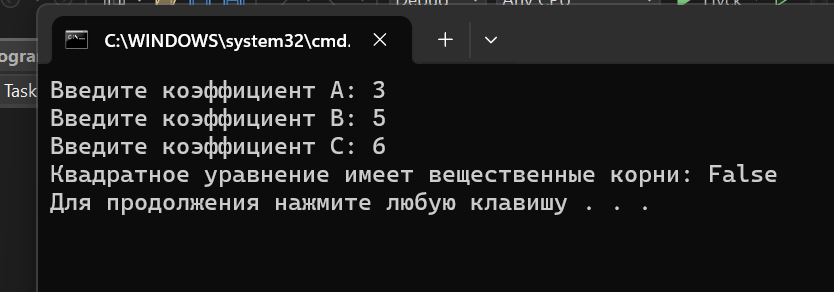
return discriminant >= 0;

}

}

}

**Анализ результата**

****