**3.** **Методы. Реализация полиморфизма**

Задание 1. Напишите статический метод, который преобразует число в двоичное представление.

Листинг задачи:

namespace Task1

{

public static class BinaryConvert

{

public static string ConvertToBinary(int decimalNumber)

{

return System.Convert.ToString(decimalNumber, 2);

}

}

}

using Task1;

public class Program

{

public static void Main(string[] args)

{

Console.Write("Введите целое число: ");

string input = Console.ReadLine();

if (int.TryParse(input, out int decimalNumber))

{

string binaryRepresentation = BinaryConvert.ConvertToBinary(decimalNumber);

Console.WriteLine("Десятичное число: " + decimalNumber + ", Двоичное представление: " + binaryRepresentation);

}

else

{

Console.WriteLine("Некорректный ввод. Пожалуйста, введите целое число.");

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 10 | 1010 |

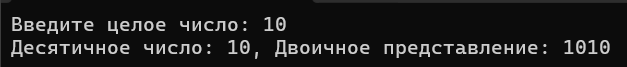
Анализ результатов:

Рисунок 1.1 – Результат работы программы

Задание 2. Описать процедуру DigitCountSum(K, C, S), находящую количество C цифр целого положительного числа K, а также их сумму S (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

Листинг задачи:

namespace Task2

{

public static class DigitCounter

{

public static void CountSum(int k, out int c, out int s)

{

if (k <= 0)

{

throw new System.ArgumentException("Число должно быть целым и положительным.");

}

string kStr = k.ToString();

c = kStr.Length;

s = 0;

foreach (char digitChar in kStr)

{

s += int.Parse(digitChar.ToString());

}

}

}

}

using Task2;

public class Program

{

public static void Main(string[] args)

{

int[] numbers = { 123, -45, 6789, 0, 555 };

foreach (int number in numbers)

{

try

{

DigitCounter.CountSum(number, out int count, out int sum);

Console.WriteLine("Число: " + number + ", Количество цифр: " + count + ", Сумма цифр: " + sum);

}

catch (System.ArgumentException ex)

{

Console.WriteLine("Число: " + number + ", Ошибка: " + ex.Message);

}

}

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123, -45, 6789, 0, 555 | 3,6,Ошибка: Число должно быть целым и положительным4, 30, Ошибка: Число должно быть целым и положительным, 3, 15 |

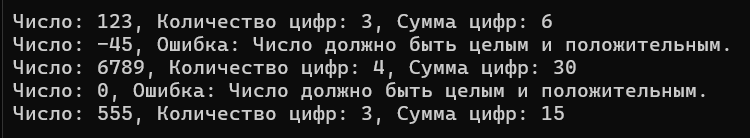
Анализ результатов:

Рисунок 1.2 – Результат работы программы