**20.** ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ

Задание 1. Создайте и реализуйте метод решения задач и выполните его в объектах класса Task используя три варианта создания объектов класса Task: Дано трехзначное число, в котором все цифры различны. Получить шесть чисел, образованных при перестановке цифр заданного числа.

static void Main(string[] args)

{

Console.WriteLine("Введите трех значное число");

int number = Convert.ToInt32(Console.ReadLine());

Task task1 = new Task(() => NumberGenerator(number));

task1.Start();

Task task2 = Task.Factory.StartNew(() => NumberGenerator(number));

Task task3 = Task.Run(() => NumberGenerator(number));

Console.ReadLine();

}

static void NumberGenerator(int number)

{

int a = number / 100;

int b = number / 10 % 10 ;

int c = number % 10;

Console.WriteLine($"Первое число {a}{b}{c}");

Console.WriteLine($"Второе число {a}{c}{b}");

Console.WriteLine($"Третье число {b}{a}{c}");

Console.WriteLine($"Четвертое число {b}{c}{a}");

Console.WriteLine($"пятое число {c}{b}{a}");

Console.WriteLine($"шестое число {c}{a}{b}");

}

Анализ результатов:

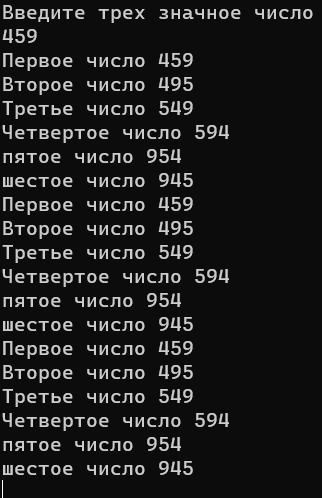


Рисунок 1.1 –работы программы

Задание 2. Создайте два объекта класса Task. Первый объект возвращает результат вычисления, второй объект является задачей продолжения первого объекта и выводит результат первой задачи на консоль. Задания для реализации метода выполняемого в первом объекте класса Task:

public class Program

{

public static async Task Main(string[] args)

{

Console.WriteLine("Введите трёх значное число");

int number = int.Parse(Console.ReadLine()); // Исходное трехзначное число

// Создание первого объекта задачи

Task<int[]> firstTask = Task.Run(() =>

{

// Выполнение вычислений - получение шести чисел, образованных при перестановке цифр заданного числа

int[] permutations = GetNumberPermutations(number);

return permutations;

});

// Создание второго объекта задачи продолжения

Task continuationTask = firstTask.ContinueWith(previousTask =>

{

// Получение результата из первой задачи и вывод на консоль

int[] permutations = previousTask.Result;

Console.WriteLine("Результат первой задачи:");

foreach (int permutation in permutations)

{

Console.WriteLine(permutation);

}

});

// Ожидание завершения второй задачи

await continuationTask;

Console.WriteLine("Вторая задача завершена");

}

private static int[] GetNumberPermutations(int number)

{

int[] permutations = new int[6];

int a = number / 100;

int b = number / 10 % 10;

int c = number % 10;

for (int i = 0; i < 6; )

{

permutations[i++] = a \* 100 + b \* 10 + c;

permutations[i++] = a \* 100 + c \* 10 + b;

permutations[i++] = c \* 100 + b \* 10 + a;

permutations[i++] = c \* 100 + a \* 10 + b;

permutations[i++] = b \* 100 + a \* 10 + c;

permutations[i++] = b \* 100 + c \* 10 + a;

}

return permutations;

}

}

Анализ результатов:

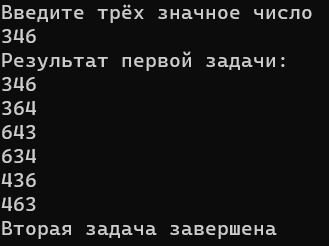
**

Рисунок 1.2 – Результат работы программы

работы программы