1. **Параллельное программирование**

Задание 1. Дано четырехзначное число. Найти число, образуемое при

перестановке второй и третьей цифр заданного числа.

Листинг программы:

using System.Diagnostics;

try

{

Console.Write("Введите строку: ");

string text = Console.ReadLine();

Stopwatch sw = Stopwatch.StartNew();

var task1 = Task.Run(() =>

{

char[] arr = text.ToCharArray();

char temp = arr[1];

arr[1] = arr[2];

arr[2] = temp;

text = String.Join("", arr);

});

var task2 = new Task(() =>

{

char[] arr = text.ToCharArray();

char temp = arr[1];

arr[1] = arr[2];

arr[2] = temp;

text = String.Join("", arr);

});

task2.Start();

var task3 = Task.Factory.StartNew(() =>

{

char[] arr = text.ToCharArray();

char temp = arr[1];

arr[1] = arr[2];

arr[2] = temp;

text = String.Join("", arr);

});

task1.Wait();

Console.WriteLine($"Полученная строка: {text}");

task2.Wait();

Console.WriteLine($"Полученная строка: {text}");

task3.Wait();

Console.WriteLine($"Полученная строка: {text}");

sw.Stop();

Console.WriteLine(sw.Elapsed);

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 20.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| abcd | acbd |

Анализ результатов:

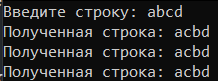


Рисунок 20.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Создайте массив из 2 задач (объектов класс Task) в каждом объекте выполните вычисление значения функций и выполните условия:

1. Дождитесь выполнения всех задач;

2. Дождитесь выполнения хот бы одной задачи

Замедлить выполнение задачи можно с помощью Thread.Sleep(n) в методе, выполняемом задачей; где n – время в миллисекундах.

Листинг программы:

Task[] tasks = new Task[2];

Console.Write("Введит a: ");

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

double z1 = 0;

tasks[0] = Task.Run( () =>

{

a = a \* Math.PI / 180;

z1 = (Math.Sin(2 \* a) + Math.Sin(5 \* a) - Math.Sin(3 \* a)) / (Math.Cos(a) + 1 - (2 \* Math.Pow(Math.Sin(2 \* a), 2)));

});

tasks[0].Wait();

Console.WriteLine($"z1 = {z1}");

double z2 = 0;

tasks[1] = Task.Run(() =>

{

z2 = 2 \* Math.Sin(a);

Thread.Sleep(3000);

});

tasks[1].Wait();

Console.WriteLine($"z2 = {z2}");

Таблица 20.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 90 | 2, 2 |

Анализ результатов:

C:\Users\1234321\Downloads\govno45.png

Рисунок 20.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Дано трёхзначное число. Найти сумму его первой и второй цифр.

Листинг программы:

try

{

Console.Write("Введите число: ");

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

int sum = 0;

var task1 = Task.Run(() =>

{

sum += number / 100 + ((number / 10) % 10);

});

Task task2 = task1.ContinueWith(t => Console.WriteLine($"Сумма первой и второй цифры числа {number} = {sum}"));

task2.Wait();

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Таблица 20.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 | 3 |

Анализ результатов:

C:\Users\1234321\Downloads\govno46.png

Рисунок 20.3 – Результат работы программы

Источник: собственная разработка

Задание 4. Выполните параллельное вычисления значений функции для каждого значения на отрезке [А,B]. Используя метод Parallel.For.

Листинг программы:

const int A = -1;

const int B = 9;

try

{

Parallel.For(A, B, x => {

if (x != 0)

Console.WriteLine($"{x} элемент = {Math.Cos(1 / x)}");

});

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Анализ результатов:

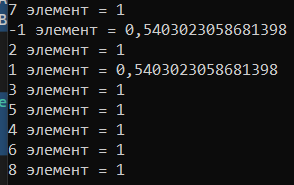


Рисунок 20.4 – Результат работы программы

Источник: собственная разработка

Задание 5. Выполните прерывание выполнения метода Parallel.Foreach при некотором условии.

Функции суммы и произведение чисел от 0 до N где N значения из массива (списка).

Листинг программы:

List<int> list = new List<int>() { 34, 7, 87, 12 };

try

{

Parallel.ForEach<int>(list, (x, state) =>

{

if (x \* 10 > 200)

{

Console.WriteLine("Выполнение прервано");

state.Break();

}

Console.WriteLine(x);

});

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

Анализ результатов:

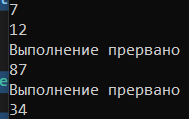


Рисунок 20.5 – Результат работы программы

Источник: собственная разработка