**20 ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ**

Задание 1. Дано трехзначное число. Найти произведение его цифр.

Листинг программы:

static void Main(string[] args)

{

Console.Write("n = ");

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

Task task1 =Task.Run(() =>

{

if (((n / 1000) > 0) || ((n < 99))) Console.WriteLine("Число не трехзначное");

else

{ int prod = 1;

while (n > 0)

{ int rem;

n = Math.DivRem(n, 10, out rem);

prod \*= rem; }

Console.WriteLine($"произведение чисел = {prod}"); } });

task1.Wait();

}

Таблица 20.1 – Выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| n=146 | Произведение чисел=24 |

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:

дождитесь выполнения всех задач;

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:

class Program

{

static void Main(string[] args)

{

double x = 6;

Console.WriteLine($"Введино число {x} ");

Task task1 = Task.Run(() =>

{

double z1 = (Math.Pow(x, 10) + x \* 2 - 3 + (x + 1) \* Math.Sqrt(Math.Pow(x, 2) - 9)) / (Math.Pow(x, 2) - x \* 2 - 3 + (x - 1) \* Math.Sqrt(Math.Pow(x, 2) - 9));

global::System.Console.WriteLine(z1);

});

Task task2 = Task.Run(() =>

{

double z2 = Math.Sqrt((x + 3) / (x - 3));

global::System.Console.WriteLine(z2);

});

task1.Wait();

task2.Wait();

}

}

}

Таблица 20.2 – Входные и выходные данные

|  |
| --- |
| Выходные данные |
| Введено число 6  1,7320508075688772  1287042,1562541919 |

Анализ результатов:



Рисунок 20.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее в конце. Найти полученное число.

Листинг программы:

internal class Program

{

static void Main(string[] args)

{

Console.WriteLine("Введите трехзначное число");

Console.Write("n = ");

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

int result = 0;

Task task1 = Task.Run(() =>

{

if (((n / 1000) > 0) || ((n < 99))) Console.WriteLine("Число не трехзначное");

else

{

result = (n % 100) \* 10 + n / 100;

}

});

Task task2 = task1.ContinueWith(task1 =>

{

Console.WriteLine("Полученное трехзначное число: {0}", result);

});

task1.Wait();

}

}

}

Таблица 20.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Введите число n = 146 | Получено трехзначное число: 461 |

Анализ результатов:



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:

class Program

{

static void Cos(int x)

{

Console.WriteLine(Math.Cos(x));

}

static void Main(string[] args)

{

Console.Write("Введите x: ");

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

Parallel.For(-7, 3, Cos);

Console.ReadLine();

}

}

}

Таблица 20.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Введите x = 14 | -0,6536436208636119  0,28366218546322625  0,5403023058681398  0,960170286650366  0,7539022543433046  -0,4161468365471424  1  -0,9899924966004454  -0,4161468365471424  0,5403023058681398 |

Анализ результатов:

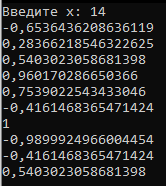


Рисунок 20.4 – Результат работы программы

Источник: собственная разработка

Задание 4. 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:

class Program

{

static void Sum(int x)

{

int result = 0;

for (int i = 0; i < x; i++)

{

result += i;

}

Console.WriteLine($"Сумма = {result}");

}

static void Mul(int x)

{

int result = 1;

for (int i = 0; i < x; i++)

{

result \*= i;

}

Console.WriteLine($"Произведение = {result}");

}

static void Main(string[] args)

{

ParallelLoopResult result1 = Parallel.ForEach<int>(new List<int>() { 54, 17, 25, 66 }, Sum);

ParallelLoopResult result2 = Parallel.ForEach<int>(new List<int>() { 54, 17, 25, 66 }, Mul);

}

}

}

Таблица 20.5 – Входные и выходные данные

|  |
| --- |
| Выходные данные |
| Сумма = 2145  Сумма = 1431  Сумма = 136  Сумма = 300  Произведение = 0  Произведение = 0  Произведение = 0  Произведение = 0 |

Анализ результатов:

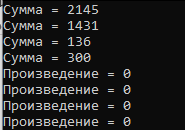


Рисунок 20.5 – Результат работы программы

Источник: собственная разработка