**20 ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ**

Задание 1. Дано четырехзначное число. Найти число, образуемое при перестановке двух первых и двух последних цифр заданного числа.

Листининг программы:

using System;

using System.Threading.Tasks;

namespace task1

{

class Program

{

static int Repl(int number)

{

return (number % 10 \* 10 + number % 100 / 10 + number / 1000 \* 100 + number / 100 % 10 \* 1000);

}

static void Main(string[] args)

{

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

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

Task task1 = new Task(()=>Console.WriteLine("1 способ: "+Repl(number)));

task1.Start();

task1.Wait();

Task task2 = Task.Factory.StartNew(() => Console.WriteLine("2 способ: " + Repl(number)));

task2.Wait();

Task task3 = Task.Run(() => Console.WriteLine("3 способ: " + Repl(number)));

task3.Wait();

}

}

}

Таблица 20.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 4321 | 1 способ: 3412  2 способ: 3412  3 способ: 3412 |

Анализ результатов:

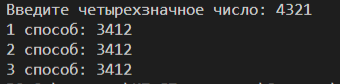


Рисунок 20.1 – Результат работы программы

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.

Листининг программы:

using System;

using System.Threading;

using System.Threading.Tasks;

namespace task2

{

class Program

{

static double Z1(double a)

{

return ((Math.Sin(2 \* a) + Math.Sin(5 \* a) - Math.Sin(3 \* a)) / (Math.Cos(a) - Math.Cos(3 \* a) + Math.Cos(5 \* a)));

}

static double Z2(double a)

{

return (2 \* Math.Sin(a));

}

static void Main(string[] args)

{

Console.Write("Введите число: ");

double a = Convert.ToInt32(Console.ReadLine());

Task[] tasks = new Task[2]

{

new Task(() => Console.WriteLine("1 задача: "+Z1(a))),

new Task(() =>

{

Thread.Sleep(1000);

Console.WriteLine("2 задача: "+Z2(a));

})

};

foreach (var t in tasks)

{

t.Start();

t.Wait();

}

}

}

}

Таблица 20.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 30 | 1 задача: -1,4702821055397988  2 задача: -1,9760632481857237 |

Анализ результатов:



Рисунок 20.2 – Результат работы программы

Источник – собственная разработка

Задание 3. Дано трёхзначное число. Найти произведение его второй и последней цифр.

Листининг программы:

using System;

using System.Threading;

using System.Threading.Tasks;

namespace task3

{

class Program

{

static int Calculation(int number)

{

return ( (number%10) \* (number/10%10) );

}

static void Main(string[] args)

{

Console.Write("Введите число: ");

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

Task task1 = Task.Factory.StartNew(() =>

{

Console.WriteLine("Первая задача началась");

Task<int> task2 = Task<int>.Factory.StartNew(() =>

{

Console.WriteLine("Вторая задача началась");

return Calculation(a);

});

task2.Wait();

Console.WriteLine("Вторая задача завершилась");

Console.WriteLine(task2.Result);

});

task1.Wait();

Console.WriteLine("Первая задача завершилась");

}

}

}

Таблица 20.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 20 | Первая задача началась  Вторая задача началась  Вторая задача завершилась  0  Первая задача завершилась |

Анализ результатов:

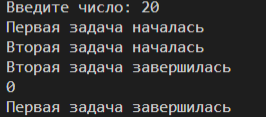


Рисунок 20.3 – Результат работы программы

Источник – собственная разработка

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

Листининг программы:

using System;

using System.Threading.Tasks;

namespace task4

{

class Program

{

static void Factorial(int x)

{

double result = Math.Sin(Math.Pow(x, 2));

Console.WriteLine($"Значение функции sin(x^2) при x={x} будет равно {Math.Round(result,3)}");

}

static void Main(string[] args)

{

Console.Write("Введите A: ");

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

Console.Write("Введите B(оно должно быть больше А): ");

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

Parallel.For(a, b+1, Factorial);

}

}

}

Таблица 20.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 20, 30 | Значение функции sin(x^2) при x=20 будет равно -0,851  Значение функции sin(x^2) при x=22 будет равно 0,194  Значение функции sin(x^2) при x=27 будет равно 0,15  Значение функции sin(x^2) при x=21 будет равно 0,923  Значение функции sin(x^2) при x=25 будет равно 0,176  Значение функции sin(x^2) при x=23 будет равно 0,936  Значение функции sin(x^2) при x=24 будет равно -0,886  Значение функции sin(x^2) при x=28 будет равно -0,985  Значение функции sin(x^2) при x=26 будет равно -0,529  Значение функции sin(x^2) при x=29 будет равно -0,812  Значение функции sin(x^2) при x=30 будет равно 0,998 |

Анализ результатов:

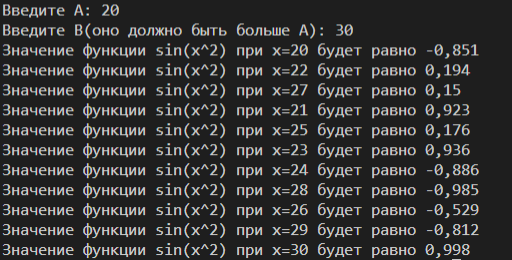


Рисунок 20.4 – Результат работы программы

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).

Условия выхода, т.е. значения вычисления функций выбираете самостоятельно.

Пример условия, массив значений {3,5,9,30,67}. Выполняется выход из параллельного вычисления суммы или произведения для этих значений при условии, если сумма больше 535, а произведение больше 535.

Листининг программы:

using System;

using System.Threading.Tasks;

using System.Collections.Generic;

namespace task5

{

class Program

{

static void Factorial(int x, ParallelLoopState pls)

{

long work = 1;

for (int i = 1; i <= x; i++)

{

work \*= i;

}

int sum = 0;

for (int i = 1; i <= x; i++)

{

sum += i;

}

if ((work > 300) && (sum > 300))

{

pls.Break();

}

Console.WriteLine($"Сумма: {sum}, произведение: {work}");

}

static void Main(string[] args)

{

ParallelLoopResult result = Parallel.ForEach<int> (new List<int>() { 47, 16, 34, 87, 23 }, Factorial);

if (!result.IsCompleted)

Console.WriteLine($"Выполнение цикла завершено на итерации {result.LowestBreakIteration}");

}

}

}

Таблица 20.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Сумма: 595, произведение: 4926277576697053184  Сумма: 1128, произведение: -1274672626173739008  Сумма: 136, произведение: 20922789888000  Выполнение цикла завершено на итерации 2 |

Анализ результатов:

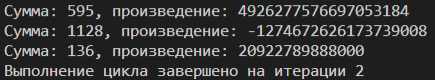


Рисунок 20.5 – Результат работы программы

Источник – собственная разработка