**20. Параллельное программирование**

Задание №1. Создайте и реализуйте метод решения задачи и выполните его в объектах класса Task используя три варианта создания объектов класса Task. Дано четырехзначное число. Найти число, полученное при прочтении его цифр справа налево.

Листинг программы:

void ReverseNumber(int number)

{ int thousands, thousands2, hundreds, hundreds2, dozens, dozens2, units;

dozens = 1;

if ((number > 10) && (number < 1000) && (dozens != 0))

{Console.WriteLine("Введите другое число");}

else

{ thousands = number / 1000;

thousands2 = number % 1000;

hundreds = thousands2 / 100;

hundreds2 = thousands2 % 100;

dozens = hundreds2 / 10;

dozens2 = hundreds2 % 10;

units = dozens2 % 10;

Console.WriteLine($"{units} {dozens} {hundreds} {thousands}");}}

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

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

var task = Task<int>.Run(() => ReverseNumber(number));

Thread.Sleep(1000);

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

task1.Start();

Thread.Sleep(1000);

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

Thread.Sleep(1000);

while (!task1.IsCompleted)

{ Thread.Sleep(2000);

Console.WriteLine("Wait...");

Console.Clear();}

Таблица 20.1 – Входные и выходные данных задание №1

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1234 | 4321 |

Анализ результатов:

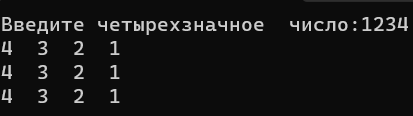


Рисунок 20.1 – Результат работы программы из задания №1

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

Функции для выполнения задачи представлены на рисунке 20.2



Рисунок 20.2- Функции для выполнения задания №2

Листинг программы:

Console.Write("Введите a:");

double a = double.Parse(Console.ReadLine());

Task[] tasks1 = new Task[2]

{new Task(() =>Console.WriteLine( ResultFuncZ1(a, 2000))),

new Task(() => Console.WriteLine( ResultFuncZ2(a,5000)))};

foreach (var t in tasks1)

t.Start();

Task.WaitAny(tasks1);

Console.WriteLine("First");

Task.WaitAll(tasks1);

Console.WriteLine("End");

double ResultFuncZ2(double a,int milliseconds)

{ Thread.Sleep(milliseconds);

Console.WriteLine(Thread.CurrentThread.Name);

return 4 - Math.Pow(a, 2) / 2;}

double ResultFuncZ1(double a,int milliseconds)

{ Thread.Sleep(milliseconds);

Console.WriteLine(Thread.CurrentThread.Name);

return Math.Pow(((1 + a + Math.Pow(a, 2)) / ((2 \* a) + Math.Pow(a, 2))) \* (5 - 2 \* Math.Pow(a, 2)),-1);}

Таблица 20.2 – Входные и выходные данных задание №2

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| A=2 | -0,38095238095238093  2 |

Анализ результатов:

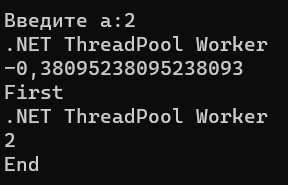


Рисунок 20.3 – Результат работы программы из задания №2

Источник: собственная разработка

Задание №3. Создайте два объекта класса Task. Первый объект возвращает результат вычисления, второй объект является задачей продолжения первого объекта и выводит результат первой задачи на консоль. Дано трёхзначное число. Найти произведение его цифр.

Листинг программы

Console.Write("Введите число:");

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

Task <int> task1 = new Task<int>(() => MulNumber(number));

Task task2 = task1.ContinueWith(task=>Print(task.Result));

task1.Start();

task2.Wait();

static int MulNumber(int number)

{ int hundreds, hundreds2, dozens, dozens2, units;

int mul = 1;

dozens = 1;

if ((number > 10) && (number < 100) && (dozens != 0))

{ Console.WriteLine("Введите другое число");}

else

{ hundreds = number / 100;

hundreds2 = number % 100;

dozens = hundreds2 / 10;

dozens2 = hundreds2 % 10;

units = dozens2 % 10;

mul = units \* dozens \* hundreds;

Console.WriteLine($"Произведение цифр число {number} = {mul} "); }

return mul;}

static void Print(int mulNumber)

{ Console.WriteLine($"Произведение: {mulNumber}");}

Листинг программы:

Таблица 20.3 – Входные и выходные данных задание №3

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 234 | Произведение:24 |

Анализ результатов:

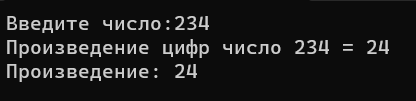


Рисунок 20.4 – Результат работы программы из задания №3

Источник: собственная разработка

Задание №4. Выполните параллельное вычисления значений функции Xsin x

для каждого значения на отрезке [0,10]. Используя метод Parallel.For.

Листинг программы:

static void Sin(int x)

{ Console.WriteLine($"Выражение равно = { x\* Math.Sin(x)}");

Thread.Sleep(1000);}

Console.Write("Введите х:");

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

Parallel.For(0, 10, Sin);

Таблица 20.4 – Входные и выходные данных задание №4

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Х=3 | Выражение равно = 0  Выражение равно = 7,914865972987054 |

Анализ результатов:

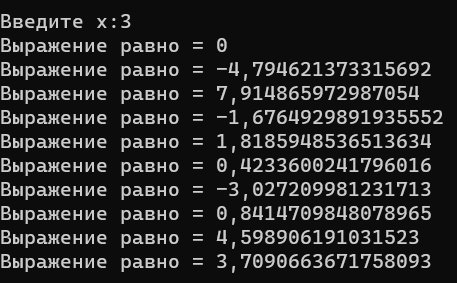


Рисунок 12.5 – Результат работы программы из задания №4

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). {65,2,6,87}

Листинг программы:

void Sum(int n, ParallelLoopState pls)

{ int result = 0;

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

{if (result >= 10)

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

pls.Break();

break;}

result += i;}}

void Mul(int n, ParallelLoopState pls)

{ int result = 1;

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

{ if (result >= 10)

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

pls.Break();

break;}

result \*= i;}}

ParallelLoopResult result = Parallel.ForEach<int>(new List<int>() { 65, 2, 6, 87 }, Sum);

ParallelLoopResult result2 = Parallel.ForEach<int>(new List<int>() { 65, 2, 6, 87 }, Mul);

if (!result.IsCompleted)

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

Таблица 20.5 – Входные и выходные данных задание №5

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Сумма = 10  Произведение = 24 |

Анализ результатов:

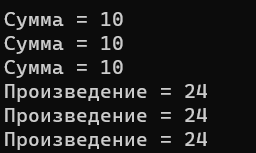


Рисунок 20.6 – Результат работы программы из задания №5

Источник: собственная разработка