**20 «ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРВАНИЕ».**

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, образуемое при перестановке двух первых и двух последних цифр заданного числа.

Листинг программы:

using System;

public class Task

{

public int SwapDigits(int number)

{

if (number < 1000 || number > 9999)

{

throw new ArgumentException("Number must be a four-digit number.");

}

int firstTwoDigits = number / 100;

int lastTwoDigits = number % 100;

return lastTwoDigits \* 100 + firstTwoDigits;

}

}

class Program

{

static void Main(string[] args)

{

Task task1 = new Task();

Console.WriteLine(task1.SwapDigits(1234)); // Output: 3412

Task task2 = new Task();

Console.WriteLine(task2.SwapDigits(5678)); // Output: 7856

Task task3 = new Task();

Console.WriteLine(task3.SwapDigits(9012)); // Output: 1290

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 1234  5678  9012 | 3412  7856  1290 |
|  |  |

Анализ результатов:

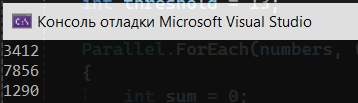


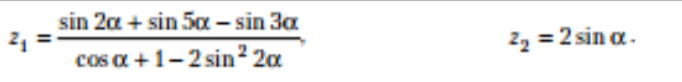
Рисунок 1.1 – Результат работы программы

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:

using System;

using System.Threading;

using System.Threading.Tasks;

public class Task

{

private double \_a;

public Task(double a)

{

\_a = a;

}

public double CalculateZ1()

{

Thread.Sleep(1000); // Замедление выполнения задачи на 1 секунду

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

}

public double CalculateZ2()

{

Thread.Sleep(1000); // Замедление выполнения задачи на 1 секунду

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

}

}

class Program

{

static void Main(string[] args)

{

Task[] tasks = new Task[2]

{

new Task(0.5),

new Task(1.0)

};

// Дождитесь выполнения всех задач

System.Threading.Tasks.Task[] allTasks = new System.Threading.Tasks.Task[2]

{

System.Threading.Tasks.Task.Run(() => Console.WriteLine($"z1 = {tasks[0].CalculateZ1()}")),

System.Threading.Tasks.Task.Run(() => Console.WriteLine($"z2 = {tasks[1].CalculateZ2()}"))

};

System.Threading.Tasks.Task.WaitAll(allTasks);

// Дождитесь выполнения хотя бы одной задачи

System.Threading.Tasks.Task[] anyTask = new System.Threading.Tasks.Task[2]

{

System.Threading.Tasks.Task.Run(() => Console.WriteLine($"z1 = {tasks[0].CalculateZ1()}")),

System.Threading.Tasks.Task.Run(() => Console.WriteLine($"z2 = {tasks[1].CalculateZ2()}"))

};

System.Threading.Tasks.Task.WaitAny(anyTask);

}

}

Таблица 1.2 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | z2 = 1,682941969615793  z1 = 77,59872941884666  z1 = 77,59872941884666  z2 = 1,682941969615793 |

Анализ результатов:

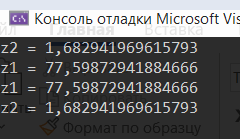


Рисунок 1.2 – Результат работы программы

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. Задания для реализации метода выполняемого в первом объекте класса Task: Дано трёхзначное число. Найти произведение его второй и последней цифр.

Листинг программы:

using System;

using System.Threading.Tasks;

public class Task

{

private int \_number;

public Task(int number)

{

if (number < 100 || number > 999)

{

throw new ArgumentException("Number must be a three-digit number.");

}

\_number = number;

}

public int CalculateProduct()

{

int secondDigit = (\_number / 10) % 10;

int lastDigit = \_number % 10;

return secondDigit \* lastDigit;

}

}

class Program

{

static void Main(string[] args)

{

Task task1 = new Task(123);

System.Threading.Tasks.Task<int> t1 = System.Threading.Tasks.Task.Run(() => task1.CalculateProduct());

System.Threading.Tasks.Task t2 = t1.ContinueWith(t => Console.WriteLine($"The product of the second and last digit is: {t.Result}"));

t2.Wait();

}

}

Таблица 1.3 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 | The product of the second and last digit is: 6 |

Анализ результатов:

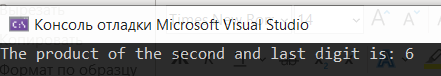


Рисунок 1.3 – Результат работы программы

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



Листинг программы:

using System;

using System.Threading.Tasks;

class Program

{

static void Main(string[] args)

{

int A = -5;

int B = 16;

Parallel.For(A, B + 1, i =>

{

double x = i;

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

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

});

}

}

Таблица 1.4 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | f(-1) = 0,8414709848078965  f(-5) = -0,13235175009777303  f(-4) = -0,2879033166650653  f(-2) = -0,7568024953079282  f(-3) = 0,4121184852417566  f(12) = -0,49102159389846933  f(1) = 0,8414709848078965  f(2) = -0,7568024953079282  f(15) = -0,9300948780045254  f(16) = -0,9992080341070627  f(5) = -0,13235175009777303  f(6) = -0,9917788534431158  f(7) = -0,9537526527594719  f(8) = 0,9200260381967907  f(9) = -0,6298879942744539  f(10) = -0,5063656411097588  f(11) = 0,9988152247235795  f(13) = -0,6019998676776046  f(0) = 0  f(14) = 0,9395300555699313  f(3) = 0,4121184852417566  f(4) = -0,2879033166650653 |

Анализ результатов:

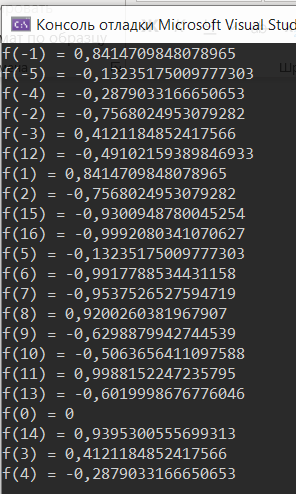


Рисунок 1.4 – Результат работы программы

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). {47,16,34,87,23}

Листинг программы:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading;

using System.Threading.Tasks;

class Program

{

static void Main(string[] args)

{

int[] numbers = { 47, 16, 34, 87, 23 };

int threshold = 13;

Parallel.ForEach(numbers, (number, state) =>

{

int sum = 0;

long product = 1;

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

{

sum += i;

product \*= i;

if (sum > threshold)

{

Console.WriteLine($"Execution stopped: the sum exceeded the threshold of {threshold}.");

state.Stop();

return;

}

}

Console.WriteLine($"For N = {number}: Sum = {sum}, Product = {product}");

});

}

}

Таблица 1.5 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3, 16, 34, 87, 23 | Execution stopped: the sum exceeded the threshold of 13.  Execution stopped: the sum exceeded the threshold of 13.  Execution stopped: the sum exceeded the threshold of 13.  Execution stopped: the sum exceeded the threshold of 13.  For N = 3: Sum = 6, Product = 0 |

Анализ результатов:

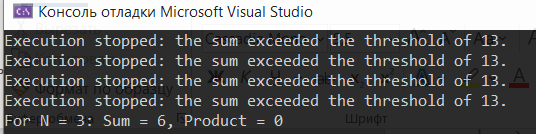


Рисунок 1.5 – Результат работы программы