**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 трехзначное число. В нем зачеркнули последнюю справа цифру и приписали ее в начале. Найти полученное число. Найти число, полученное при перестановке первой и второй цифр заданного числа. Найти произведение его цифр.

Листинг программы:

static void Main(string[] args) {Console.Write("Введите трехзначное число: ");

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

Task<int> task1 = Task.Run(() => GetNumberWithMovedDigit(number));

Task<int> task2 = Task.Factory.StartNew(() => GetNumberWithSwappedDigits(number));

Task<int> task3 = new Task<int>(() => GetProductOfDigits(number));

task3.Start(); Task.WaitAll(task1, task2, task3);

Console.WriteLine($"Число со сдвинутой цифрой: {task1.Result}");

Console.WriteLine($"Число с переставленными цифрами: {task2.Result}");

Console.WriteLine($"Произведение цифр: {task3.Result}");}

static int GetNumberWithMovedDigit(int number)

{int lastDigit = number % 10; number /= 10; int firstDigits = number / 10;

return lastDigit \* 100 + firstDigits \* 10 + number % 10;}

static int GetNumberWithSwappedDigits(int number){

int lastDigit = number % 10; number /= 10; int middleDigit = number % 10;

number /= 10; return lastDigit \* 100 + middleDigit \* 10 + number;}

static int GetProductOfDigits(int number){ int product = 1;while (number > 0)

{product \*= number % 10; number /= 10;} return product;}

Таблица 20.1 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 | Число со сдвинутой цифрой: 312  Число с переставленными цифрами: 321  Произведение цифр: 6 |

Источник: собственная разработка

Анализ результатов:

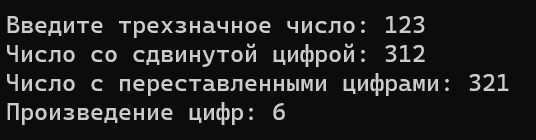


Рисунок 20.1 – Результат работы программы

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. Функция для решения задачи представлена на рисунке 20.2



Рисунок 20.2 – Функция Z1 и Z2

Источник: задание 2

Листинг программы:

Console.Write("Введите значение b: ");

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

Task<double> task1 = Task.Run(() =>{Thread.Sleep(1000);

return (Math.Sqrt(2 \* b + 2 \* Math.Sqrt(Math.Pow(b, 2) - 4))) / (Math.Sqrt(Math.Pow(b, 2) - 4) + b + 2);});

Task<double> task2 = Task.Run(() =>{Thread.Sleep(1000);

return 1 / (Math.Sqrt(b + 2));});

Task.WaitAll(task1, task2);

Console.WriteLine($"z1 = {task1.Result}");Console.WriteLine($"z2 = {task2.Result}");

Task<double> anyTask = Task.WhenAny(task1, task2).Result;

Console.WriteLine($"Первая выполненная задача: {anyTask.Result}");

Таблица 20.2 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 2 | z1 = 0,5  z2 = 0,5  Первая выполненная задача: 0,5 |

Источник: собственная разработка

Анализ результатов:

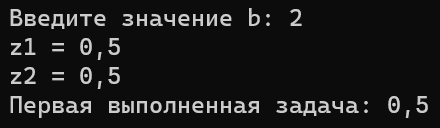


Рисунок 20.3 – Результат работы программы

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. Задания для реализации метода выполняемого в первом объекте класса Task: Дано трехзначное число. Найти число, полученное при прочтении его цифр справа налево.

Листинг программы:

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

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

Task<int> task1 = new Task<int>(() =>{int reverseNumber = 0;

while (number > 0){int remainder = number % 10;

reverseNumber = reverseNumber \* 10 + remainder; number /= 10;}

return reverseNumber;});

Task task2 = task1.ContinueWith((Task<int> t) =>{

Console.WriteLine("Число, полученное при прочтении цифр справа налево: " + t.Result);}); task1.Start(); Task.WaitAll(task1, task2);

Таблица 20.3 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 123 | Число, полученное при прочтении цифр справа налево: 321 |

Источник: собственная разработка

Анализ результатов:

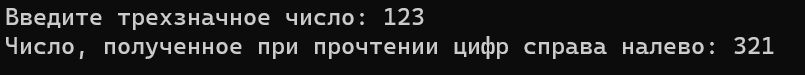


Рисунок 20.4 – Результат работы программы

Источник: собственная разработка

Задание 4. Выполните параллельное вычисления значений функции ArctgX для каждого значения на отрезке [-10,1]. Используя метод Parallel.For.

Листинг программы:

int a = -10, b = 1; int N = 11; int h = (b - a) / N;

Parallel.For(0, N, i =>{int x = a + i \* h; int y = Math.Atan(x);

Console.WriteLine("arctan({0:f2}) = {1:f2}", x, y); });

Таблица 20.4 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| -10; 1 | arctan(-10,00) = -1,47  arctan(-9,00) = -1,46  arctan(-1,00) = -0,79  arctan(0,00) = 0,00  arctan(-7,00) = -1,43  arctan(-5,00) = -1,37  arctan(-8,00) = -1,45  arctan(-2,00) = -1,11  arctan(-6,00) = -1,41  arctan(-3,00) = -1,25  arctan(-4,00) = -1,33 |

Источник: собственная разработка

Анализ результатов:

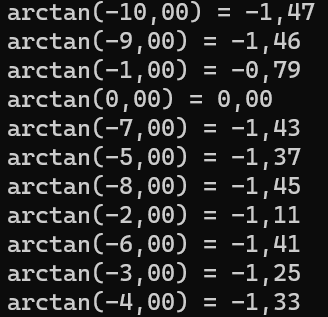


Рисунок 20.5 – Результат работы программы

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 {3,70,54}.

Листинг программы:

int[] values = { 3, 70, 54 }; CancellationTokenSource cts = new CancellationTokenSource(); ParallelOptions options = new ParallelOptions { CancellationToken = cts.Token }; int sum = 0; int product = 1;

try{ Parallel.ForEach(values, options, (n) => {

if (options.CancellationToken.IsCancellationRequested)

{throw new OperationCanceledException(options.CancellationToken);}

sum += n; product \*= n;});}

catch (OperationCanceledException)

{Console.WriteLine("Выполнение было прервано.");}

Console.WriteLine($"Сумма: {sum}");

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

Таблица 20.5 – Выходные и входные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  | Сумма: 127  Произведение: 11340 |

Источник: собственная разработка

Анализ результатов:

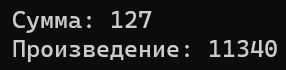


Рисунок 20.6 – Результат работы программы

Источник: собственная разработка