

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное учреждение высшего образования  
«Чувашский государственный университет И.Н. Ульянова»  
Факультет информатики и вычислительной техники  
Кафедра вычислительной техники

Параллельное программирование  
Лабораторная работа 1  
Выполнение заданий с 1 – 5

**Выполнил:**

Студент группы ИВТ-41-20  
Галкин Д.С.

**Проверил:**

Ковалев С.В.

## Цель работы (Технология программирования в System.Threading.Tasks):

### Задание 1. Создание проекта в среде Rider с поддержкой System.Threading.Tasks:

#### Задание для выполнения

Создайте проект в среде Rider с поддержкой System.Threading.Tasks

### Задание 2. Многопоточная программа "Hello World!":

#### Задание для выполнения

Напишите программу, в которой создается 4 нити и каждая нить выводит на экран строку "Hello World!"

Входные данные: нет

Выходные данные: 4 строки "Hello World!"

Входные данные	Выходные данные
	Hello World!
	Hello World!
	Hello World!
	Hello World!

#### Полный текст программы:

1. Интерфейс ILab (используется во всех лабораторных работах)

```
using System.Runtime.InteropServices;

namespace ParallelsProgramming.Intrefeces;

public interface ILab
{
    public string Name { get; set; }

    /// <summary>
    /// Задание лабораторной работы    /// </summary>    public void ThreadStart();

    /// <summary>
    /// Запуск лабораторной работы    /// </summary>    public void Start();
}
```

2. Класс Main

```
var lab1 = new Lab1("Lab1");
lab1.Start();
```

### 3. Класс Lab1

```
using ParallelsProgramming.Intrefeces;

namespace ParallelsProgramming.Labs;

public class Lab1 : ILab
{
    public string Name { get; set; }

    public Lab1(string name)
    {
        Name = name;
    }

    public void ThreadStart()
    {
        Parallel.For(0, 4, i =>
        {
            Print();
        });
    }

    public void Start()
    {
        Console.WriteLine(this);
        ThreadStart();
        Console.WriteLine("\n");
    }

    public override string ToString()
    {
        return $"{Name} started classes:";
    }

    private void Print() => Console.WriteLine("Hello world!");
}
```

**Результат:**

```
Lab1 started classes:
Hello world!
Hello world!
Hello world!
Hello world!
```

### Задание 3. Программа "I am!":

#### Задание для выполнения

Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер и общее количество нитей в параллельной области в формате:

"I am <Номер нити> thread from <Количество нити> threads!"

**Входные данные:** k – количество нитей в параллельной области

**Выходные данные:** k строка вида "I am <Номер нити> thread from <Количество нитей> threads!"

Входные данные	Выходные данные
3	I am 0 thread from 3 threads!
	I am 1 thread from 3 threads!
	I am 2 thread from 3 threads!

#### Задание для выполнения

Модифицируйте программу таким образом, чтобы строку "I am <Номер нити> thread from <Количество нитей> threads!" выводили только нити с четным номером

Входные данные	Выходные данные
3	I am 0 thread from 3 threads!
	I am 2 thread from 3 threads!

**Полный текст программы:**

1. Класс Main

```
var lab2 = new Lab2("Lab2");  
lab2.Start();  
  
var lab2 = new Lab2("Lab2", true);  
lab2.Start();
```

## 2. Класс Lab2

```
using ParallelsProgramming.Intrefeces;

namespace ParallelsProgramming.Labs;

public class Lab2 : ILab
{
    public string Name { get; set; }
    private readonly bool _isPos;

    public Lab2(string name, bool isPos = false)
    {
        Name = name;
        _isPos = isPos;
    }
    public void ThreadStart()
    {
        Console.WriteLine("Введите кол-во нитей: ");

        if (int.TryParse(Console.ReadLine(), out int threads) && threads > 0)
        {
            Parallel.For(0, threads, i =>
            {
                if (!_isPos || i % 2 == 0)
                {
                    Print(i, threads);
                }
            });
        }
        else
        {
            Console.WriteLine("Некорректный ввод. Убедитесь, что вводите положительное  
целое число.");
        }
    }

    public void Start()
    {
        Console.WriteLine(this);
        ThreadStart();
        Console.WriteLine("\n");
    }
    public override string ToString()
    {
        return $"{Name} started classes:";
    }

    private void Print(int index, int threads) => Console.WriteLine($"I am {index}  
thread from {threads} threads!");
}
```

Результат:

```
Lab2 started classes:  
Введите кол-во нитей:  
3  
I am 1 thread from 3 threads!  
I am 0 thread from 3 threads!  
I am 2 thread from 3 threads!
```

Модификация

```
Lab2 started classes:  
Введите кол-во нитей:  
3  
I am 2 thread from 3 threads!  
I am 0 thread from 3 threads!
```

## Задание 4. Общие и частные переменные в System.Threading.Tasks: программа "Скрытая ошибка":

### Задание для выполнения

Изучите конструкции для управления работой с данными shared и private. Напишите программу, в которой создается k нитей, и каждая нить выводит на экран свой номер через переменную rank следующим образом:

**Входные данные:** целое число k - кол-во нитей в параллельной области

**Выходные данные:** k строка вида "I am <Номер нити>."

Входные данные	Выходные данные
3	I am 0 thread
	I am 1 thread
	I am 2 thread

**Полный текст программы:**

1. Класс Main

```
var lab3 = new Lab3("Lab3");  
lab3.Start();
```

## 2. Класс Lab3

```
using ParallelsProgramming.Intrefeces;

namespace ParallelsProgramming.Labs;

public class Lab3 : ILab
{
    public string Name { get; set; }

    public Lab3(string name)
    {
        Name = name;
    }

    public void ThreadStart()
    {
        Console.WriteLine("Введите кол-во нитей: ");

        if (int.TryParse(Console.ReadLine(), out int threads) && threads > 0)
        {
            Parallel.For(0, threads, Print);
        }
        else
        {
            Console.WriteLine("Некорректный ввод. Убедитесь, что вводите положительное  
целое число.");
        }
    }

    public void Start()
    {
        Console.WriteLine(this);
        ThreadStart();
        Console.WriteLine("\n");
    }

    public override string ToString()
    {
        return $"{Name} started classes:";
    }

    private void Print(int index) => Console.WriteLine($"I am {index} thread.");
}
```

Результат:

```
Lab3 started classes:
Введите кол-во нитей:
3
I am 1 thread.
I am 0 thread.
I am 2 thread.
```



## Задание 5. Общие и частные переменные в System.Threading.Tasks: параметр reduction":

### 📌 Задание для выполнения

Напишите программу, в которой две нити параллельно вычисляют сумму чисел от 1 до N. Распределите по нитям с помощью оператора `if` языка C.

**Входные данные:** целое число N - количество чисел

**Выходные данные:** каждая нить выводит всю частичную сумму в формате "[Номер нити]: Sum = [Частичная сумма]", один раз выводится общая сумма в формате "Sum = [Сумма]"

Входные данные	Выходные данные
2	0: Sum = 3
4	1: Sum = 7 Sum = 10

### 📌 Задание для выполнения

Модифицируйте программу таким образом, чтобы она работала для k нитей.

**Входные данные:** целое число k - кол-во нитей, целое число N - кол-во чисел.

**Выходные данные:** каждая нить выводит всю частичную сумму в формате "[Номер нити]: Sum = [Частичная сумма]", один раз выводится общая сумма в формате "Sum = [Сумма]"

Входные данные	Выходные данные
2	0: Sum = 3
4	1: Sum = 7 Sum = 10
2	0: Sum = 1
2	1: Sum = 2 Sum = 3
3	0: Sum = 1
2	1: Sum = 2 2: Sum = 0 Sum = 3

**Полный текст программы:**

1. Класс Main

```
var lab4 = new Lab4("Lab4");  
lab4.Start();  
  
var lab4 = new Lab4("Lab4", true);  
lab4.Start();
```

## 2. Класс Lab4

```
using ParallelsProgramming.Intrefeces;

namespace ParallelsProgramming.Labs;

public class Lab4 : ILab
{
    public string Name { get; set; }
    private readonly bool _isModif;

    public Lab4(string name, bool isModif = false)
    {
        Name = name;
        _isModif = isModif;
    }

    public void ThreadStart()
    {
        Console.WriteLine("Введите число N: ");

        if (int.TryParse(Console.ReadLine(), out int N) && N > 0)
        {
            int half = N / 2;
            int sum1 = 0, sum2 = 0;
            // Создание и запуск первой задачи для вычисления суммы от 1 до half
            var task1 = Task.Run(() =>
            {
                for (int i = 1; i <= half; i++)
                {
                    sum1 += i;
                }
            });
            Print(0, sum1);
            // Создание и запуск второй задачи для вычисления суммы от half+1 до N
            var task2 = Task.Run(() =>
            {
                for (int i = half + 1; i <= N; i++)
                {
                    sum2 += i;
                }
            });
            Print(1, sum2);
            Task.WaitAll(task1, task2); // Ожидание завершения обеих задач

            int totalSum = sum1 + sum2; // Общая сумма
            Console.WriteLine($"Общая сумма: {totalSum}");
        }
        else
        {
            Console.WriteLine("Некорректный ввод. Убедитесь, что вводите положительное целое число.");
        }

        private void ThreadStartModif()
        {
            Console.WriteLine("Введите кол-во нитей K: ");
            if (!int.TryParse(Console.ReadLine(), out int K) && K <= 0)
            {
                Console.WriteLine("Некорректный ввод. Убедитесь, что вводите положительное целое число.");
                return;
            }
            Console.WriteLine("Введите число N: ");
            if (!int.TryParse(Console.ReadLine(), out int N) && N <= 0)
            {
                Console.WriteLine("Некорректный ввод. Убедитесь, что вводите положительное целое число.");
                return;
            }
        }
    }
}
```

```

    }
    int[] partialSums = new int[K];
    Task[] tasks = new Task[K];
    // Распределение работы по нитям
    for (int taskNum = 0; taskNum < K; taskNum++)
    {
        var self = N > K ? N / K : K / N;
        int localTaskNum = taskNum; // Локальная переменная для каждой итерации
        int start = localTaskNum * (self) + 1;
        int end = (localTaskNum == K - 1) ? N : (localTaskNum + 1) * (self);

        tasks[localTaskNum] = Task.Run(() =>
        {
            for (int i = start; i <= end; i++)
            {
                partialSums[localTaskNum] += i;
                Print(localTaskNum, partialSums[localTaskNum]);
            }
        });
    }
    Task.WaitAll(tasks); // Ожидание завершения всех задач

    int totalSum = 0;
    foreach (int sum in partialSums)
    {
        totalSum += sum;
    }
    Console.WriteLine($"Sum = {totalSum}"); // Вывод общей суммы
}

public void Start()
{
    Console.WriteLine(this);
    if (!_isModif)
        ThreadStart();
    else
        ThreadStartModif();
    Console.WriteLine("\n");
}

public override string ToString()
{
    return $"{Name} started classes:";
}

private void Print(int index, int sum) => Console.WriteLine($"[{index}]: Sum = {sum}");
}

```

**Результат:**

```

Lab4 started classes:
Введите число N:
4
[1]: Sum = 7
[0]: Sum = 3
Общая сумма: 10

```

```
Lab4 started classes:  
Введите кол-во нитей K:  
2  
Введите число N:  
4  
[1]: Sum = 7  
[0]: Sum = 3  
Sum = 10
```

```
Lab4 started classes:  
Введите кол-во нитей K:  
2  
Введите число N:  
2  
[1]: Sum = 2  
[0]: Sum = 1  
Sum = 3
```

Lab4 started classes:

Введите кол-во нитей K:

3

Введите число N:

2

[1]: Sum = 2

[2]: Sum = 0

[0]: Sum = 1

Sum = 3