

## Лабораторная работа 19 2. Организация многопоточной обработки на основе класса Thread

### Цель работы

Изучить представленный код.

```
class Test
{
    double progress;
    int sum;
    ссылка: 1
    public Test()
    {
        Progress = 0;
        Sum = 0;
    }

    ссылка: 0
    public Test(double progress, int sum)...

    ссылка: 6
    public double Progress...

    ссылка: 5
    public int Sum...
}
```

```
ссылка: 0
static void Main(string[] args)
{
    Func<int,Test, int> F = SumDigit;
    Test t = new Test();
    IAsyncResult result = F.BeginInvoke(152646,t, AsyncCallback, t);
    Task.Run(() =>
    {
        while (t.Progress < 99)
        {
            Console.WriteLine($"Сумма {t.Sum} прогресс {t.Progress}");
            Thread.Sleep(5000);
        }
    });
    //полезная работа
    int answer = F.EndInvoke(result);

    Console.WriteLine(answer);
    Console.ReadKey();
}
```

```

static void AsyncCallback(IAsyncResult result)
{
    Test t = result.AsyncState as Test;
    Console.WriteLine($"Async Сумма {t.Sum} прогресс {t.Progress}");
}
//ссылка:1
static int SumDigit(int value, Test t)
{
    var sum = 0;
    var count = value.ToString().Length;
    while (value > 0)
    {
        sum += value % 10;
        value /= 10;
        Thread.Sleep(2000);
        t.Progress = 100f / count * (count - value.ToString().Length + 1);
        t.Sum = sum;
    }
    return sum;
}

```

На основании этого кода необходимо реализовать следующее.

Выполнение указанного действия асинхронно через делегат, считающий сумму ряда Тейлора. В объект Test добавить возможность отмены выполнения асинхронной операции.

Пока выполняется асинхронная операция пользователь должен иметь возможность отменить её выполнения, путем нажатия клавиши.

Процент выполнения задания указывается каждый раз на очищенном экране.

Точность указывается как параметр делегата. Ряд заканчивает подсчитываться в том случае, когда следующее слагаемое будет меньше указанной точности.

## Варианты

1. найти результат операции Sin не используя функции Math.

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, \text{ с точностью } \delta, \text{ где } x < \infty$$

2. найти результат операции Cos не используя функции Math.

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots, \text{ с точностью } \delta, \text{ где } x < \infty$$

3. найти результат операции не используя функции Math.

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots, \text{ с точностью } \delta, \text{ где } x \text{ от } -1 \text{ до } 1$$

4. найти результат операции не используя функции Math.

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \dots, \text{ с точностью } \delta, \text{ где } x \text{ от } -1 \text{ до } 1$$

5. найти результат операции не используя функции Math.

$$\frac{1}{(1+x)^2} = 1 - 2x + 3x^2 - 4x^3 + 5x^4 - \dots, \text{ с точностью } \delta, \text{ где } x \text{ от } -1 \text{ до } 1.$$

6. найти результат операции не используя функции Math.

$$\frac{1}{(1+x)^3} = 1 - \frac{2 \cdot 3}{2}x + \frac{3 \cdot 4}{2}x^2 - \frac{4 \cdot 5}{2}x^3 + \dots, \text{ с точностью } \delta, \text{ где } x \text{ от } -1 \text{ до } 1.$$

Контрольные вопросы:

1. Получение результат асинхронной операции через делегат
2. Назначение слов Async Await
3. Какие языковые конструкции позволяют вызывать асинхронные операции последовательно(параллельно)?