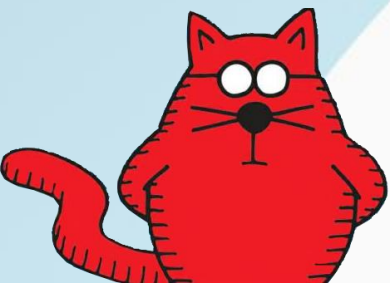


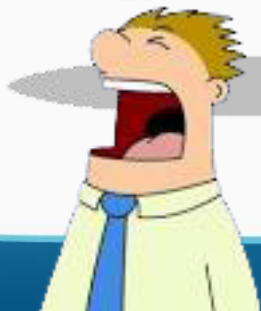
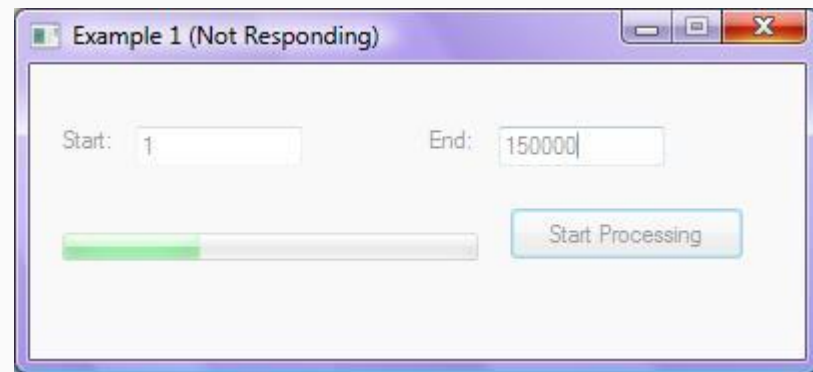


Тема: Task Parallel Library



Крыжановский Анатолий

Одного потока хватит всем?



Варианты многопоточности

System.Threading.Thread

System.Threading.ThreadPool

System.ComponentModel.
BackgroundWorker

System.Threading.Tasks.Task

Препарируем TPL – создание задачи

```
void SomeWork()  
{  
}  
}
```

```
Task.Run(() => SomeWork())
```

```
var task = new Task(() => SomeWork());  
task2.Start();
```

```
var taskFactory = new TaskFactory();  
var task = taskFactory.StartNew(() => SomeWork());
```

```
var task = Task.Factory.StartNew(() => SomeWork());
```



Цепочки задач

```
int Somework(int id)
{
    var duration = (int)Math.Round(new Random(id).NextDouble() * 3 + 1);
    Thread.Sleep(duration * 1000);
    Console.WriteLine("Work {0} done in {1} seconds", id, duration);

    return id;
}
```

```
var tasks = Enumerable
    .Range(0, 5)
    .Select(x => Task.Run(() => Somework(x)))
    .ToArray();
    .ContinueWith(t => Console.WriteLine(t.Result));

Task.WhenAny(tasks).ContinueWith(t => Console.WriteLine("any task done"));
```

Тонкая настройка

```
Console.WriteLine("Begin");
```

```
var task = Task.Run(() =>
```

IsCompleted	IsCanceled	IsFaulted	
true	false	false	NotOnFaulted, NotOnCanceled, OnlyOnRanToCompletion
false	true	false	NotOnRanToCompletion, NotOnFaulted, OnlyOnCanceled
false	false	true	NotOnRanToCompletion, NotOnCanceled, OnlyOnFaulted

```
task.Wait();
```

```
Console.WriteLine("End");
```

Подождите, подождите!!!

```
int SomeWork(int id)
{
    var duration = (int)Math.Round(new Random(id).NextDouble() * 3 + 1);
    Thread.Sleep(duration * 1000);
    Console.WriteLine("Work {0} done in {1} seconds", id,
duration);

    return id;
}
```

```
var tasks = Enumerable
    .Range(0, 5)
    .Select(x => Task.Run(() => SomeWork(x)))
    .ToList();
var result = task.Result;

Task.WaitAny(tasks, 1);
```

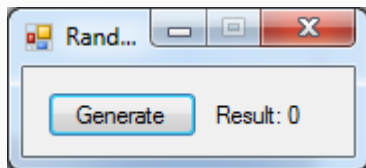


Отмена задачи

```
void Work(CancellationToken token)
{
    for (var i = 0; i < 10; i++)
    {
        token.ThrowIfCancellationRequested();
        Thread.Sleep(1000);
        Console.WriteLine("Iteration {0} done", i);
        Thread.Sleep(1000);
    }
}
}
```

```
var cancellationSource = new CancellationTokenSource();
var task = Task.Run(() => Work(cancellationSource.Token), cancellationSource.Token)
    .ContinueWith(t => HandleResult(t));
```


Оживляем UI



```
var task = new Task<int>(() => DoCalculation()); ;
task.ContinueWith(t =>
    ShowResult(t.Result),
    TaskScheduler.FromCurrentSynchronizationContext());
task.Start();
ShowResult(t.Result);
});
task.Start();
```

Warning: Cross-thread operation not valid: Control 'result'

Если для этого исключения имеется обработчик, выполнение программы

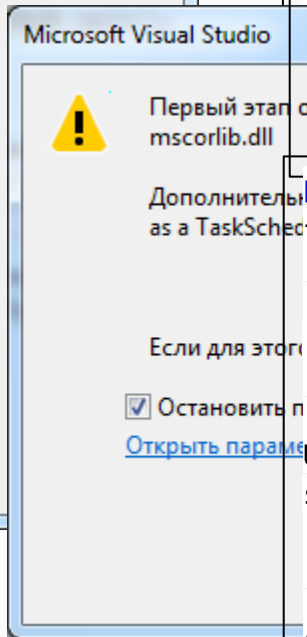
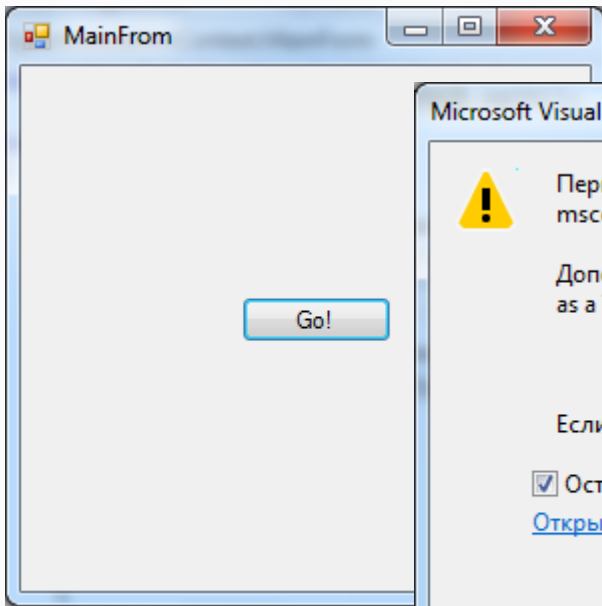
Остановить при возникновении исключения этого типа

[Открыть параметры исключений](#)

Прервать Продолжить Пропустить



Оживляем UI



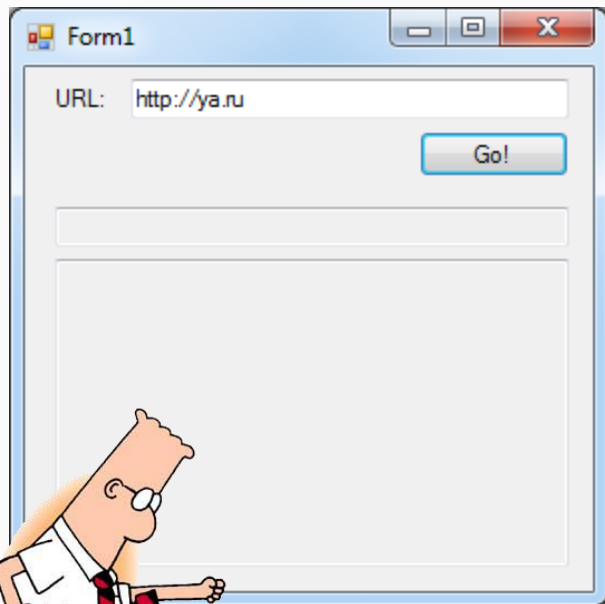
```
var context = TaskScheduler.FromCurrentSynchronizationContext();
Task.Factory.StartNew(() => Calc(context));
.ContinueWith((t) => calc.Text = "Go!", context);

private void Calc(TaskScheduler scheduler)
{
    for (int i = 0; i < 10; i++)
    {
        Thread.Sleep(1000);

        var updateUiTask = Task.Factory.StartNew(() =>
            Update(i), CancellationToken.None, TaskCreationOptions.None,
            scheduler);

        updateUiTask.Wait();
    }
}
```

Плюшки



```
var syncContext = ta(string url)_Click(object sender,
TaskScheduler.FromCurrentSynchronizationContext());
status.Text = "Download data..."; ) =>
var t1 = GetData(url.Text);
var t2 = t1.ContinueWith(t => ent().DownloadString(url);
{ return result;
    status.Text = "Processing data..."; };
return t.Result; it GetData(url.Text);
}, syncContext); t = "Processing data...";
var t3 = t2.ContinueWith(t => data) t DataHandler(data);
{
    status.Text = "Done";
    var t4 = DataHandler(t.Result);
    var t5 = t4.ContinueWith(p => ta;
        { .Sleep(5000);
            status.Text = "Done";
            result.Text = p.Result;
        }, syncContext); );
});;
};;
```

Что почитать?

- [https://msdn.microsoft.com/en-us/library/dd537609\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/dd537609(v=vs.110).aspx)
- [https://msdn.microsoft.com/ru-ru/library/dd537607\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/dd537607(v=vs.110).aspx)
- [https://msdn.microsoft.com/ru-ru/library/dd997402\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/dd997402(v=vs.110).aspx)
- [https://msdn.microsoft.com/ru-ru/library/dd997394\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/dd997394(v=vs.110).aspx)
- <https://msdn.microsoft.com/ru-ru/library/hh191443.aspx>
- <http://habrahabr.ru/post/168669/>
- <http://habrahabr.ru/post/139734/>
- <http://habrahabr.ru/post/162353/>

