Microsoft Partner
Silver Learning

C# Professional

Async & Await

ITVDN
IT VIDEO DEVELOPERS NETWORK

Информационный видеосервис для разработчиков программного обеспечения

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Async & Await

C# 5.0 – What's new?

C# 2 = Generics, Lambda

C# 3 = var, LINQ

C# 4 = dynamic, TPL

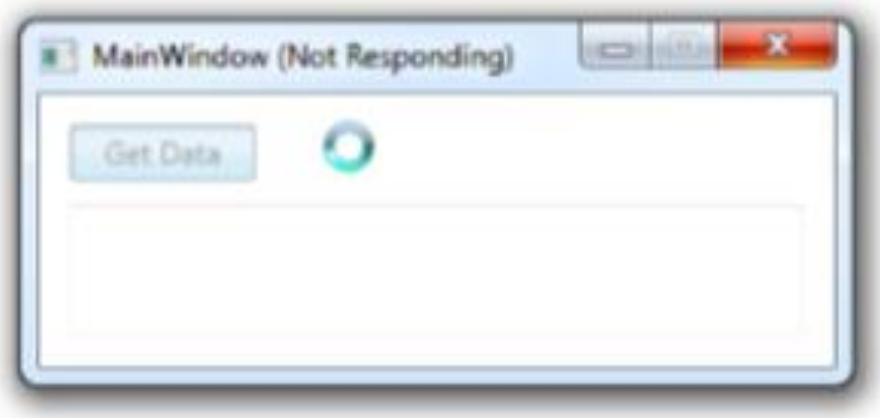
C# 5 = Async

Синхронность: проблемы

```
private void GetButtonClick(object sender, RoutedEventArgs e)
{
        var req= (HttpWebRequest) WebRequest.Create("http://microsoft.com/");

        ……..

        var resp= (HttpWebResponse) req.GetResponse();

        dataTextBox.Text+= resp.Headers.ToString();
}
```

Синхронность: проблемы

Вызывающий поток блокируется, до завершения длительной операции

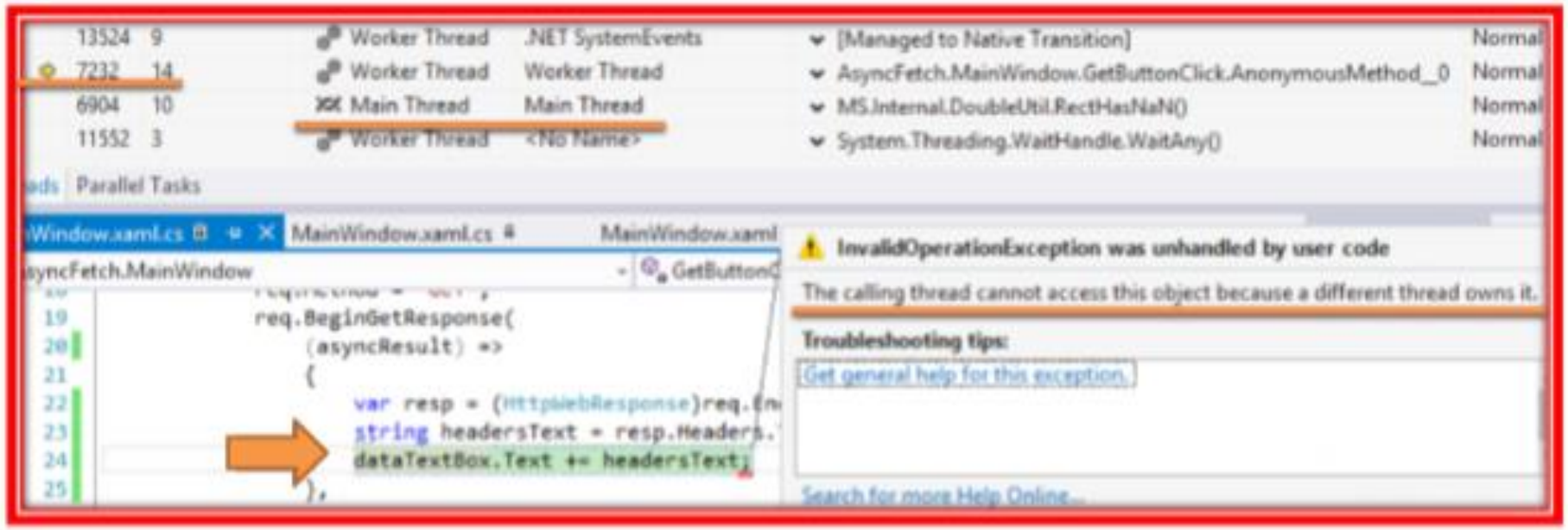


Асинхронная модель до C# 5

```
var req = (HttpWebRequest)WebRequest.Create("http://www.google.com");

req.Method = "GET";

req.BeginGetResponse((asyncResult) =>
        {
            var resp = (HttpWebResponse)req.EndGetResponse(asyncResult);

            string headersText = resp.Headers.ToString();

            dataTextBox.Text += headersText;
        },
    null);
```

Асинхронная модель до C# 5

Попытка обновления UI из другого потока



Асинхронная модель: обновление UI

Обновление UI из другого потока через SynchronizationContext

```
var sync = SynchronizationContext.Current;

req.BeginGetResponse(asyncResult=>
        {
            var resp = (HttpWebResponse)req.EndGetResponse(asyncResult);

            sync.Post(delegate
                    {
                        // ОБНОВЛЕНИЕ UI
                    },
                null);
        },
    null);
```

Асинхронная простота

Ключевое слово async указывает компилятору, что метод является асинхронным.

```
async void getButton_Click(object sender, RoutedEventArgs e)
{
    var w = new WebClient();

    string txt = await w.DownloadStringTaskAsync("...");

    dataTextBox.Text = txt;
}
```

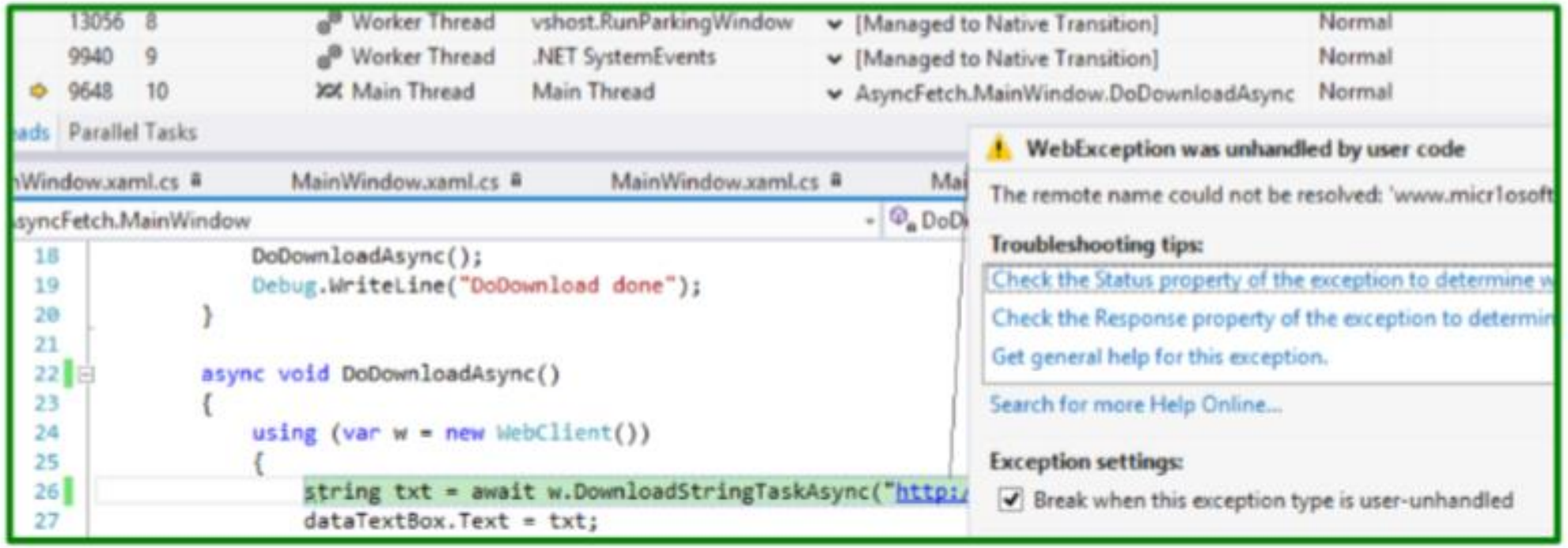
await указывает компилятору, что в этой точке необходимо дождаться окончания асинхронной операции (при этом управление возвращается вызвавшему методу).

Асинхронная простота

```
async void DoDownloadAsync()
{
    using (var w = new WebClient())
    {
        string txt = await w.DownloadStringTaskAsync("http://www.microsoft.com/");
        dataTextBox.Text = txt;
    }
}

void DoDownload()
{
    using (var w = new WebClient())
    {
        string txt = w.DownloadString("http://www.microsoft.com/");
        dataTextBox.Text = txt;
    }
}
```

Исключения

Удобная обработка исключений



Исключения «выбрасываются» в месте вызова асинхронной операции, а не Callback-метода!

Асинхронная и Многопоточность

Task.WhenAll() , Task.WhenAny()

```
var wc1 = new WebClient();
var wc2 = new WebClient();

Task<string> task1 = wc1.DownloadStringTaskAsync(url1);
Task<string> task2 = wc2.DownloadStringTaskAsync(url2);

string[] results = await Task.WhenAll(task1, task2);
```

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association