



The C# Programming Language

What's new in C# 5.0 ?

Lesson 1


Обзор ключевых возможностей

Добрый день!

Семинар «What's new in C# 5.0 ?»

- Инструктор: Олег Кулыгин

www.linkedin.com/in/kulygin




Олег Кулыгин


Тренер-консультант, эксперт по построению сервис-ориентированных и облачных приложений.


Сертифицированный специалист Microsoft (MCTS, MCPD, MCT).


Основные направления:

• C#	• Azure	• UML
• ASP .NET MVC	• Entity Framework	• Patterns of Design
• WCF	• ADO.NET	• ООД









НОВОВВЕДЕНИЯ

C# 5.0 - What's new?

C# 2 = Generics, Lambda

C# 3 = var, LINQ

C# 4 = dynamic, TPL

C# 5 = Async

Синхронность: проблемы

C# 5.0 - What's new?

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

    .....

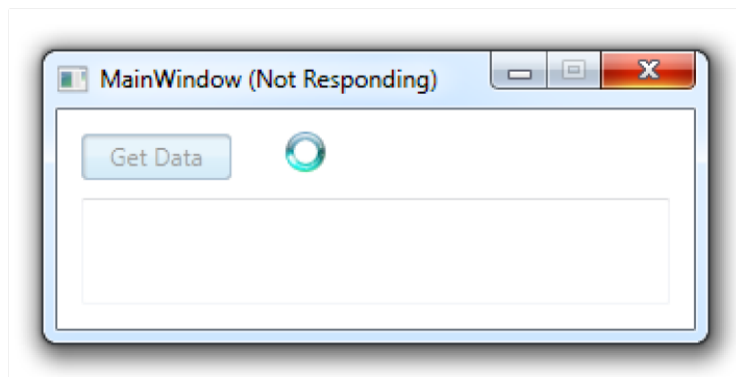
    ➡ var resp = (HttpWebResponse) req.GetResponse();

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

Синхронность: проблемы

C# 5.0 - What's new?


**Вызывающий поток блокируется,
до завершения длительной операции**



Асинхронная модель до C# 5

C# 5.0 - What's new?

```
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: Обновление UI

C# 5.0 - What's new?

Попытка обновления UI из другого потока

The screenshot displays a Visual Studio interface with a thread dump at the top and a code editor below. The thread dump lists several threads, with the 'Main Thread' highlighted in red. The code editor shows the 'GetButtonClick' method in 'MainWindow.xaml.cs', where an asynchronous request is being processed. A red arrow points to the line `dataTextBox.Text += headersText;`, which is highlighted in green. To the right, an exception message is displayed: 'InvalidOperationException was unhandled by user code'. The message states: 'The calling thread cannot access this object because a different thread owns it.' Below the message, there are 'Troubleshooting tips' and a link to 'Get general help for this exception.'.

Id	Priority	Thread Name	Source	Exception	State
13524	9	Worker Thread	.NET SystemEvents	[Managed to Native Transition]	Normal
7232	14	Worker Thread	Worker Thread	AsyncFetch.MainWindow.GetButtonClick.AnonymousMethod_0	Normal
6904	10	Main Thread	Main Thread	MS.Internal.DoubleUtil.RectHasNaN()	Normal
11552	3	Worker Thread	<No Name>	System.Threading.WaitHandle.WaitAny()	Normal

```
18 req.Method = GET;
19 req.BeginGetResponse(
20     (asyncResult) =>
21     {
22         var resp = (HttpWebResponse)req.EndGetResponse(this);
23         string headersText = resp.Headers.ToString();
24         dataTextBox.Text += headersText;
25     },
```

InvalidOperationException was unhandled by user code

The calling thread cannot access this object because a different thread owns it.

Troubleshooting tips:

[Get general help for this exception.](#)

[Search for more Help Online...](#)

Асинхронная модель: Обновление UI

C# 5.0 - What's new?

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); });
```


Async & Await

C# 5.0 - What's new?

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).

Async & Await: Асинхронная простота

C# 5.0 - What's new?

```
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;
    }
}
```

Async & Await: Исключения

C# 5.0 - What's new?

Удобная обработка исключений

The screenshot displays the Visual Studio IDE with a C# project. The top pane shows the 'Parallel Tasks' window with three threads: two Worker Threads and one Main Thread. The bottom pane shows the code for 'AsyncFatch.MainWindow'. The 'DoDownloadAsync' method is highlighted, showing an await statement for 'w.DownloadStringTaskAsync'. A right-click context menu is open over the await statement, displaying a warning: 'WebException was unhandled by user code'. The menu also shows the message 'The remote name could not be resolved: 'www.micr1osoft''. Troubleshooting tips include checking the Status property, checking the Response property, and getting general help. Exception settings are shown at the bottom, with a checked box for 'Break when this exception type is user-unhandled'.

Thread ID	Thread Name	Thread Type	Thread State	Thread Priority
13056	8	Worker Thread	vshost.RunParkingWindow	Normal
9940	9	Worker Thread	.NET SystemEvents	Normal
9648	10	Main Thread	Main Thread	Normal

```
18 DoDownloadAsync();
19 Debug.WriteLine("DoDownload done");
20 }
21
22 async void DoDownloadAsync()
23 {
24     using (var w = new WebClient())
25     {
26         string txt = await w.DownloadStringTaskAsync("http://www.micr1osoft.com/");
27         dataTextBox.Text = txt;
28     }
29 }
```

WebException was unhandled by user code

The remote name could not be resolved: 'www.micr1osoft'

Troubleshooting tips:

- [Check the Status property of the exception to determine w](#)
- [Check the Response property of the exception to determin](#)
- [Get general help for this exception.](#)

[Search for more Help Online...](#)

Exception settings:

- ☒ Break when this exception type is user-unhandled

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

Асинхронность и Многопоточность

C# 5.0 - What's new?

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);
```

Caller Information

C# 5.0 - What's new?

Появилась возможность, получить информацию о методе, который произвел вызов.

```
using System.Runtime.CompilerServices;
```

```
static void TraceMessage(string message,  
    [CallerMemberName] string memberName = "",  
    [CallerFilePath] string sourceFilePath = "",  
    [CallerLineNumber] int sourceLineNumber = 0)  
{  
    Console.WriteLine("message: " + message);  
    Console.WriteLine("member name: " + memberName);  
    Console.WriteLine("source file path: " + sourceFilePath);  
    Console.WriteLine("source line number: " + sourceLineNumber);  
    Console.WriteLine(new string('-',25));  
}
```

Caller Information

C# 5.0 - What's new?

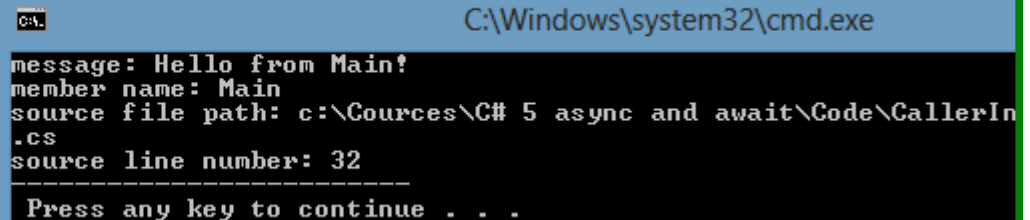
Возможность, получить информацию о методе, который произвел вызов.

using System.Runtime.CompilerServices;

```
static void TraceMessage(string message,
    [CallerMemberName] string memberName = "",
    [CallerFilePath] string sourceFilePath = "",
    [CallerLineNumber] int sourceLineNumber = 0)
{
    Console.WriteLine("message: " + message);
    Console.WriteLine("member name: " + memberName);
    Console.WriteLine("source file path: " + sourceFilePath);
    Console.WriteLine("source line number: " + sourceLineNumber);
    Console.WriteLine(new string('-',25));
}

static void Main()
{
    TraceMessage("Hello from Main!");

    Console.ReadKey();
}
```



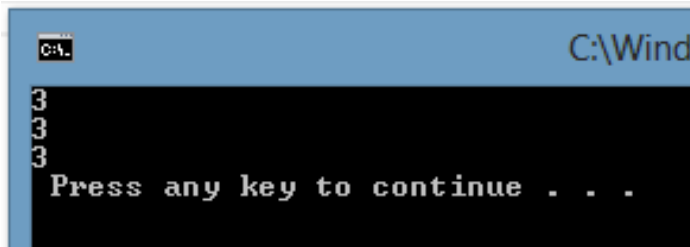
```
C:\Windows\system32\cmd.exe
message: Hello from Main!
member name: Main
source file path: c:\Sources\C# 5 async and await\Code\CallerIn
.cs
source line number: 32
-----
Press any key to continue . . .
```

Замыкания на переменных цикла

C# 5.0 - What's new?

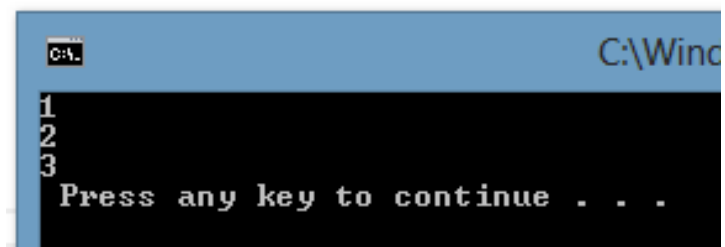
```
var actions = new List<Action>();  
  
foreach (var i in Enumerable.Range(1, 3))  
    actions.Add(() => Console.WriteLine(i));  
  
foreach (var action in actions) action.Invoke();
```

C# 4.0



A screenshot of a Windows command prompt window titled 'C:\Wind'. The output shows three lines of the number '3' stacked vertically, followed by the text 'Press any key to continue . . .'. This demonstrates a closure bug in C# 4.0 where the loop variable 'i' is not captured by value, leading to all iterations printing the final value of 'i' (which is 3).

C# 5.0



A screenshot of a Windows command prompt window titled 'C:\Wind'. The output shows three lines of the numbers '1', '2', and '3' stacked vertically, followed by the text 'Press any key to continue . . .'. This demonstrates the correct behavior in C# 5.0 where the loop variable 'i' is captured by value, allowing each iteration to print its respective value.

Замыкания на переменных цикла

C# 5.0 - What's new?

Closure

Int32 i = 3

```
class Closure { public int i; }
```

```
var actions = new List<Action>();

using (var enumerator = Enumerable.Range(1, 3).GetEnumerator())
{
    var closure = new Closure();
    while (enumerator.MoveNext())
    {
        closure.i = enumerator.Current;
        var action = new Action(()=>Console.WriteLine(closure.i));
        actions.Add(action);
    }
}

foreach (var action in actions) action();
```

C# 5

Замыкания на переменных цикла

C# 5.0 - What's new?

Closure

Int32 i = 3

```
class Closure { public int i; }
```

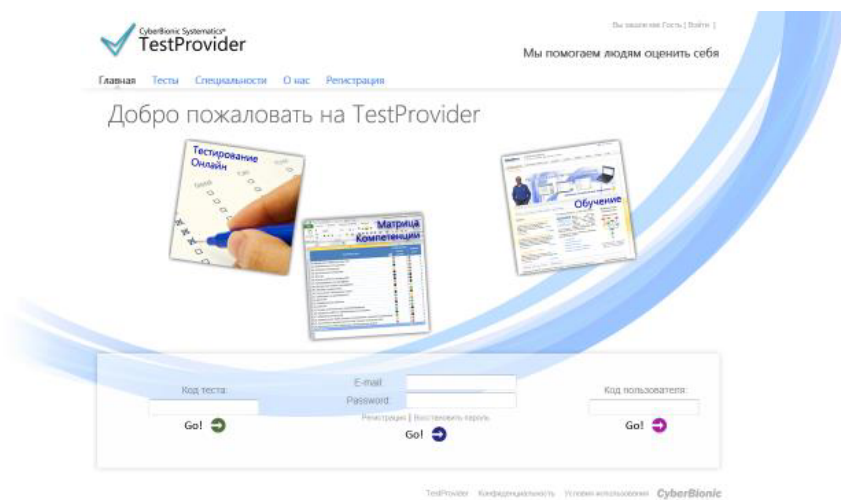
```
var actions = new List<Action>();

using (var enumerator = Enumerable.Range(1, 3).GetEnumerator())
{
    var closure = new Closure();
    while (enumerator.MoveNext())
    {
        closure.i = enumerator.Current;
        var action = new Action(()=>Console.WriteLine(closure.i));
        actions.Add(action);
    }
}

foreach (var action in actions) action();
```

C# 5

Q&A



Перейти к тестированию
www.TestProvider.com

Тестирование IT-специалистов

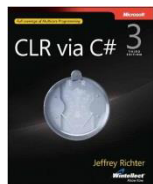
TestProvider обеспечивает надежную и объективную оценку технических знаний и опыта работы IT-специалиста с программными продуктами Microsoft.

Сертификация IT-специалистов

Подтвердите ваш практический опыт работы с технологиями **Microsoft**, получив сертификацию, соответствующую той работе, которую вы выполняете сейчас или желаете получить в будущем.

Компании **Microsoft**, **CyberBionic Systematics** и **Intel** на базе портала TestProvider компании CyberBionic Systematics с использованием платформы Microsoft Azure совместно с Министерством науки и образования Украины проводят Всеукраинское дистанционное мониторинговое исследование уровня сформированности у выпускников учебных заведений навыков использования информационно-коммуникативных технологий в практической деятельности.

Курс рекомендован
Джеффри Рихтером



Дорогие студенты школы CyberBionic Systematics,

Курс “C# для профессионалов” включает темы, которые излагаются в моей книге «CLR via C#» и учит, как работает CLR, показывая вам как строить приложения и повторно используемые компоненты для .NET Framework. Этот курс не предназначен для начинающих программистов; вам необходимо базовое понимание Объектно-Ориентированного Программирования и опыт работы с языком C# (а не только элементарное понимание синтаксиса). Я считаю, что вы найдете много информации о свойствах CLR и о том как использовать современные методы программирования в школе CyberBionic Systematics.

Желаю вам всего самого наилучшего в вашей карьере,

Джеффри Рихтер, автор книги «CLR via C#»

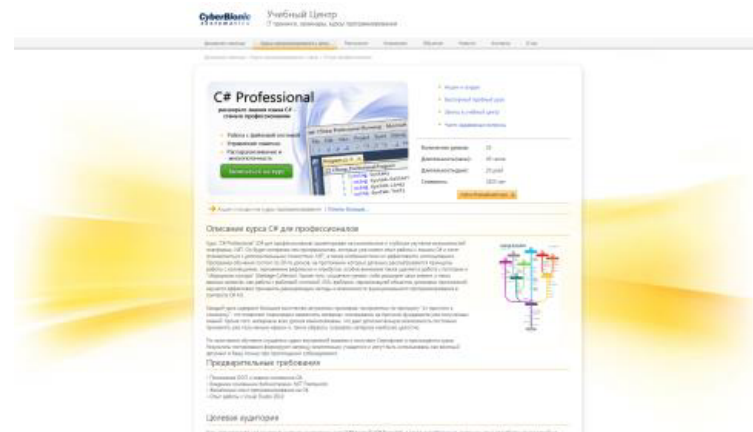
Описание курса:

Курс дает возможность глубже освоить возможности языка C# и изучить основные библиотеки платформы .NET Framework. Курс поможет освоить возможности языка, которые используются при профессиональном проектировании и разработке программного обеспечения. Курс будет полезен программистам с опытом программирования на C#.

Длительность:

40 часов/20 дней.

Узнать более подробно о курсе на сайте:



www.edu.cbsystematics.com

CyberBionic **s y s t e m a t i c s**

Coevolution of humans and machines.