

С# Асинхронное программирование

Асинхронный шаблон программирования
Task-based asynchronous pattern

C# Асинхронное программирование

Автор курса



Гнатюк Владислав



MCID:16354168

Асинхронное программирование

После урока обязательно



Повторите этот урок в видео формате на [ITVDN.com](http://itvdn.com)



Проверьте как Вы усвоили данный материал на [TestProvider.com](http://testprovider.com)

Асинхронный шаблон программирования Task-based Asynchronous Pattern

C# Асинхронное программирование

План урока

- 1) Библиотека TPL (Task Parallel Library)
- 2) Шаблон TAP (Task-based Asynchronous Pattern)
- 3) Задача (Task)
- 4) Фабрика задач (Task Factory)
- 5) Продолжения задач (Task Continuations)
- 6) ValueTask/ValueTask<TResult>
- 7) Task vs ValueTask

C# Асинхронное программирование

Асинхронный шаблон TAP

```
{ } System.Threading.Tasks
  ▷ ConcurrentExclusiveSchedulerPair
  ▷ Parallel
  ▷ ParallelLoopResult
  ▷ ParallelLoopState
  ▷ ParallelOptions
  ▷ Task
  ▷ Task<TResult>
  ▷ TaskCanceledException
  ▷ TaskCompletionSource<TResult>
  ▷ TaskContinuationOptions
  ▷ TaskCreationOptions
  ▷ TaskFactory
  ▷ TaskFactory<TResult>
  ▷ TaskScheduler
  ▷ TaskSchedulerException
  ▷ TaskStatus
  ▷ UnobservedTaskExceptionEventArgs
```

Шаблон TAP (Task-based Asynchronous Pattern) – основанный на библиотеке TPL – Task Parallel Library.

Для работы с шаблоном TAP используется пространство имен System.Threading.Tasks

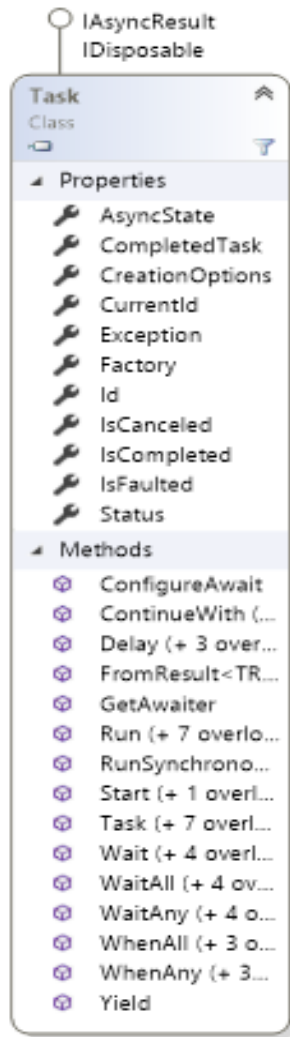
C# Асинхронное программирование

Преимущество TAP

- 1) Простая инициализация и завершение асинхронной операции.
- 2) Удобный способ получения возвращаемого значения асинхронной операции.
- 3) Получение исключения, возникшего в асинхронной операции для его обработки.
- 4) Просмотр состояния асинхронной операции.
- 5) Поддержка отмены выполнения (Необязательно).
- 6) Продолжения задач (Task Continuations)
- 7) Планирование выполнения асинхронной операции

C# Асинхронное программирование

Задача - Task



Task (задача) – конструкция, которая реализует модель параллельной обработки, основанной на обещаниях (Promise). Задача «обещает», что работа будет выполнена позже, позволяя взаимодействовать с помощью обещания с чистым API.

Для работы с задачами в .NET используют класс Task.

C# Асинхронное программирование

Способы создания задач

- Создания экземпляра класс Task и вызов на нем метод Start().

```
Task task = new Task(new Action(Download));  
task.Start();
```

- С помощью фабрики задач и метода StartNew().

```
TaskFactory taskFactory = new TaskFactory();  
taskFactory.StartNew(new Action(Download));
```

- Вызов статического метода Task.Run().

```
Task.Run(new Action(Download));
```

- Вызов метода RunSynchronously() для выполнения в основном потоке.

```
Task task = new Task(new Action(Download));  
task.RunSynchronously();
```

C# Асинхронное программирование

Продолжения - Continuations

Продолжения – это асинхронная задача, вызываемая другой задачей при своем завершении. Это некий вариант метода обратного вызова (Callback method).

Для добавления продолжения используют метод `ContinueWith()`

```
Task task = new Task(new Action(Download));  
Task continuation = task.ContinueWith(new Action<Task>(ShowData));
```

Метод `ContinueWith()` возвращает новый экземпляр класса `Task`, что позволит выстраивать цепочки продолжений.

C# Асинхронное программирование

Фабрика задач

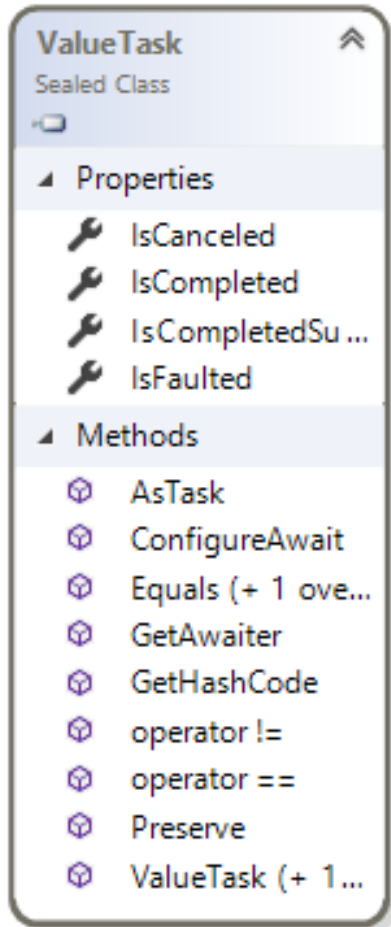
Фабрика задач — механизм, позволяющий настроить набор сгруппированных задач, которые находятся в одном состоянии.

Классы для работы с фабрикой `TaskFactory` и `TaskFactory<TResult>`.



C# Асинхронное программирование

Значимая задача - ValueTask



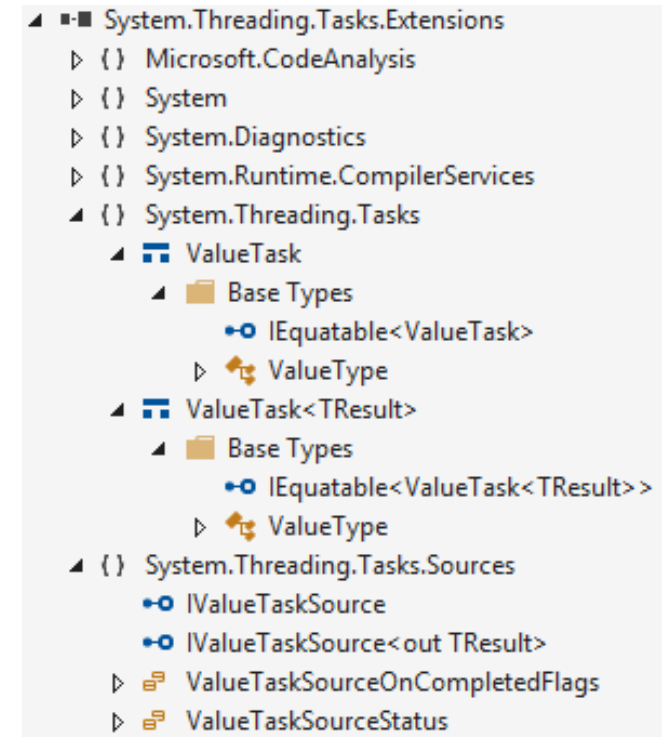
ValueTask – представляет собой обертку (wrapper) над обыкновенной задачей (Task). Был создан для уменьшения потребления ресурсов управляемой кучи.

В некоторых случаях является сомнительной оптимизацией.

С# Асинхронное программирование

ValueTask в сборке System.Threading.Tasks.Extensions

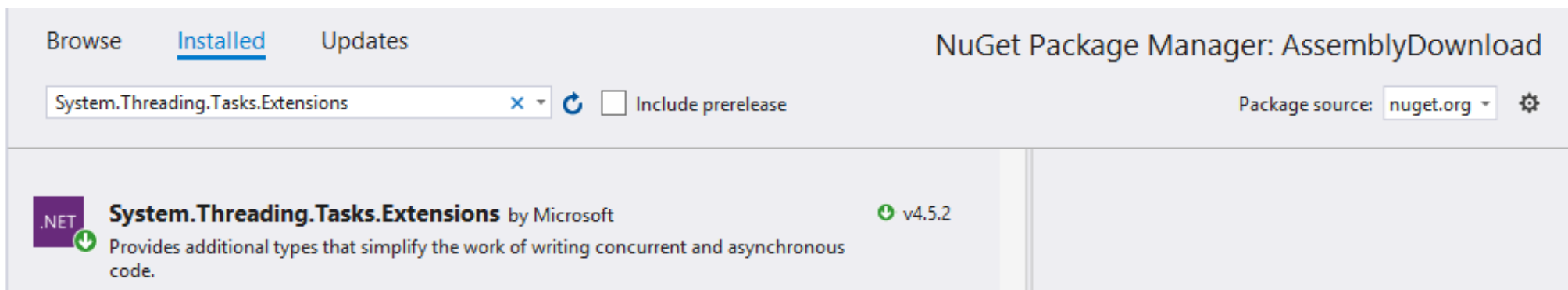
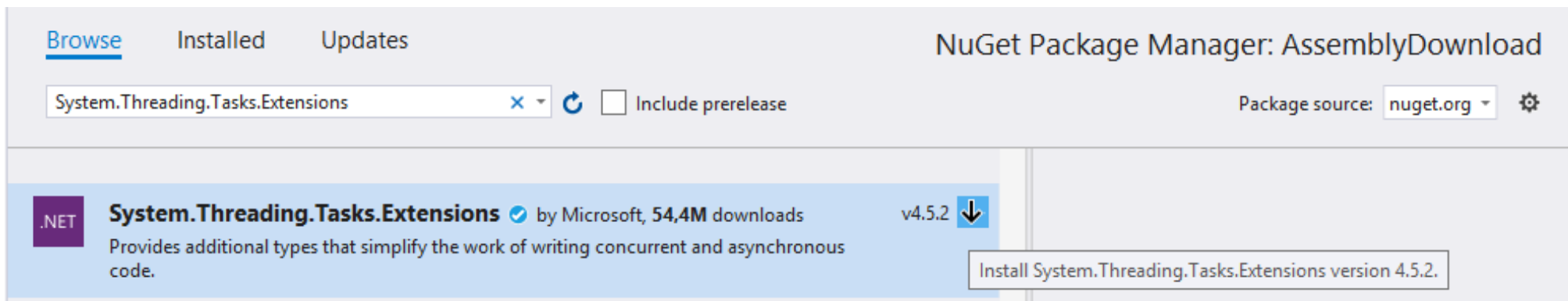
`System.Threading.Tasks.Extensions` – библиотека для расширения работы TPL. В ней находятся классы для работы со значимыми задачами.



C# Асинхронное программирование

Загрузка сборки System.Threading.Tasks.Extensions – 1 способ

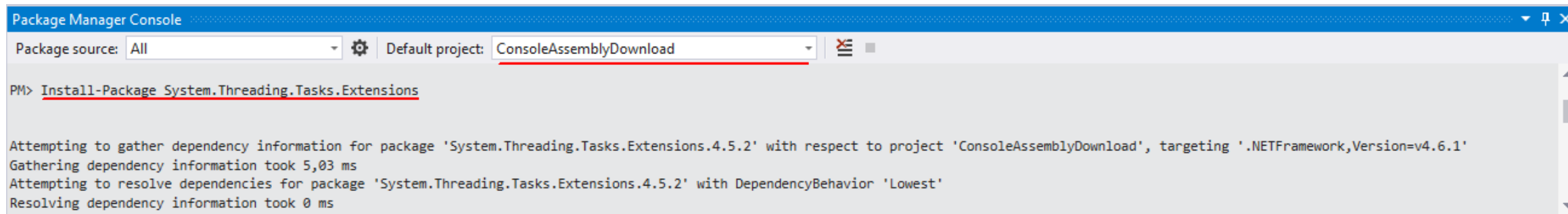
- 1) Открыть менеджер управления пакетами NuGet.
- 2) В поиске написать «System.Threading.Tasks.Extensions».
- 3) Загрузить сборку. В вкладке «Установленные» у вас отобразится наличие сборки.



C# Асинхронное программирование

Загрузка сборки System.Threading.Tasks.Extensions – 2 способ

- 1) Открыть вкладку «Вид», открыть «Другие окна», выбрать «Консоль менеджера пакетов»
(View -> Other Windows -> Package Manager Console)
- 2) Выбрать проект для установки сборки.
- 3) В консоли написать строку : «Install-Package **имя_сборки**».
 - Команда с указанием проекта установки : «Install-Package **имя_сборки** –Project **имя_проекта**»



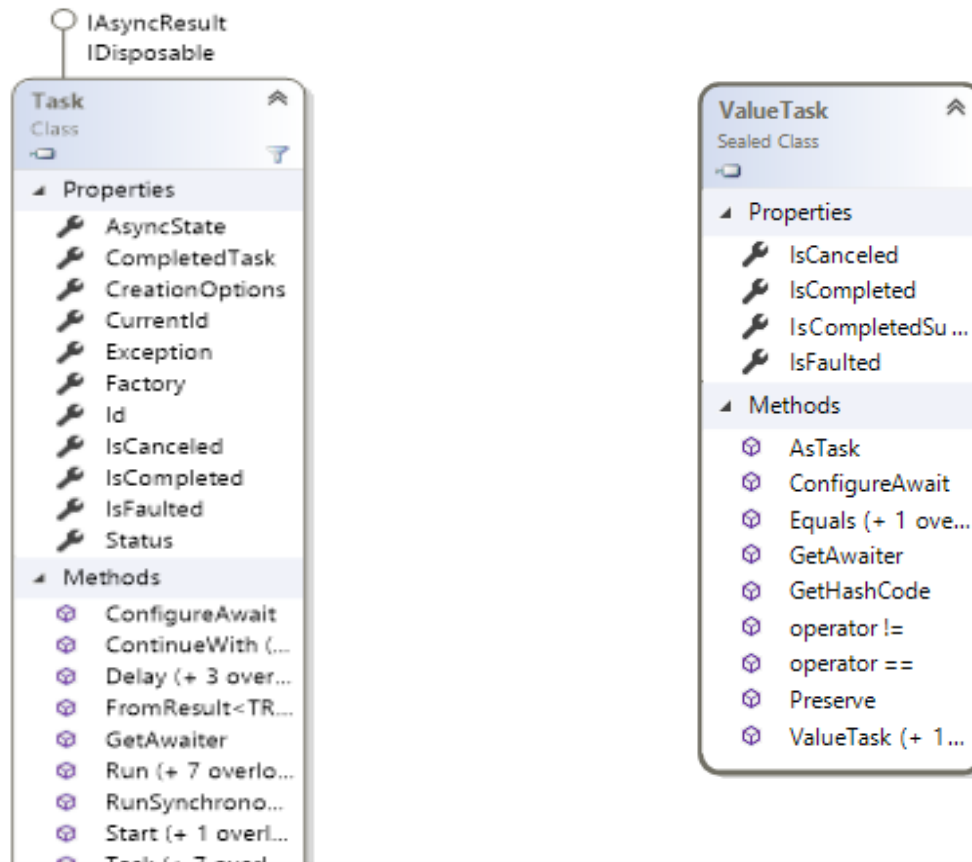
```
Package Manager Console
Package source: All [v] [g] Default project: ConsoleAssemblyDownload [v] [x] [■]
PM> Install-Package System.Threading.Tasks.Extensions

Attempting to gather dependency information for package 'System.Threading.Tasks.Extensions.4.5.2' with respect to project 'ConsoleAssemblyDownload', targeting '.NETFramework,Version=v4.6.1'
Gathering dependency information took 5,03 ms
Attempting to resolve dependencies for package 'System.Threading.Tasks.Extensions.4.5.2' with DependencyBehavior 'Lowest'
Resolving dependency information took 0 ms
```

C# Асинхронное программирование

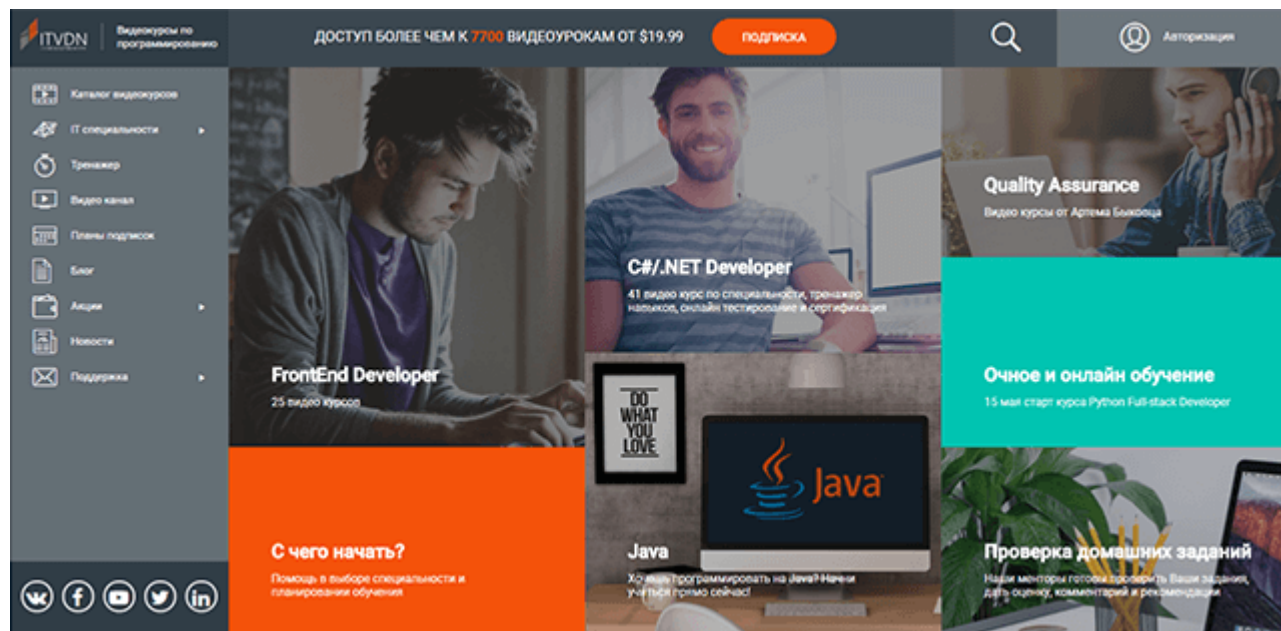
Task vs ValueTask

Что же лучше использовать Task (Task<TResult>) или ValueTask (ValueTask<TResult>) ?



Смотрите наши уроки в видео формате

ITVDN.com



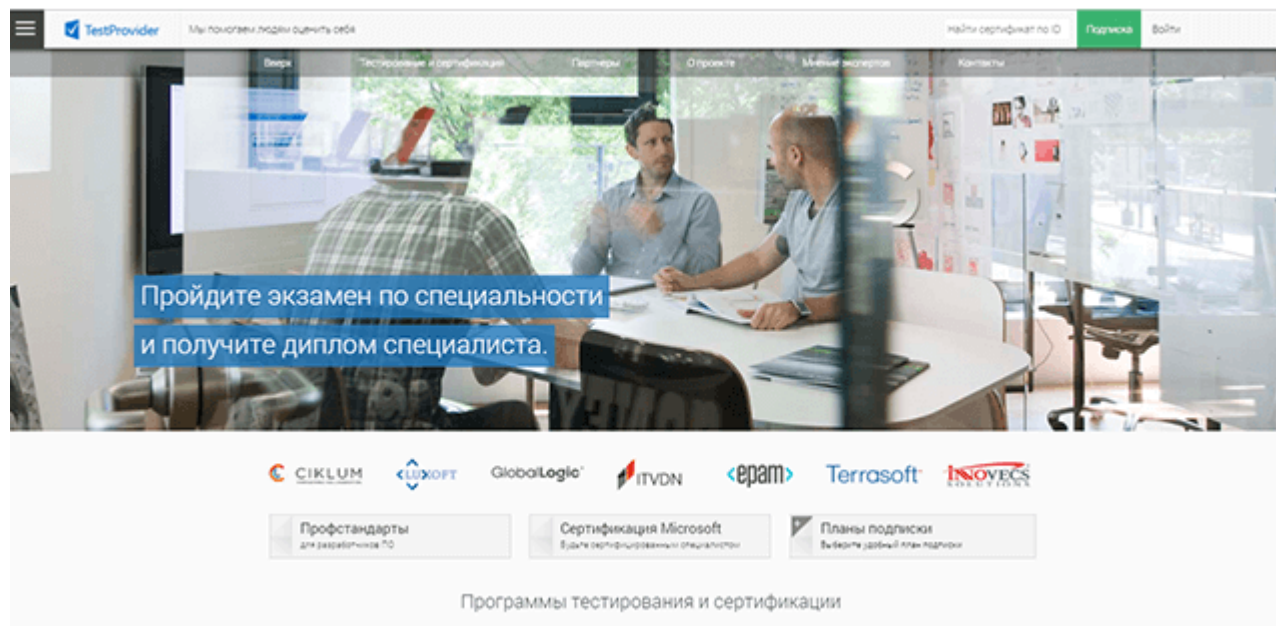
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.



Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на [TestProvider.com](https://testprovider.com)

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.



C# Асинхронное программирование

Q&A

Информационный видеосервис для разработчиков программного обеспечения

