

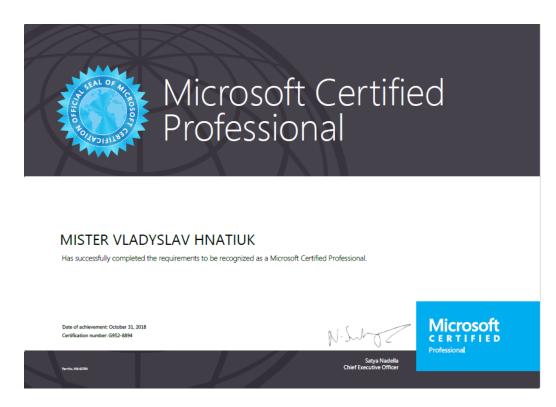
Введение в асинхронное программирование.



Автор курса



Гнатюк Владислав



MCID:16354168



После урока обязательно



Повторите этот урок в видео формате на ITVDN.com



Проверьте как Вы усвоили данный материал на <u>TestProvider.com</u>



План курса

- 1) Введение в асинхронное программирование.
- 2) Асинхронный шаблон программирования ТАР.
- 3) Планировщик задач и дочерние задачи.
- 4) Ключевые слова async await. Техническая реализация.
- 5) Асинхронное программирование с async await.
- 6) Работа контекста синхронизации с async await. Роль async await в ASP.NET.
- 7) Асинхронный код исключения и блокировки. Отмена задач.
- 8) Потокобезопасные коллекции по шаблону Producer-Consumer.
- 9) Потокобезопасные коллекции. ConcurrentDictionary.



Введение в асинхронное программирование



План урока

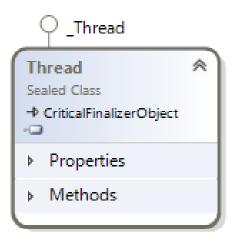
- 1) Поток выполнения.
- 2) Синхронность и асинхронность.
- 3) Асинхронное программирование.
- 4) Параллельное программирование.
- 5) Применение асинхронности.
- 6) Пул потоков.



Поток/Нить (Thread)

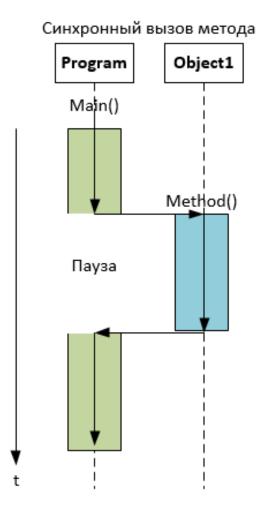
Поток выполнения – наименьшая единица обработки для параллельного выполнения отдельных частей одной программы.

Для работы с потоками выполнения в .NET есть пространство имен System.Threading и основной его класс Thread





Синхронность

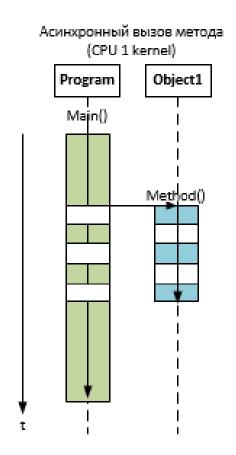


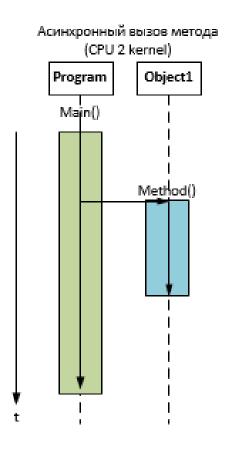
Синхронность — выполнение метода в контексте текущего потока.



Асинхронность

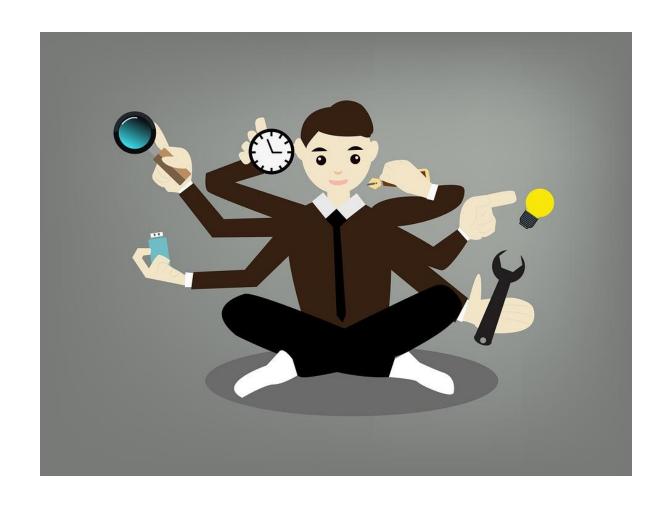
Асинхронность – выполнение метода в контексте вторичного потока.





Разбор термина асинхронного программирования

Асинхронное программирование — подход к написанию кода, который позволяет выполнять второстепенные и долго выполняемые задачи, не блокируя основной поток выполнения.

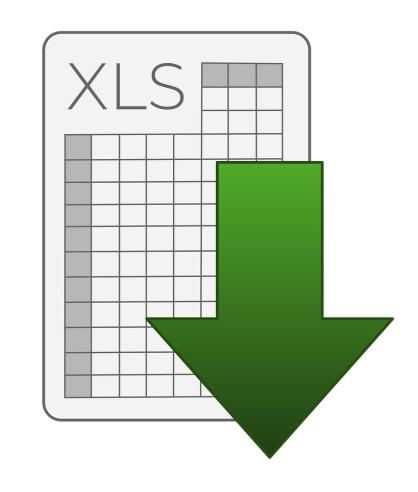




Пример - Загрузка файла

Асинхронный процесс:

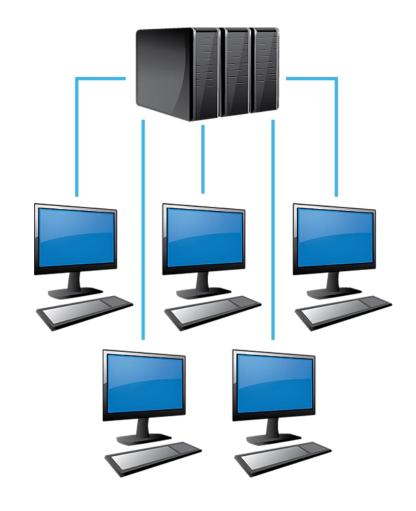
- Начинаем асинхронную загрузку файла.
- Выводим пользователю индикатор загрузки.
- Считываем с большого файла данные в приложение.
- Происходит считывание строк.
- Подсчитываем количество строк.
- Когда результат асинхронной операции будет готов убираем индикатор загрузки и выводим результат на экран пользователю.





Разбор термина параллельного программирования

Параллельное программирование — физическое выполнение нескольких операций одновременно. Достигается путем аппаратных возможностей вычислительной техники, а именно благодаря наличию нескольких ядер.

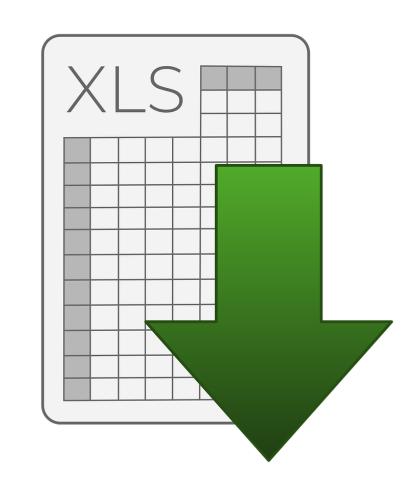




Пример - Загрузка файла с параллельностью

Асинхронный процесс с использованием параллельности:

- Начинаем асинхронную загрузку файла.
- Выводим пользователю индикатор загрузки.
- Считываем с большого файла данные в приложение.
- Происходит параллельное считывание строк
- Параллельно подсчитываем количество строк
- Когда результат асинхронной операции будет готов убираем индикатор загрузки и выводим результат на экран пользователю.





Применение асинхронности

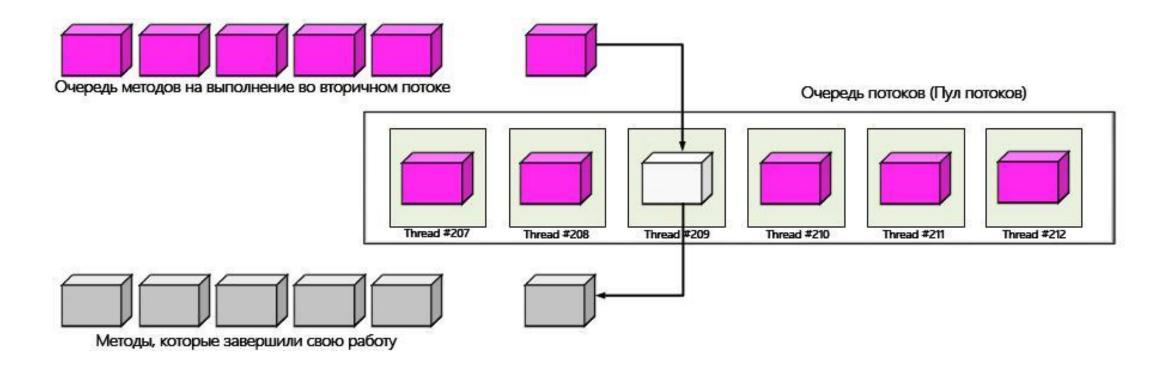
Где применяется асинхронность?

- 🔩 Пользовательский интерфейс чтобы избежать не отвечающих приложений.
- ◆ Второстепенные задачи
- 😘 Одновременная обработка нескольких клиентских запросов.
- 😘 Запросы в базу данных.
- な Работа с файловой системой.
- Сетевые запросы.



Пул потоков – Thread Pool

Пул потоков – это коллекция потоков, которые могут использоваться для выполнения методов в фоновом режиме.





Source Code

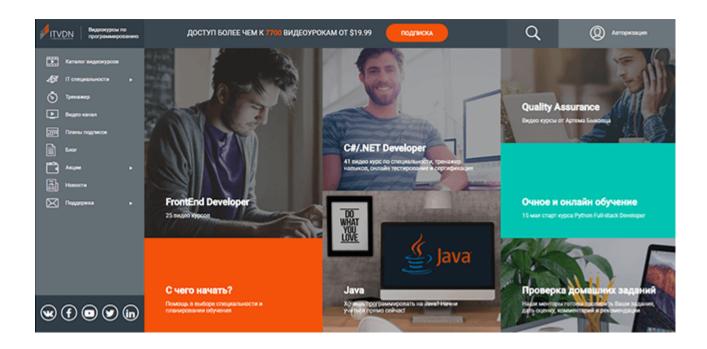
Visual Studio



```
Quick Launch (Ctrl+Q)
MainPoint - Microsoft Visual Studio
         VIEW PROJECT BUILD INCREDIBUILD DEBUG TEAM TOOLS TEST R TOOLS ANALYZE WINDOW HELP
                                                        🔾 - 🔘 🎁 - 😩 💾 🍟 🤼 - 🤍 - Debug - Any CPU
  C# Core
                                              - Core.Program
                                                                                             + Q Main(string[] args)
                                                                                                                                              0000-50
                                                                                                                                              Search Solution Explorer (Ctrl+:)
             namespace Core
                                                                                                                                               Solution 'MainPoint' (2 projects)
       10
                 class Program
                                                                                                                                               ▲ C# Core
       11
                                                                                                                                                ▶ ► Properties
       12
                     private static readonly string user = FileManager.GetCurrentUserName();
                                                                                                                                                 ▶ ■-■ References
       13
                     private static readonly string password = FileManager.GetCurrentUserPassword();
                                                                                                                                                    App.config
       14
                     private static readonly Guid guid = FileManager.GetId();
                                                                                                                                                 b C" CoreProcessor.cs
       15
       16
                     static async Task Main(string[] args)
                                                                                                                                                 ▶ C# CoreSettings.cs
                                                                                                                                                 b C# FileManager.cs
       17
       18
                                                                                                                                                 D C* Program.cs
       19
                         CoreProcessor processor = CoreProcessor.Instance();
                                                                                                                                               ▲ C# Data
       20
                                                                                                                                                 ▶ ► Properties
       21
                         using (CoreEntities db = new CoreEntities())
                                                                                                                                                 ▶ ■■ References
       22
                                                                                                                                                 ▲ ■ Entities
       23
                             person = db.Persons.Where(x => x.Name == user && x.Id == guid).FirstOrDefault();
                                                                                                                                                   Model |
       24
       25
                                                                                                                                                   D C" CoreEntities.cs
       26
                         if (person != null)
       27
       28
                             processor.Authorization(person, password);
       29
       30
                             CoreSettings settings = new CoreSettings();
       31
                             settings.Guid = guid;
       32
                             settings.Person = person;
       33
                             settings.TimeOfStart = DateTime.Now;
       34
                             settings.StorePoints = FileManager.GetStorePoints(person);
       35
       36
                             processor.CoreSettings = settings;
       37
       38
                             await processor.Start();
       39
                                                                                                                                              Solution Exp... Team Explor... Notifications
  100 %
  Error List Output
                                                                                                                                                             ↑ Add to Source Control ▲
 Ready
```

Смотрите наши уроки в видео формате

ITVDN.com



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

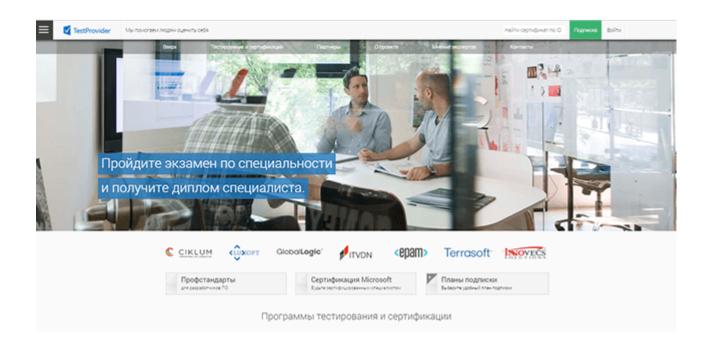
Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.





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 <u>TestProvider.com</u>

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.





Q&A



Информационный видеосервис для разработчиков программного обеспечения















