# Уточнение с наследованием

**{** public class Person  
 **{** public string Name **{** get**;** set**; }** public virtual void VirtualDisplay**()  
 {** Console**.**WriteLine**(**$"Person {Name}"**);  
 }** public void NotVirtualDisplay**()  
 {** Console**.**WriteLine**(**$"Person {Name}."**);  
 }  
 }** class Employee : Person  
 **{** public string Company **{** get**;** set**; }** public void SomeEmployeeMethod**() {}** public override void VirtualDisplay**()  
 {** Console**.**WriteLine**(**$"Employee {Name}"**);  
 }** public void NotVirtualDisplay**()  
 {** Console**.**WriteLine**(**$"Employee {Name}."**);  
 }  
 }** static class Program  
 **{** public static void Local\_Main**()  
 {** Person person = new Employee **{** Name = "Sam"**,** Company = "Microsoft" **};** *// person.SomeEmployeeMethod(); //ERROR я понимаю почему* person**.**VirtualDisplay**();** *//”Employee Sam”* person**.**NotVirtualDisplay**();** *//”Person Sam.”* Console**.**ReadKey**();  
 }  
 }**

Ответ с метанита: На консоль будет выведена строка "Employee Sam". Потому что переменная person хранит ссылку на объект Employee, и при вызове метода Display будет использоваться именно реализация данного метода в классе Employee. *Я это понимаю.*

Как я вижу: сделали объект Employee, сказали что теперь это Person, просто ссылку присвоили, он остался Employee, но с ограничениями Person, то есть методы из Employee которых нет в person он не видит, *SomeEmployeeMethod* тому подтверждение. Теперь делаем методы с одинаковыми именами, виртуальный и обычный. Виртуальный он видит потому он есть в обоих и что он переопределен поэтому и выводит Employee Sam, а не потому что ссылка. А с обычным он видит свой, а на Employee.NotVirtualDisplay стоят ограничения, поэтмоу он и не работает.

Просто если в new Employee ссылка на объект employee откуда он берет метод person**.**NotVirtualDisplay**()**?

# Можно ли сделать несколько своих классов типов Т? Типа это либо этот либо этот? Как тогда сделать такое? Оболочку писать? А если у них тоже наследование от чего то еще

class Account<T> where T: Class1||Class2

# Как переопределить явное/неявное преобразование с обобщенными классами? (Вопрос если можно сделать в вопросе 2.)

Чтоб можно было сделать как в последней строчке, тут понятно что не заработает, просто теоретически как это можно сделать?

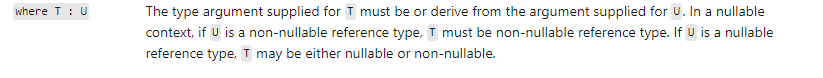
Типо как тут прописать https://metanit.com/sharp/tutorial/3.37.php

class Account<T>   
**{** public static T Session**; // int or string** public int Id **{** get**;** set**; }** public int Sum **{** get**;** set**; }** public void Display**()  
 {** Console**.**WriteLine**(**$"Session: {Session} \nId: {Id} \nSum: {Sum}\n" + new string**(**'-'**,** 10**));  
 }  
}**

main:

Account<int> example = new Account<int>**();**var a = **(**Account<string>**)**example**;**

# А еще я вот это не понял, первое предложение



***The type argument supplied for T must be*** or derive from the argument supplied for U.

-Просто должен быть?

The type argument supplied for T must be ***or derive from the argument supplied for U.***

-Должен быть производным от U?

# (тот же что и 1 вопрос, не смотреть)Почему выводится false? Если сделать virtual+override то будет выводиться true.

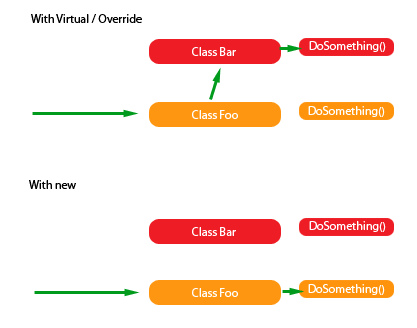
public class Foo  
**{** public bool DoSomething**() {** return false**; }  
}**public class Bar : Foo  
**{** public new bool DoSomething**() {** return true**; }  
}**public class Test  
**{** public static void Main **()  
 {** Foo test = new Bar **();** Console**.**WriteLine **(**test**.**DoSomething **());  
 }  
}**

Я не понимаю как это работает, по логике, в

Foo test = new Bar **();**мы делаем новый Bar, потом его ссылку даем Foo test, и раз ссылка на Bar он не видит метод Bar’a, а свой он не имеет потому что объект создавался Bar’a

***Ответ со стэка***: идем от base to childs:

Мы как бы говорим, когда new, мы не имеем ничего общего с Bar, поэтмоу он и юзает свой метод



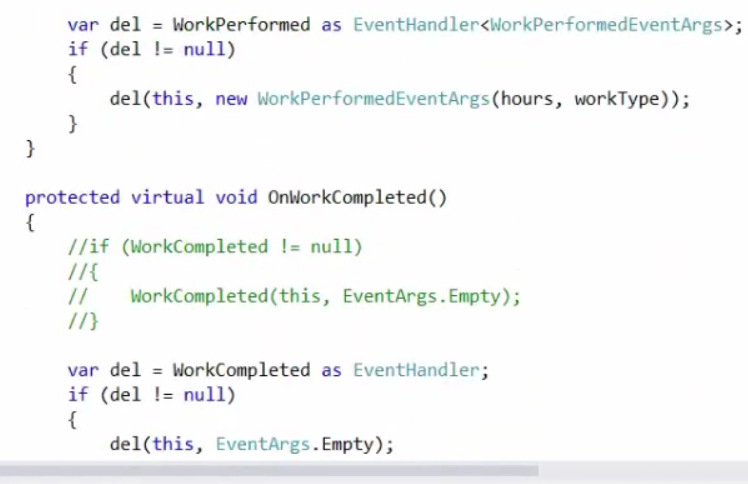
Я все равно не понимаю как это внутри работает

# Почему нестатические методы класса не могут использоваться в статических методах?

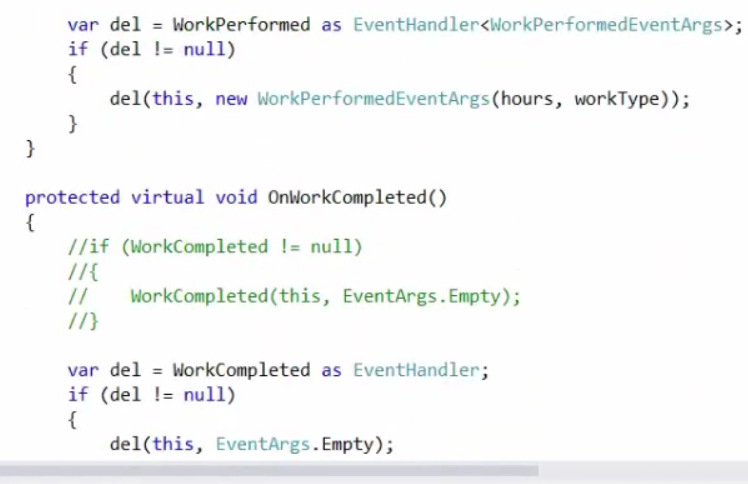
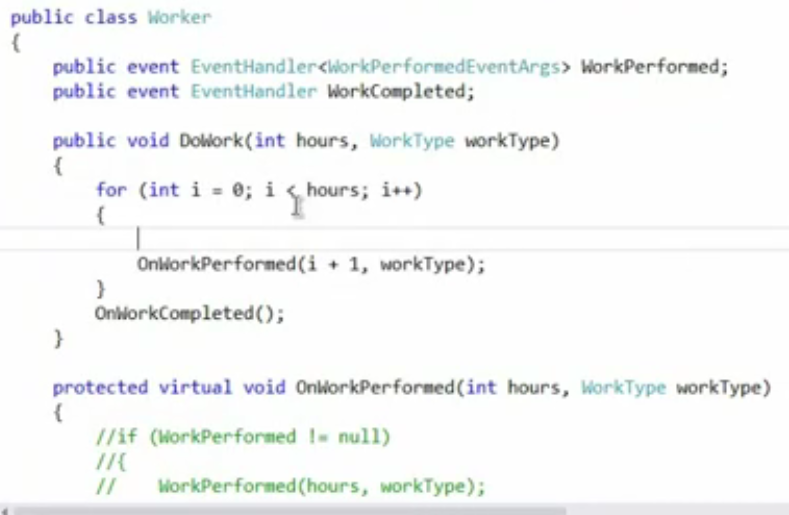
В чем логика? Они же могут быть чистой функцией, ну если нет то ошибку бы высвечивали и намана, свободу ограничивают))))

# Зачем здесь нам приводить event k EventHandler’y если он и так EventHandler<same>

Я заметил так постоянно делают когда эвенты пишут, просто он мог начать указывать на еще чтоото? Но он же все равно будет того же типа



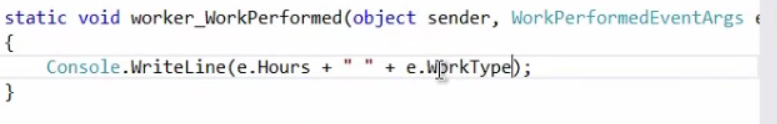
Весь код:

Main:

workerPerfomed)



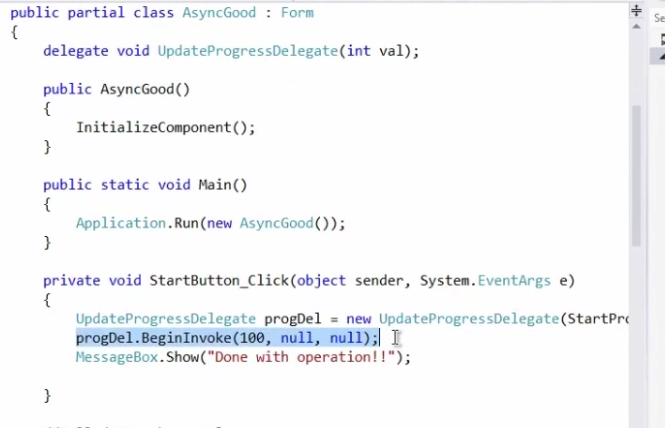


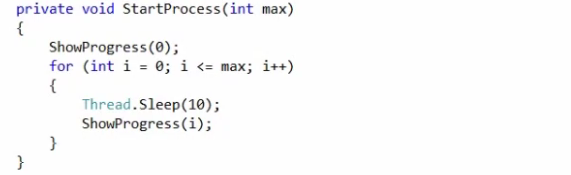
# Из самого EventArgs мы же никак данные не вытащим, он просто чтобы наследоваться от него же и писать EventArgs.Empty?

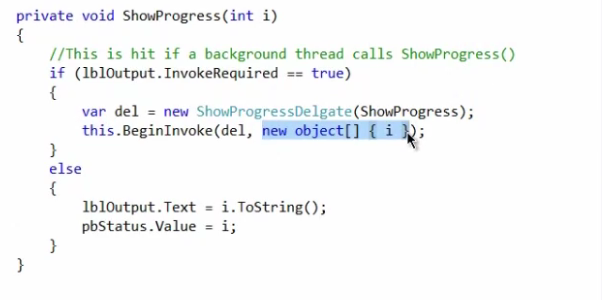
# Async Delegates – good

Будет наверное лучше понятно когда потоки зашарю и лучше оставить на потом?

edited







BeginInvoke(Delegate del, params object[] args) – вызывает делегат в контексте this(FORMS)

-Итак, начинаем, запускаем progdel.BeginInvoke(StartProcess), он стартует параллельно GUI Thread, и мы как бы хотим в будущем передать в ГУИ поток дату(100)

-вызывается StartProcess(100), где вызывается ShowProgress(1-100), это все в потоке progdel.BeginInvoke(StartProcess), приходим в ShowProgress

-InvokeRequired делает проверку, в каком контексте работает текущий код. InvokeRequired = true, если контекст чужого потока (не потока GUI). Получается в else – только поток GUI

Теперь у нас идет поток progdel.BeginInvoke(StartProcess), и мы вызываем ShowProgress в контексте this(Forms):



Только в другом потоке вызываем ShowProgress, и там уже нормально меняем значения

Но если честно я все равно не очень понял, мы типо сначала передали i а потом перевызвали в другом потоке, то есть поток progdel.BeginInvoke(StartProcess) донес в своем потоке i вызвал метод и потом вызвал метод в другом потоке, верно?

# Интерфейсы

Из метода возращается Person[], и все работает, я не очень понимаю что тут написано и какое отношение к интерфейсам имеет не пойму. Ну то есть понимаю что IEnumerable это интерф. Но они же просто описывают поля и методы, какое отношение это здесь имеет?



Как мы тут вообще создаем объект интерфейса, они же как абстрактный класс примерно?

Я попробовал сделать так:

IEnumerable squares = new**[] {** new Square**(**1**,** 1**,** 1**),** new Square**(**2**,** 2**,** 2**),** new Square**(**3**,** 3**,** 3**) };**

И у нас нет доступа к public полям squares, хоть и показывает тип Square[]

Но если сделать

IEnumerable<Square> squares = new**[] {** new Square**(**1**,** 1**,** 1**),** new Square**(**2**,** 2**,** 2**),** new Square**(**3**,** 3**,** 3**) };**

То все окей, ну так вроде понятно интуитивно, но как работает нет

Типо есть объект Square[], мы говорим что теперь IEnumerable<Square> squares указывает на него, поэтому ему добавляются свойства IEnumerable?

То есть когда мы пишем IEnumerable squares это как бы новый тип и он перекрывает поля Square

А когда IEnumerable<Square> он реально итератор из <Square> создает, поэтому у нас есть поля

Так а когда IEnumerable squares пишем он же по идее относится чисто к массиву как отдельному типу, то что внутри он не должен трогать, ограничивать, это не становится массив из IEnumerable

Короче я запутался))) Ну и плюс почему мы вообще создаем объекты интерфейса

# Как посмотреть иерархию

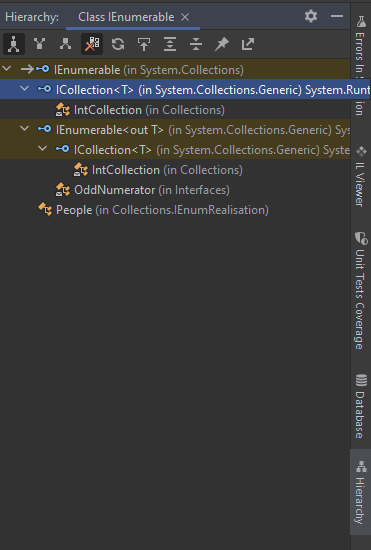
Я решил посмотреть иерархию IEnumerable, вообще я смотрел Array

public abstract partial class Array : ICloneable**,** IList

ну понятно что лист скорее всего наследуется от Енумерабле

public interface IList : ICollection**,** IEnumerable

Я не знаю в студии есть такое или нет, но в райдере есть Hierarchy



И в ней нет IList, хоть он и наследуется от нее, она в студии нормально работает? или просто я не, знаю как посмотреть всех наследников

# Все те же интерфейсы

Здесь <https://metanit.com/sharp/tutorial/4.12.php>

Есть такой код:

public IEnumerable GetBooks(int max)

    {

        for (int i = 0; i < max; i++)

        {

            if (i == books.Length)

            {

                yield break;

            }

            else

            {

                yield return books[i];

            }

        }

    }

Или вот еще был с IEnumerator

   public IEnumerator GetEnumerator()

        {

            for(int i = 0; i < 6; i++)

            {

                yield return i \* i;

            }

        }

Суть и смысл мне понятна, но до меня до сих пор не доходит, почему если yield то обязательно возращает тип IEnumerable, IEnumerable это же просто интерфейс в ктором написано что должны быть функции   
 bool MoveNext**();** object? Current **{** get**; }** void Reset**();**

Я понимаю, почему это называется итератором, но не понимаю как это реализовывается, мы же нигде не прописываем как это работает, **это какая то внутренняя штука** втихоря все понимает?

Потому что если мы сами напишем интерфейс

interface ISomeInterface  
**{** bool MoveNext**();** object? Current **{** get**; }** void Reset**();  
}**

То все равно он не даст нам ничего написать.

# -----------------------------------Новые вопросы—

# Как дебажить катч блок?

# (вроде прочитал понял)зачем тут локер и почему без него нельзя лочить? Мы же туда передаем не то что мы хотим заблокировать?

**КАК Я ПОНЯЛ: системе нужен каждый раз файл чтобы им закрывать как бы то что внутри происходит, данные в него записывать походу, а сама она его не может сделать потому что надо их все отличать. А только я не помню, что будет если двум потокам один и тот же object в lock передать, вроде ничего не изменится, да?**

class Program  
 **{** static int x = 0**;** static object locker = new object**();** static void Main**(**string**[]** args**)  
 {** for **(**int i = 0**;** i < 5**;** i++**)  
 {** Thread myThread = new Thread**(**Count**);** myThread**.**Name = "Поток " + i**.**ToString**();** myThread**.**Start**();  
 }** Console**.**ReadLine**();  
 }** public static void Count**()  
 {** lock **(**locker**)  
 {** x = 1**;** for **(**int i = 1**;** i < 9**;** i++**)  
 {** Console**.**WriteLine**(**"{0}: {1}"**,** Thread**.**CurrentThread**.**Name**,** x**);** x++**;** Thread**.**Sleep**(**100**);  
 }  
 }  
 }  
 }**

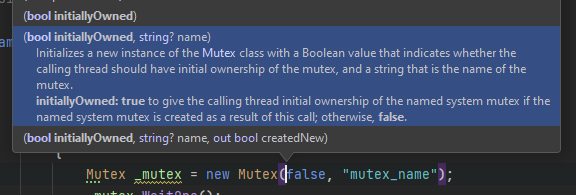
# как отловить Monitor.Pulse? Юзаете вообще его?

# Scratches and consoles

Как настроить кодировку в консоли? не нашел нигде

# Что такое bool initial ownership? Как это работает и как юзать?

То что из другого проекта блочится я проверил, и с true и с false

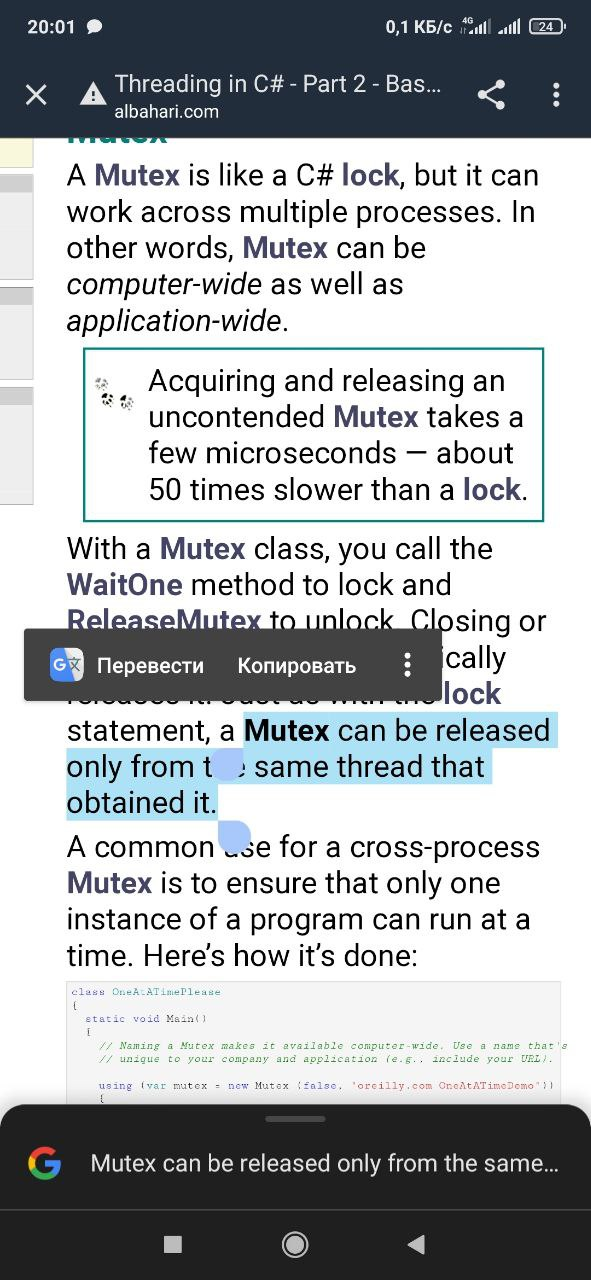


Но если у них не одинаковые имена, как блочить, чето как то мутно

# Что если Mutex’y поставить разные имена в разных проектах? Тогда не будет работать?

Там пример который я делал, он работает потому что одинаковые имена, а если сделать разны то нет. Что делать? Только так?

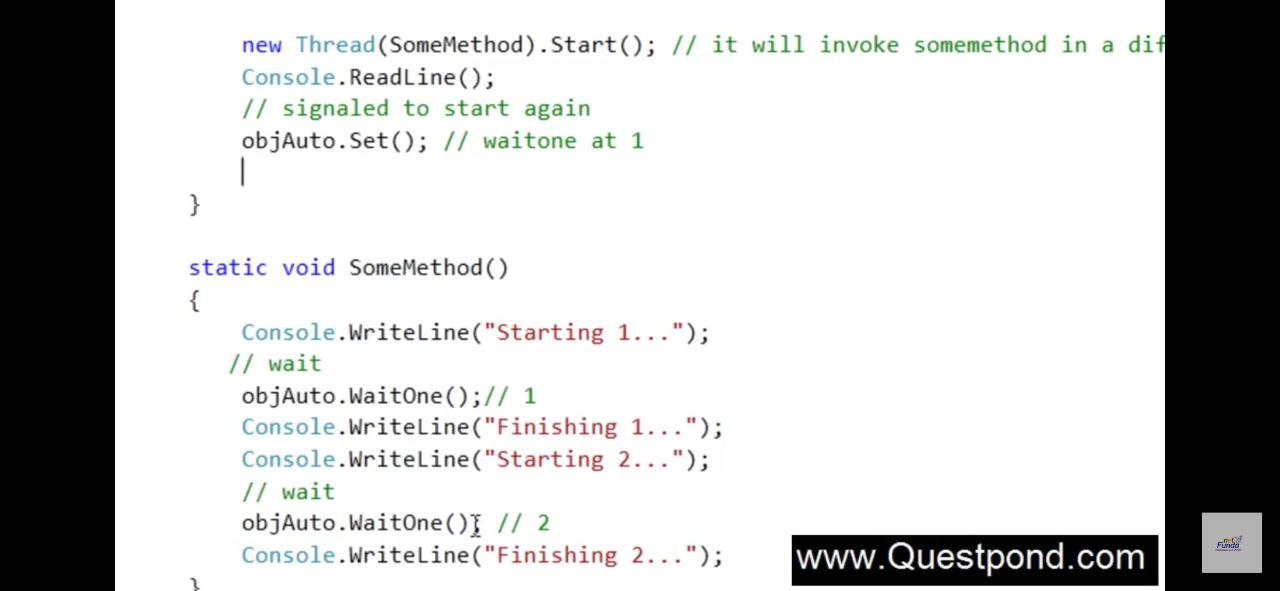
# Mutex может быть релизнут только из потока в котором объявился, так он в асинхроне вообще не может быть релизнут, я так понял, в примере моем в else запихал



# AutoResetEvent vs ManualResetEvent

AutoResetEvent - на каждый waitone надо вызывать Set

ManualResetEvent - на все waitone один set



С AutoResetEvent он остановится перед вторым WaitOne, а с мануал нет

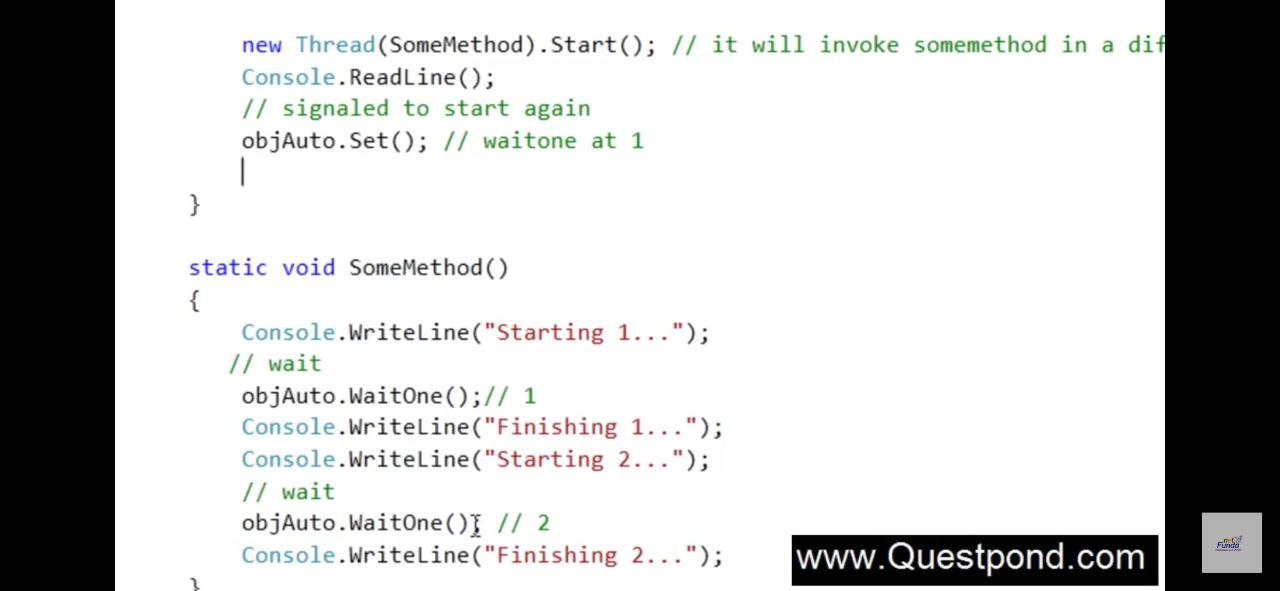
А это все различии?

# Я вообще немного не понимаю, когда что юзать, вот то, что я знаю:

Mutex – работает в нескольких экземплярах Арр, для этого собственно и нужен, медленный

Lock(obj) – быстрый, пускает по одному потоку

AutoResetEvent vs ManualResetEvent понимаю,

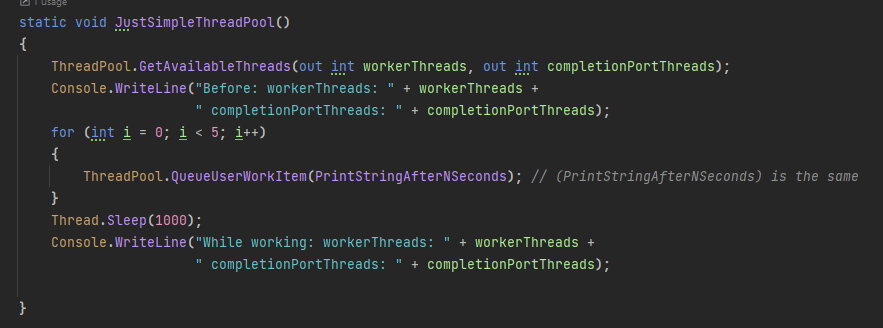


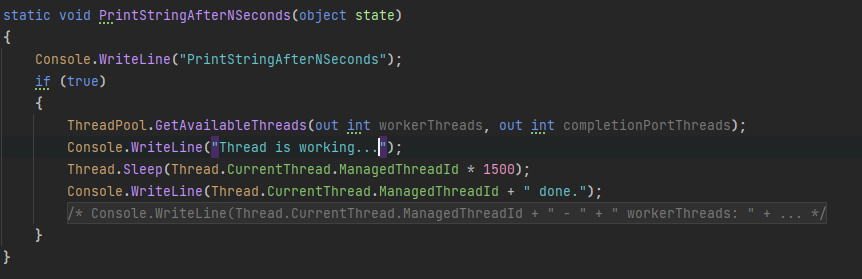
С AutoResetEvent он остановится перед вторым WaitOne, а с мануал нет. Но собственно все.

но когда юзать autoresetEvent, а не Lock тот же не понимаю:

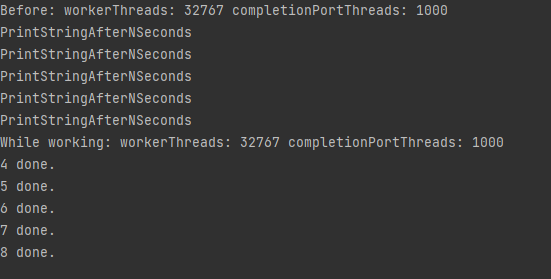
# Вопрос про ThreadPool

В мэйне Я запускаю единственный метод



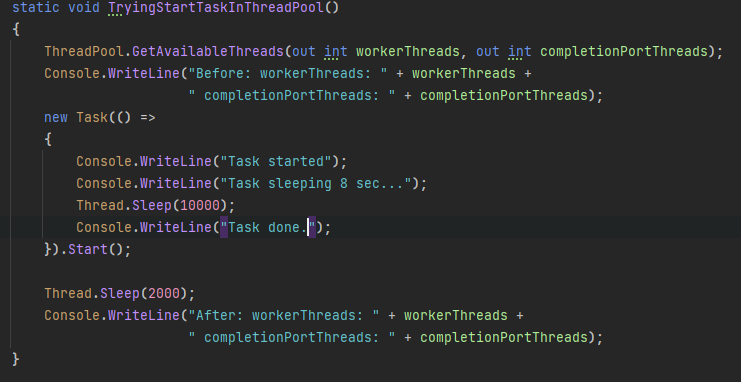


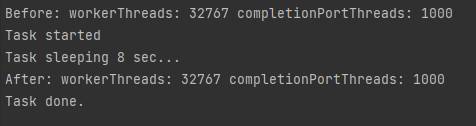
Где смотрю сколько потоков доступно до, потом занимаю 5 потоков и опять вывожу опять сколько доступно, количество доступных не меняется, хотя я спецом сделал чтобы они не освобождались одинаково через sleep



Почему не видно, что они заняты?

Я хотел сделать еще также таски, они же в пуле тоже делаются? Хотел посмотреть, что они тоже занимают потоки, но тоже нет





# Посоветуй ресурс пожалуйста про потоки пулы таски и как с ними работать, чтобы кто нибудь кого нибудь ждал, приостанавливали и запускали, что нибудь, я так больше не могууууу))) неструктурированно очень перемешивается все

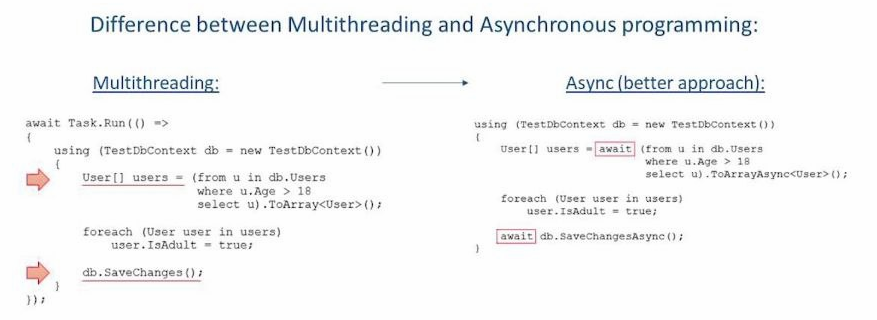
cancelation tokens, скедлеры, thread join’ы там всякие

# MultiThreading vs async

1. Async же работает через threadpool? Просто освобождает поток сама когда надо
2. Вообще асинк- главный тред освобождается|многопоточность-не освобождается в перерывах

Так чел с презентации сказал. Я протсо, не очень понимаю что он имел ввиду под главным тредом. Щас как я понимаю async await:

Вот скрин



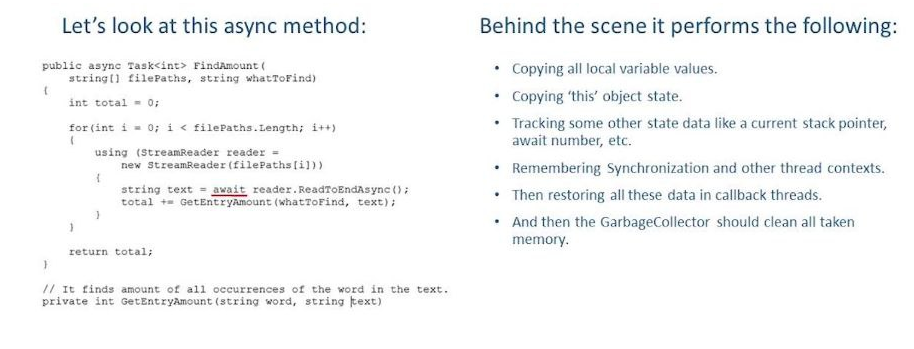
Слева на скрине MultiThread: делается поток **из ThreadPool’a**(правильно же?) и пока делается запрос к бд(подчекнутые) он как бы спит и остается

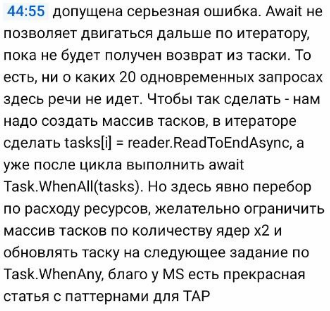
Справа на скрине: делается тоже поток **из ThreadPool’a**(правильно же?) и пока делается запрос к бд(подчекнутые) он освобождает поток, запоминает все, потом запрос проходит, он опять берет поток и идет дальше, и на втором await делает тоже самое

1. Вот пример, почему тут лучше асинк, а не параллельно сделать?

Пример: https://www.youtube.com/watch?v=mtTZu8CXaGE&t=2965s&ab\_channel=FestGroup

будет 1 поток делать все и все успеет, потому что будут перерывы 20мс пока начнет следующий открываться и поток за это время все успеет, но он же будет стоять на await’e, как в коментах и написали(ниже скрин). Я и сам не особо понял почему тут надо await делать а не через 20 параллельных тасков.





# Где потренить асинхронщину и потоки, на плюралсайте не нашел ничего протсо(

# Что за пример там?

Если что я только начал читать про waitHandles, я так понимаю он нужен чтобы ждать потоки через WaitAny, All… Там вообще много всего, но сейчас такаая каша что лучше потом когда со старым четко все будет

<https://docs.microsoft.com/ru-ru/dotnet/api/system.threading.waithandle?view=net-5.0>

static WaitHandle[] waitHandles = new WaitHandle[]

{

new AutoResetEvent(false),

new AutoResetEvent(false)

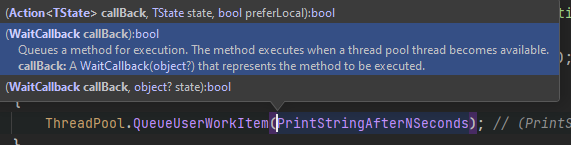
};

Зачем понятно, почему AutoResetEvent(false)? Это типо мы блокируем 2 потока, понятно. почему в массив WaitHandle мы пихаем AutoResetEvent, я залез внутрь и там ничего не понял

А еще вот….

ThreadPool.QueueUserWorkItem(new WaitCallback(DoTask), waitHandles[0]);

ThreadPool.QueueUserWorkItem(new WaitCallback(DoTask), waitHandles[1]);



1й колбэк понятно, а второй стейт, чооо это же object который в делегат передаем, а там waitHandles[1]

Ничо не понятно с точни , хотя смысл ясен

# Сделать поттом:

В нескольких потоках лезешь в базу, Parallel, и попробовать connection через open or openAsync, в pervmon чек колво конекшнов к бд, должно зашкаливать с асинком.

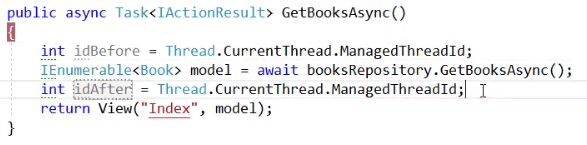
# \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_-

# Async/await

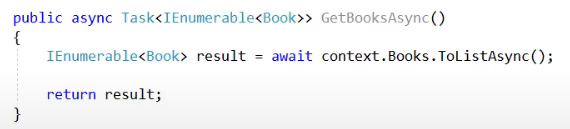
Var a =await someMethodAsync();

Метод someMethodAsync() выполняется в новом потоке?

**Вот например**



BooksRepository.GetBooksAsync:



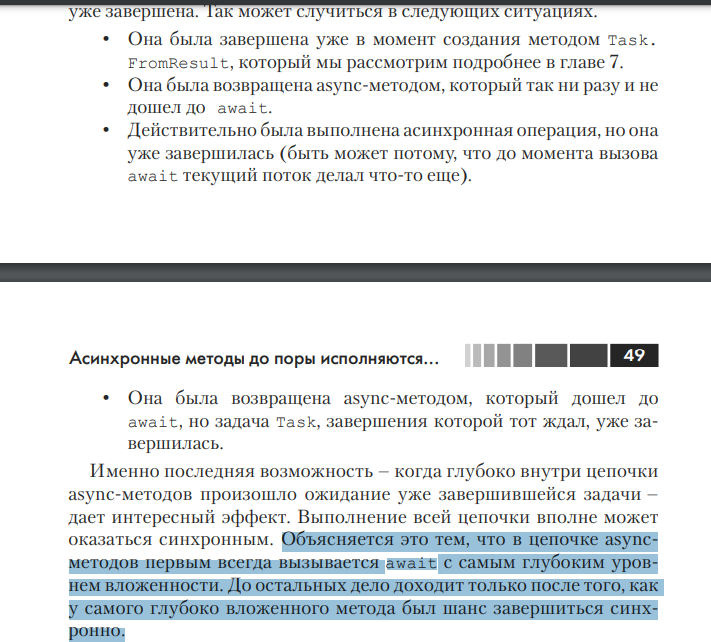
**Правильно понимаю как все происходит?:**

1. Раз **ВСЕ ТАСКИ ВЫПОЛНЯЮТСЯ В THREAD POOL (да ведь???)** значит мы доходим до *await* тут:



1. booksRepositiry.GetBooksAsync() выполняется в другом потоке, тк это таска.
2. В это время поток который делал вот то что на картинке выше не стоит а уходит куда то
3. Потом как таска та отработала продолжается выполнение программы каким то потоком, понятно что не обязательно тем же

# Чего



Не понял, я сделал такую шутку:

static async Task SomeAsync1**()  
{** Thread**.**Sleep**(**1000**);** Console**.**WriteLine**(**"SomeAsync1"**);** await SomeAsync2**();  
}**static async Task SomeAsync2**()  
{** Thread**.**Sleep**(**1000**);** Console**.**WriteLine**(**"SomeAsync2"**);** await SomeAsync3**();  
}**static async Task SomeAsync3**()  
{** Thread**.**Sleep**(**1000**);** Console**.**WriteLine**(**"SomeAsync3"**);  
}**static async void Test**()  
{** await SomeAsync1**();  
}**

static void Main**()  
{** Test**();  
}**

вывод:

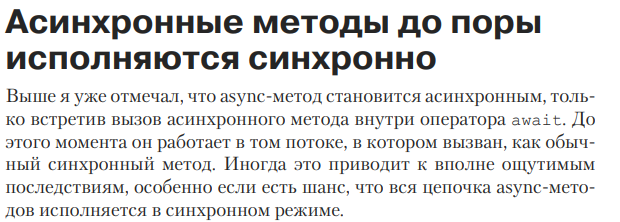
SomeAsync1

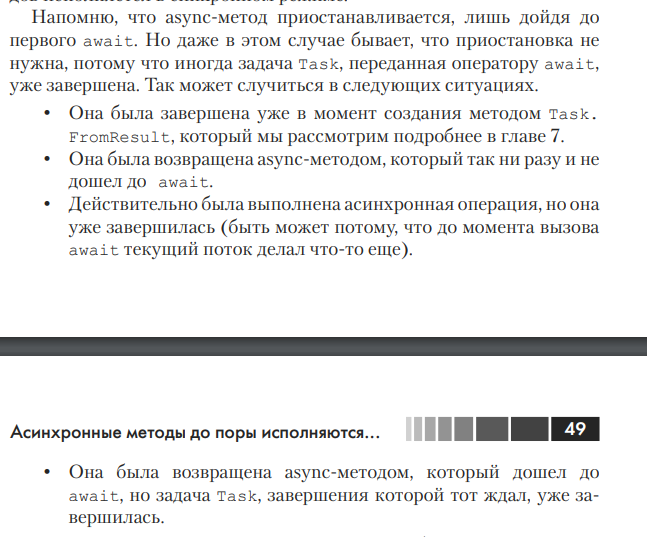
SomeAsync2

SomeAsync3

SomeAsync3- самый глубокий, а началось не с него(что и естественно)

# Короче я не очень понял посыл





Что за ощутимые последствия и в чем проблема то вообще? Ну вызываем иногда синхронно, и что теперь? Все равно не поделаешь ничего

# Почему компилятор сам не ловит ошибки в тасках? Надо обязательно try catch. А если не писать await то вообще не ловит

В книжке просто написали ждать тасков чтобы они ловились как бы

# Юзаете syncronisationContext?

# Короче я зашарил parallel aggregate но как сделать полностью параллельно не придумал