

C# Professional

Потоки.

C# Professional

После урока обязательно



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



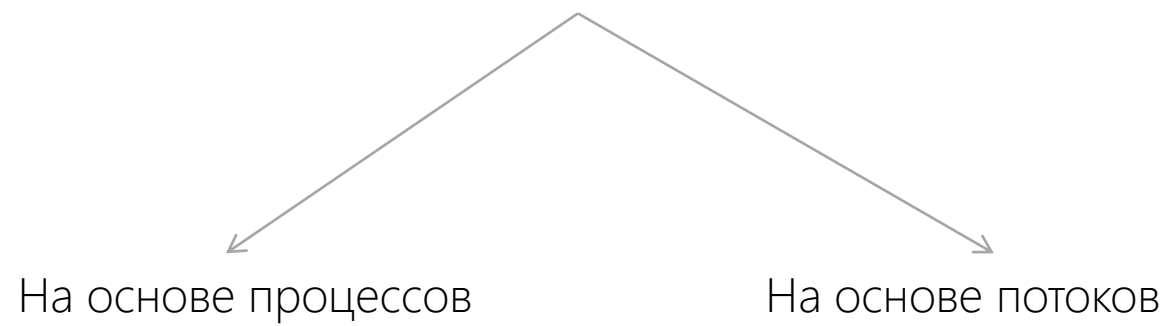
Проверьте как Вы усвоили данный материал на
TestProvider.com

ПОТОКИ

C# Professional

Multitasking

Многозадачность



```
graph TD; A[Многозадачность] --> B[На основе процессов]; A --> C[На основе потоков];
```

На основе процессов

На основе потоков

C# Professional

На основе процессов

Позволяет выполнять одновременно более одной программы в контексте Операционной Системы.

При использовании многозадачности на основе процессов, программа является наименьшей единицей кода, выполнение которой может контролировать планировщик задач.

C# Professional

На основе потоков

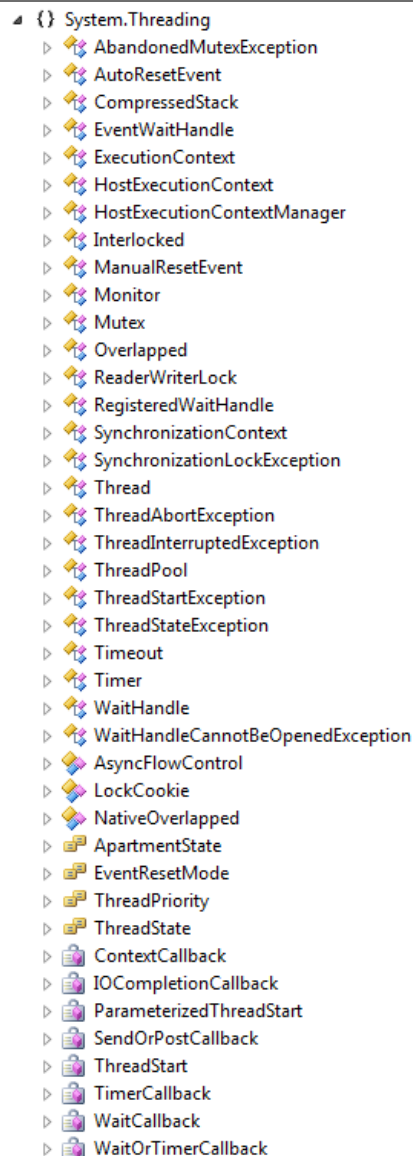
У каждого процесса может быть один или более потоков.

Это означает, что процесс может решать более одной задачи одновременно.

Многозадачность на основе потоков означает параллельное выполнение отдельных частей программы.

C# Professional

Пространство имен

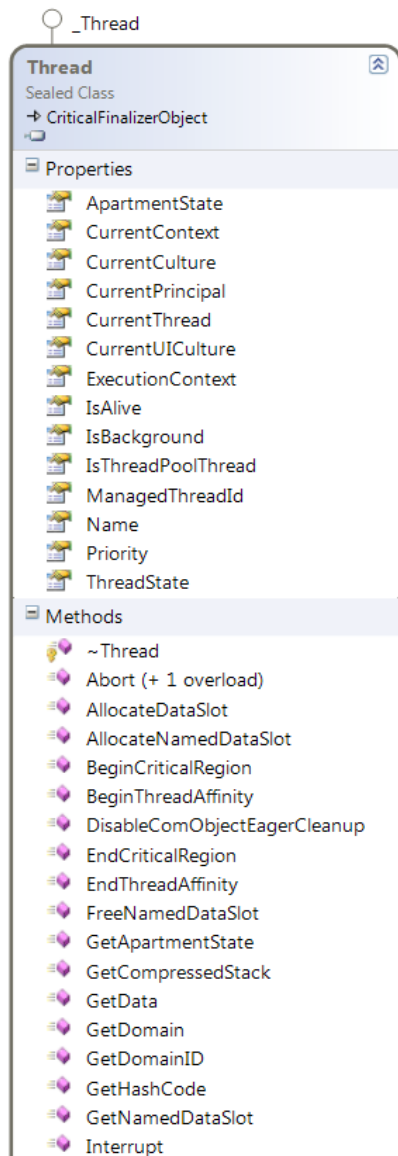


System.Threading - пространство имен для работы с потоками, содержит классы для управления потоками, такие как:

Thread, Monitor, Interlocked

C# Professional

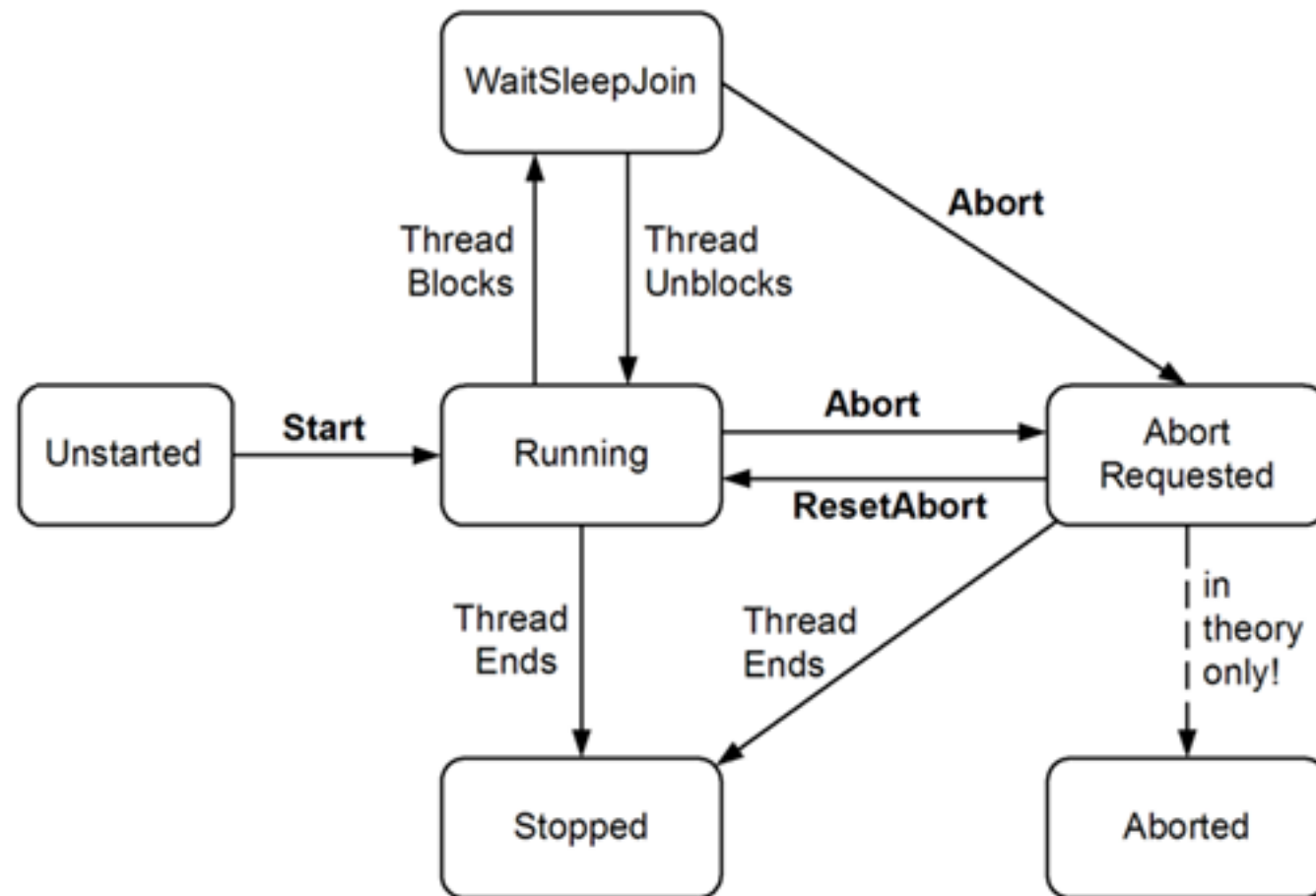
Класс



Класс **Thread**, представляет собой поток. Он позволяет создавать новые потоки, управлять приоритетом потоков и получать информацию о всех потоках, существующих в рамках приложения.

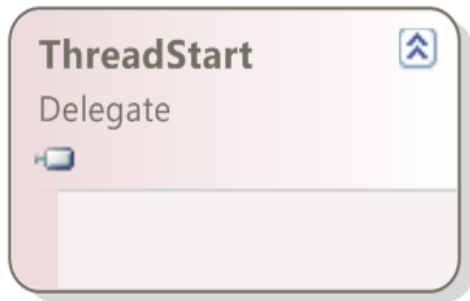
C# Professional

Состояния потока



C# Professional

Делегат



Делегат **ThreadStart** представляет метод, который выполняется в указанном потоке **Thread**.

ThreadStart не позволяет передавать данные в поток.

C# Professional

Делегат



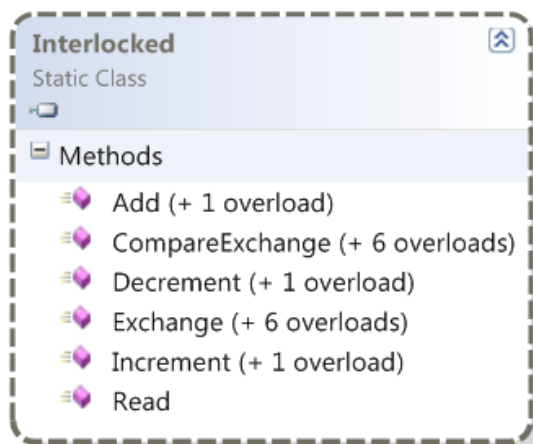
Делегат **ParameterizedThreadStart** представляет метод, который выполняется в указанном потоке **Thread**.

ParameterizedThreadStart позволяет передавать данные в поток упакованные в **object**.

C# Professional

Класс

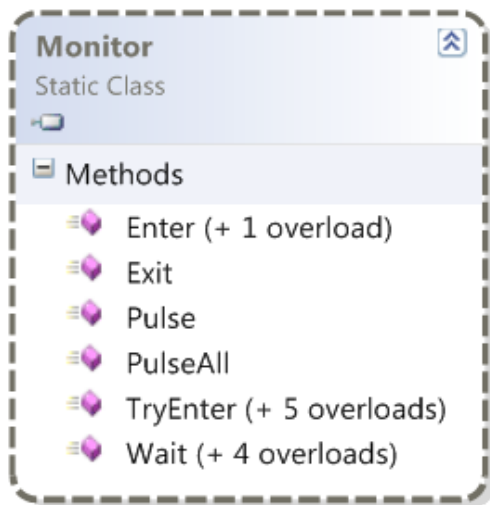
Класс **Interlocked** предоставляет доступ к атомарным операциям, доступным в нескольких потоках.



Операции, выполняемые при помощи методов класса **Interlocked** гарантировано блокируются для остальных потоков, что позволяет избежать некоторых проблем синхронизации.

C# Professional

Класс



Класс **Monitor** представляет собой механизм для синхронизации доступа к объектам.

Класс **Monitor** используется для создания критических секций.

C# Professional

Оператор

При помощи ключевого слова **lock**, блок операторов можно пометить как критическую секцию, получив блокировку взаимного исключения для указанного объекта синхронизации доступа, выполнить действия с общим ресурсом, а затем снять блокировку.

```
lock (obj)
{
    DoSomething();
}
```



```
object tmp = obj;
try
{
    Monitor.Enter(tmp);
    DoSomething();
}
finally
{
    Monitor.Exit(tmp);
}
```

Q&A

Информационный видеосервис для разработчиков программного обеспечения

