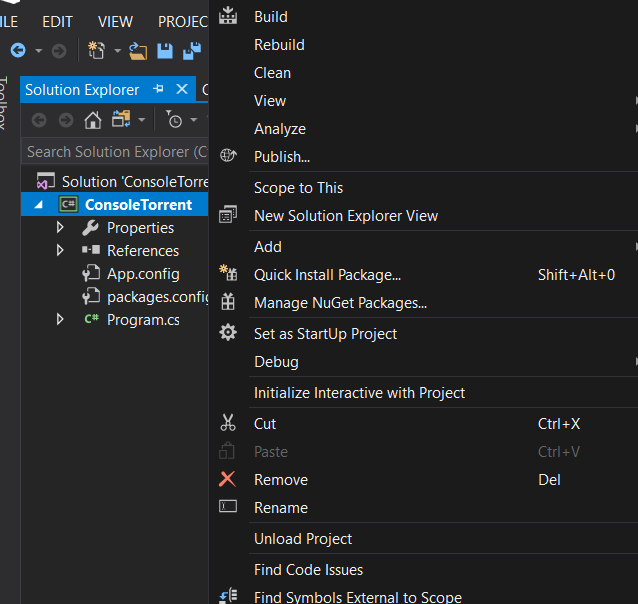
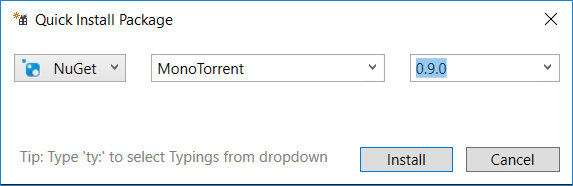
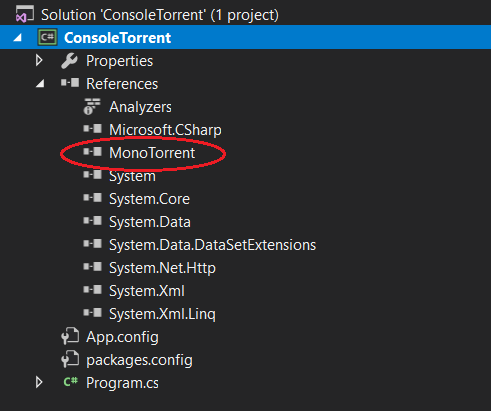
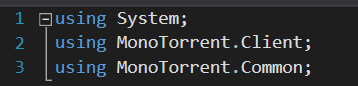
В качестве первого мини-проекта на курсах предлагаю вам создать Torrent-client приложение, способное осуществлять загрузку torrent-файлов.

Сначала это будет обычный консольный загрузчик без возможности докачки и загрузки нескольких файлов одновременно. После приобретения знаний о некоторых необходимых ключевых механизмов С#, программу можно будет развивать и дополнять так, что в конечном итоге получится неплохой торрент-клиент.

Итак, всё, что нам понадобится – это среда разработки Visual Studio и установленная в проект библиотека MonoTorrent для работы с протоколом BitTorrent.

**Подключение библиотеки MonoTorrent**

Как собственно установить эту библиотеку:

1. Для начала создадим обычное консольное приложение для платформы .NET, затем кликаем ПКМ по проекту и выбираем Quick Install Package\*
2. Затем в появившемся окне вводим название библиотеки: MonoTorrent, если всё веррно введено, то справа появится её актуальная версия – 0.9.0, нажимаем Install и ожидаем завершения процесса установки (статус установки отображается в окне Output (по умолчанию его можно открыть в нижней части Visual Studio)) 
3. Если всё прошло успешно, в разделе References проекта появится ссылка на нашу библиотеку 
4. Всё, что нам осталось сделать: подключить пространства имён MonoTorrent.Client и MonoTorrent.Common в нашем .cs файле

**Работа с библиотекой**

Для реализации нашего первого загрузчика, нам потребуется взаимодействовать с тремя основными и двумя вспомогательными классами библиотеки.

Для получения информации о torrent-файле и последующей его загрузки, нам потребуется класс **Torrent** из пространстваимён MonoTorrent.Common, а точнее его статический метод Load, один из конструкторов которого принимает строковый аргумент, содержащий путь до вашего torrent-файла.

Загрузим информацию о файле в экземпляр класса Torrent: 

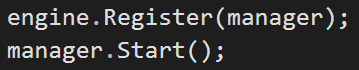
Следующий класс **TorrentManager** из пространстваимён MonoTorrent.Client предназначен для управления одним торрентом (его загрузкой, приостановлением, возобновлением, получением информации об этом и т.д.).

Создадим экземпляр класса **TorrentManager**, передав в конструктор экземпляр нашего торрента (torrent), строку – путь к папке, в которую должна произвестись загрузка файлов и экземпляр вспомогательного класса **TorrentSettings**, созданный по слабой ссылке, который отвеччает за настройки менеджера, так как пока они нам не нужны, не передаём в его конструктор никаких аргументов. 

Последний основной класс необходимый для работоспособности загрузчика – класс **ClientEngine**, необходимый для осуществления контроля и управления за всеми торрентами и клиента в общем, а также получения информации о % загруженных данных, скорости загрузки и т.д.

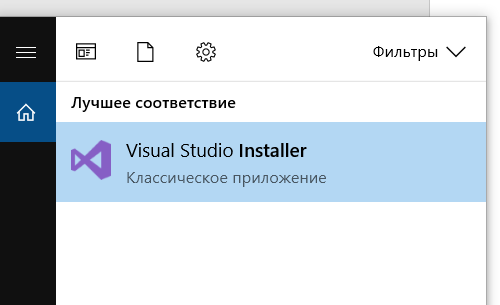
Инстанцируем этот класс, передав в аргументы конструктора экземпляр класса **EngineSettings**, который ответственен за настройки самого движка нашего клиента, для начала пренебрегая этими настройками. 

Всё, что нам теперь осталось сделать – начать отслеживание движком нашего менеджера с помощью метода Register(), а также вызвать метод Start() на экземпляре нашего менеджера для начала загрузки файлов.



Создайте взаимодействие пользователя с программой на свой вкус: передача аргументов с помощью командной строки или ввод с консоли или тому подобное.

Выберите небольшую раздачу на любом торрент-трекере, скачайте torrent-файл (расширение должно быть .torrent) и протестируйте то, что получилось.

\*Если у вас отсутствует эта опция, зайдите в Visual Studio Installer

Выберите вашу версию Visual Studio, а затем опцию Изменить, перейдите во вкладку Отдельные компоненты, и найдите чекбокс Диспетчер пакетов NuGet, выберите его и дождитесь установки.