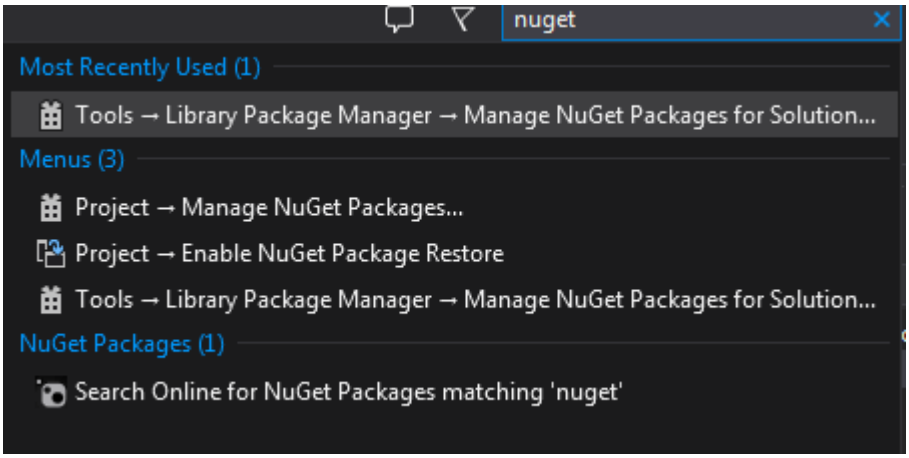
Упражнения: Използване на външна библиотека Tesseract

Ще направим малък примерен проект за разпознаване на текст от изображение чрез **Tesseract**

1. Инсталиране на Tessearct чрез NuGet

Сега трябва да инсталираме **Json.NET** библиотеката чрез **NuGet**. За целта:

- Натиснете **Ctrl + Q**, за да използвате **Quick Launch**
- Въведете **nuget**
- Изберете **Tools -> Library Package Manager -> Manage NuGet Packages for Solution...**



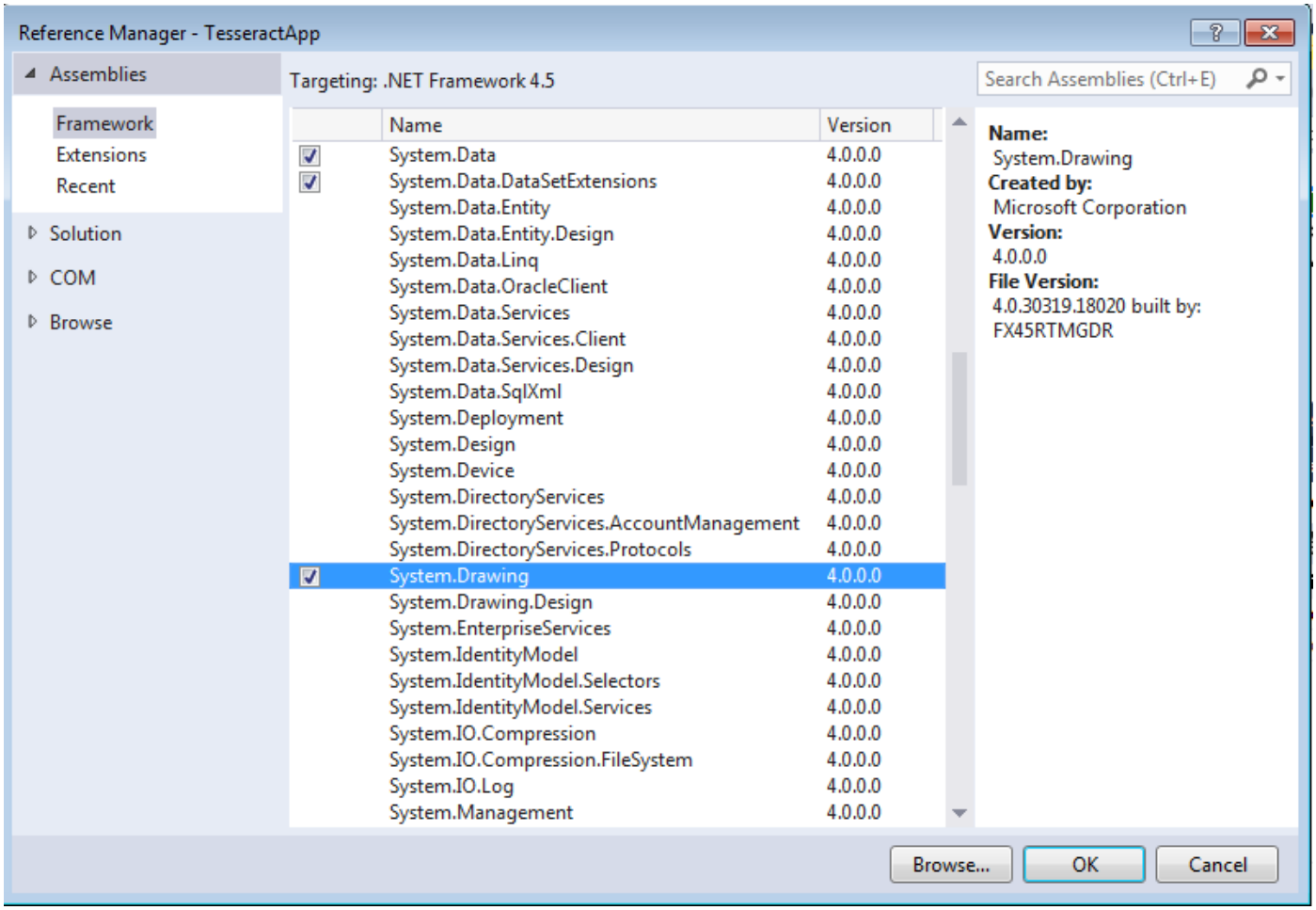
- Намерете **Tesseract** и инсталирайте.

2. Добавяне на файлове за разпознаване

Tessearct е библиотека за разпознаване на текст от изображение /**OCR**/. Няма да навлизаме в техническите детайли на това как работи тази технилогия, а ще се концентрираме върху използването на библиотеката за тази нейна цел. За да разпознава успешно отделни знаци, **Tessearct** се нуждае от модел с данни. Ще използваме готов набор от данни. Файловете на модела трябва да бъдат поставени там където се създава **.exe** файла на приложението. В случая файла се намира в **bin/debug** папката на проекта. Там трябва да поставите и папката **tessdata**, която е предоставена като допълнителен ресурс.

3. Добавяне на референция към System.Drawing

За да работи коректно нашия проект, добавете референция към System.Drawing:



4. Прочитане на текст

Програмният код, с който можем да извършим прочитане е изненадващо прост. Трябва да си създадем един низ, в който да запишем пълния път към файла и името му. След това създаваме обект от клас **TesseractEngine**, указвайки езика, на който е текста, както и името на папката с данните.

След това създаваме обект за изображението, а накрая чрез метода **Process** получаваме и обект за страницата – този обект има метод **GetText()**, който съдържа нашия текст.

```
string fileName = @"C:\Users\pc\Pictures\test.png";
using (var engine = new TesseractEngine(@"tessdata", "eng"))
{
    using (var image = Pix.LoadFromFile(fileName))
    {
        using (var page = engine.Process(image))
        {
            string text = page.GetText();
            Console.WriteLine(text);
        }
    }
}
```

За файл използвайте подадения към темата тестов файл – това е изображение на сорс кода на тази програма.

Резултатът от изпълнението на програмата е доста добър, макар и с известни неточности:

```
Console.WriteLine("Let's read! );

string fileuame : @"CI\User's\pc\Pictur'eS\test.png";

using (var engine : new TesseractEngine(@"tessdata", "eng"))
{

using (var image : Pix.LoadFr'omFile(fileName))

{

using (var page

{

engine. Process(image))

string text page.GetText();
Console.writeLine(text);


Press any key to continue . . .
```

Ще се задоволим на този етап с този резултат, но ще добавим, че все пак той може да бъде подобрен по редица начини, например:

- Допълнителна обработка на изображението, чрез методи и класове от библиотеката на **Tesseract** или външен софтуер
- Допълнително или по-добро трениране на данните на модела на **Tesseract**

Министерство на образованието и науката (МОН)

- Настоящият курс (презентации, примери, задачи, упражнения и др.) е разработен за нуждите на Национална програма "**Обучение за ИТ кариера**" на МОН за подготовка по професия "Приложен програмист".

- Курсът е базиран на учебно съдържание и методика, предоставени от **фондация "Софтуерен университет"** и се разпространява под **свободен лиценз CC-BY-NC-SA** (Creative Commons Attribution-Non-Commercial-Share-Alike 4.0 International).