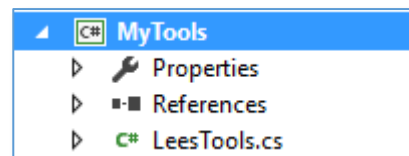


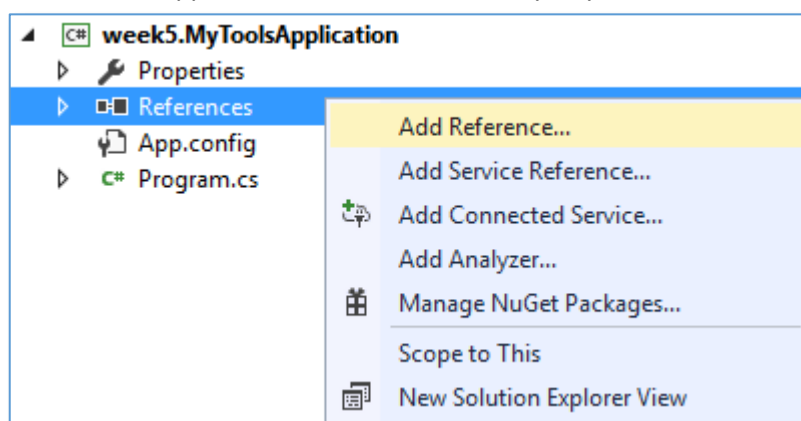
Opdracht 1 – Een bibliotheek met handige methoden

We refactoren opgave 0 van week 1 tot een Applicatie en een Class Library. Deze Class Library kunnen we dan ook in andere projecten gebruiken.

- a) Maak een nieuw Project aan van het type 'Class Library' met de naam 'MyTools'.
 Maak in dit project een nieuwe class LeesTools aan.
 Kopieer de 2 LeesInt methoden en de LeesString methode uit opdracht 0 van week 1 naar deze class.
 Maak deze 3 methoden public én static.



- b) Maak een nieuw Project aan van het type 'Console Application' met de naam 'MyToolsApplication'.
 Kopieer de Main methode van opgave 0 uit week 1 naar deze nieuwe Console Application.
 → Maak een referentie in de Application naar de Class Library 'MyTools'.



→ Zet boven aan in je Program:

```
using MyTools;
```

→ Pas de methode-aanroepen aan, zodat ze naar de Library-class LeesTools verwijzen (zie code hieronder).

→ Test je programma.

Je kan aan deze MyTools Library meerdere classes toevoegen voor eventueel andere Tool-methoden.

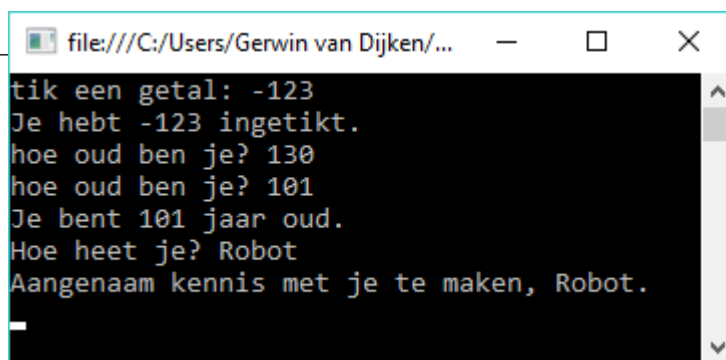
```
static void Main()
{
    Program myProgram = new Program();
    myProgram.Start();
}

void Start()
{
    int getal = LeesTools.LeesInt("tik een getal. ");
    Console.WriteLine("Je hebt {0} ingetikt.", getal);

    int leeftijd = LeesTools.LeesInt("hoe oud ben je? ", 0, 120);
    Console.WriteLine("Je bent {0} jaar oud.", leeftijd);

    string naam = LeesTools.LeesString("Hoe heet je? ");
    Console.WriteLine("Aangenaam kennis met je te maken, {0}.", naam);

    Console.ReadKey();
}
```



Opdracht 2 – CandyCrush met logica laag

Bij het CandyCrush programma kunnen we onderscheid maken tussen gebruikersinteractie en game-logica. Het is dan ook logisch om voor de CandyCrush-logica een aparte laag te maken.

- a) Maak een nieuwe CandyCrush-applicatie (Console-project) aan en kopieer de code v/d CandyCrush-applicatie uit week 4 hier naar toe.
→ Test of de applicatie nog werkt.

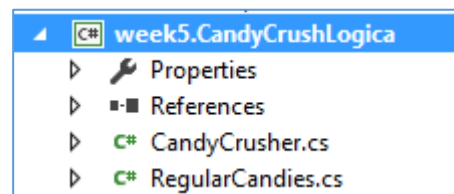
- b) Voeg een nieuw Class Library project met de naam 'CandyCrushLogica' toe aan je Solution.

Voeg een `class CandyCrusher` toe aan deze Class Library.

We plaatsen de volgende methoden in class CandyCrusher (dus in de logica-laag): `ScoreRijAanwezig` en `ScoreKolomAanwezig`.

Verplaats ook het bestand met `enum RegularCandies` naar de logica-laag (wijzig de namespace zodat deze gelijk is aan de namespace van class CandyCrusher).

→ Build deze Class Library, er moeten geen compile fouten zijn.



- c) Nu moet de CandyCrush-applicatie (aangemaakt in a) de logica-laag nog gaan gebruiken:

- Maak een reference vanuit de CandyCrush-applicatie naar de logica-laag.
- Voeg een `using`-statement voor de namespace van de logica-laag toe in Program.cs.

→ Zorg ervoor, dat alles weer gaat werken, door de aanroepen naar de methoden uit de logica-laag aan te passen.

Opdracht 3 – Vertalen

In deze opdracht gaan we een bestand inlezen met Nederlandse woorden met de bijbehorende Engelse vertaling. De gebruiker kan met dit programma een Nederlands woord invoeren en het programma toont vervolgens de Engelse vertaling. Op Moodle vind je een bestand "dictionary.csv" dat je kunt gebruiken.

- a) Maak een nieuw Console-project aan met de naam 'Vertaling' en voeg de volgende methode toe:

```
Dictionary<string, string> ReadWords(string filename)
```

Deze methode leest woorden uit een tekstbestand en plaatst deze woorden in een Dictionary. Elke regel bevat zowel een Nederlands woord als de Engelse vertaling, gescheiden met een puntkomma (bv: "slecht;bad"). Plaats elke regel in de dictionary met als sleutel (key) het Nederlandse woord, en als waarde (value) de Engelse vertaling. Je kunt elke ingelezen regel opsplitsen via `line.Split(';')`, zodat je een array krijgt met (in dit geval 2) losse velden. Retourneer de dictionary nadat alle regels verwerkt zijn.

→ Roep de ReadWords-methode aan vanuit de Start-methode. Controleer via de debugger of er woorden in de dictionary staan (je kunt eenvoudig de 'Count' controleren).

- b) Voeg de volgende methode toe:

```
void TranslateWords(Dictionary<string, string> words)
```

Deze methode leest steeds een woord van de gebruiker totdat het woord 'stop' is ingevoerd. Voor alle andere woorden wordt een vertaling gegeven via de words-dictionary. Als de dictionary het (Nederlandse) woord niet bevat, dan wordt dit door getoond (bv "woord niet gevonden"), anders wordt de (Engelse) vertaling getoond. Een aantal voorbeelden zie je in de screenshot hieronder, links (het gebruik van verschillende kleuren is optioneel).

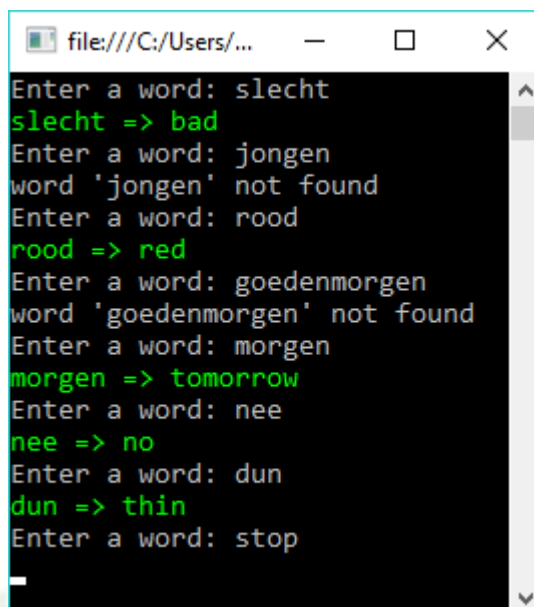
→ Roep de TranslateWords-method aan vanuit de Start-methode.

→ Test verschillende woorden om het programma te testen.

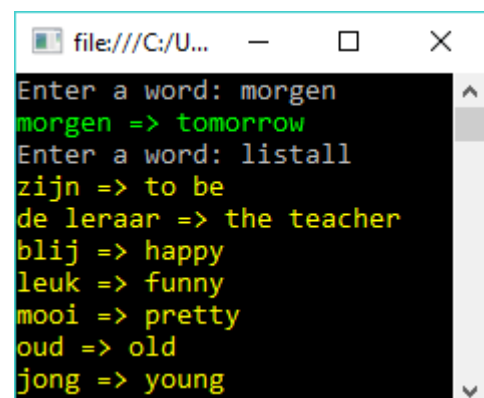
- c) Voeg de volgende methode toe:

```
void ListAllWords(Dictionary<string, string> words)
```

Deze methode toont alle woorden die in de dictionary voorkomen, met de bijbehorende vertaling. Wijzig methode TranslateWords zodat bij het woord "listall" de methode ListAllWords wordt aangeroepen. Zie een voorbeeld hieronder, rechts (het gebruik van verschillende kleuren is optioneel).



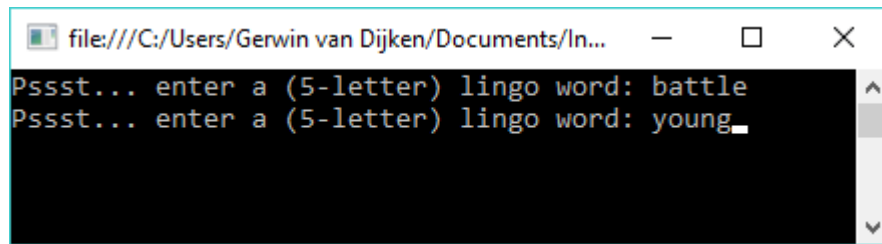
```
file:///C:/Users/...
Enter a word: slecht
slecht => bad
Enter a word: jongen
word 'jongen' not found
Enter a word: rood
rood => red
Enter a word: goedenmorgen
word 'goedenmorgen' not found
Enter a word: morgen
morgen => tomorrow
Enter a word: nee
nee => no
Enter a word: dun
dun => thin
Enter a word: stop
```



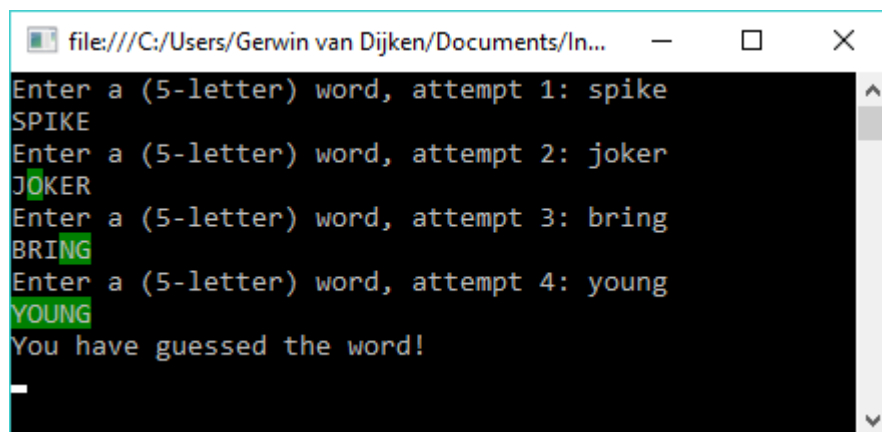
```
file:///C:/U...
Enter a word: morgen
morgen => tomorrow
Enter a word: listall
zijn => to be
de leraar => the teacher
blij => happy
leuk => funny
mooi => pretty
oud => old
jong => young
```

Opdracht 4 - Lingo

In de les is de aanpak van het Lingo spel behandeld: de hoofdtak en de benodigde subtaken. Implementeer deze applicatie door de genoemde taken (methoden) te implementeren. Je kunt uiteraard de pseudocode uit de slides gebruiken.



```
file:///C:/Users/Gerwin van Dijken/Documents/In...  
Pssst... enter a (5-letter) lingo word: battle  
Pssst... enter a (5-letter) lingo word: young_
```



```
file:///C:/Users/Gerwin van Dijken/Documents/In...  
Enter a (5-letter) word, attempt 1: spike  
SPIKE  
Enter a (5-letter) word, attempt 2: joker  
JOKER  
Enter a (5-letter) word, attempt 3: bring  
BRING  
Enter a (5-letter) word, attempt 4: young  
YOUNG  
You have guessed the word!  
_
```