

## Opdracht 1 – Rapport

In deze opdracht maken we een applicatie waarmee we een cijferlijst (rapport) kunnen bijhouden. Een rapport bevat vakken (zoals “Programmeren 2”, en elk vak bevat een theorie-cijfer en een praktijk-cijfer.

### class Vak en enum Beoordeling

- Maak een `class Vak` met de velden:
  - naam van het vak (string)
  - theorie cijfer (int)
  - praktijkbeoordeling (enumeratie)
 De praktijkbeoordeling kan de volgende waarden bevatten: Geen, Absent, Onvoldoende, Voldoende en Goed.  
 → Maak hiervoor een `enum PraktijkBeoordeling`.
- Maak de volgende methoden voor het vullen en tonen van een `PraktijkBeoordeling` en een `Vak`:
 

```
PraktijkBeoordeling LeesPraktijkBeoordeling(string vraag)
void ToonPraktijkBeoordeling(PraktijkBeoordeling praktijkBeoordeling)
Vak LeesVak(string vraag)
void ToonVak(Vak vak)
```

 → Gebruik in deze methoden de methoden voor het lezen van `int`'s en `string`'s uit week 1.  
 → Test de werking van de methoden door ze in de `Start`-methode aan te roepen.
- Maak in de `Start`-methode een lijst met beoordelingen:
 

```
List<Vak> rapport;
```

 → Vul deze lijst met 3 vakken. Maak hiervoor methode `List<Vak> LeesRapport(int aantalVakken)`.  
 → Toon het rapport op het scherm. Maak hiervoor methode `void ToonRapport(List<Vak> rapport)`.
- Geef class `Vak` de volgende methodes:
 

```
public bool IsBehaald()
public bool IsCumLaude()
```

 Een vak is behaald, als de praktijkbeoordeling Voldoende of Goed is én het theorie cijfer 55 of hoger is.  
 Een vak is cum laude behaald, als de praktijkbeoordeling Goed is én het theorie cijfer 80 of hoger is.
- Wijzig nu methode `ToonRapport` om de volgende dingen te bepalen:
  - of de student geslaagd is (alle vakken van het rapport zijn behaald)
  - of de student cum laude geslaagd is (alle vakken van het rapport zijn cum laude behaald)
  - hoeveel herkansingen de student moet doen (aantal vakken dat niet behaald is)
 → Druk de resultaten af onder het rapport (zie voorbeeld hieronder).

LeesVak

LeesVak

LeesVak

ToonRapport

```

file:///D:/Visual Studio 2015/Projects/PP2/opdracht 2-3 Rapport/...
Voer een vak in.
Naam van het vak: PP1
Cijfer voor PP1: 87
0. Geen 1. Absent 2. Onvoldoende 3. Voldoende 4. Goed
Practicum beoordeling voor PP1: 3

Voer een vak in.
Naam van het vak: PP2
Cijfer voor PP2: 54
0. Geen 1. Absent 2. Onvoldoende 3. Voldoende 4. Goed
Practicum beoordeling voor PP2: 4

Voer een vak in.
Naam van het vak: OO
Cijfer voor OO: 79
0. Geen 1. Absent 2. Onvoldoende 3. Voldoende 4. Goed
Practicum beoordeling voor OO: 1

PP1      : 87  Voldoende
PP2      : 54  Goed
OO       : 79  Absent
HELAAS, je bent gezakt en hebt 2 herkansingen
        
```

## Opdracht 2 – Galgje

### De class GalgjeSpel

Bij het spel Galgje gaat het erom om een geheim woord te raden. De twee essentiële onderdelen van het spel zijn dus het geheime woord en het (tot zover) geraden woord. Zie [www.playhangman.com](http://www.playhangman.com).

- a) We maken voor het spel Galgje dus een class `GalgjeSpel`, die twee strings bevat: `geheimWoord` en `geradenWoord`.

`GalgjeSpel` krijgt ook een methode:

```
public void Init(string geheimWoord)
```

Deze methode slaat het geheime woord op, en vult het geraden woord met evenveel punten als het aantal karakters in het geheime woord.

Dus als het geheime woord "eetlepel" is dan wordt het geraden woord gevuld met 8 punten: ".....".

We zullen de class `GalgjeSpel` in de loop van de opdracht verder uitbreiden met methoden.

→ Test je class met de volgende code (in je Start methode):

```
GalgjeSpel galgje = new GalgjeSpel();
galgje.Init("eetlepel");
Console.WriteLine("Het geheime woord is: " + galgje.geheimWoord);
Console.WriteLine("Het geraden woord is: " + galgje.geradenWoord);
```

### Random woord genereren

- b) Implementeer een (Program) methode met signatuur:

```
List<string> WoordenLijst()
```

Deze methode retournt een lijst met woorden. Je maakt in deze methode (hardcoded) een lijst met 20 woorden om galgje mee te spelen.

→ Maak in de Start-methode een `List<string>` `woordenLijst` en vul deze lijst m.b.v. de methode `WoordenLijst`.

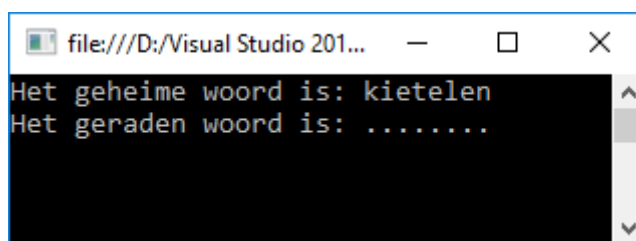
- c) Implementeer een (Program) methode met signatuur:

```
string SelecteerWoord(List<string> woorden)
```

Deze methode kiest een random woord uit de lijst met woorden.

→ Roep de methode `SelecteerWoord` aan vanuit de Start-methode met als parameter de woordenlijst die je in vraag b hebt gemaakt. Het geselecteerde woord zal als geheim woord gaan dienen.

→ Toon ter controle dit woord in de Start-methode (de gebruiker mag straks het woord natuurlijk niet zien...).



## Letters lezen en weergeven

- d) Implementeer een (Program) methode:

```
bool SpeelGalgje(GalgjeSpel galgje)
```

Deze methode retourneert **true** als het woord geraden is door de gebruiker, anders **false**.

Voor het bijhouden van de ingevoerde letters maken we een char-list aan in de methode SpeelGalgje:

```
List<char> ingevoerdeLetters;
```

Deze methode retourneer voorlopig **true**.

→ Roep de methode SpeelGalgje aan vanuit de Start-methode.

We gebruiken deze methode in de volgende vragen als test plek. Aan het eind van de opdracht gaan we deze methode definitief implementeren.

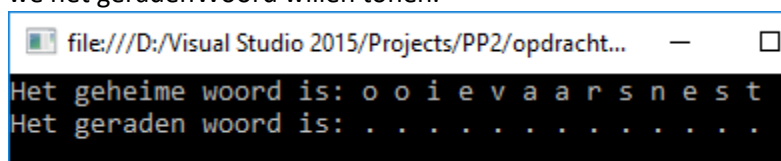
- e) Maak een (Program) methode:

```
void ToonWoord(string woord)
```

Deze methode toont het opgegeven woord met spaties tussen de letters.

We gebruiken deze methode telkens wanneer we het geradenWoord willen tonen.

→ Roep de methode ToonWoord aan in de methode SpeelGalgje en test de methode voorlopig door zowel het geheime woord als het geraden woord af te drukken.



- f) Maak een (Program) methode:

```
void ToonLetters(List<char> letters)
```

Deze methode toont de letters gescheiden door spaties.

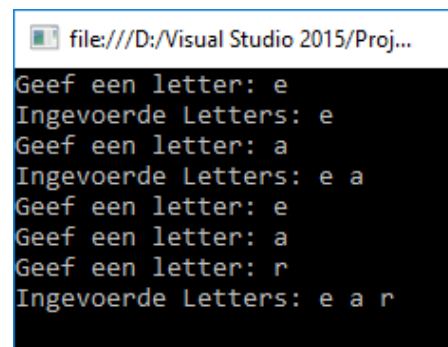
→ Roep de methode ToonLetters aan in de methode SpeelGalgje om de methode te testen. Voeg hiervoor tijdelijk een aantal letters toe aan de lijst met ingevoerde letters.

- g) Maak een (Program) methode:

```
char LeesLetter(List<char> verbodenLetters)
```

Deze methode leest letters in totdat de ingelezen letter niet in de lijst met verbodenletters voorkomt. Gebruik hiervoor `verbodenLetters.Contains(letter)`.

→ Roep de methode LeesLetter aan in de methode SpeelGalgje en voeg de ingelezen letter toe aan de lijst met ingevoerde letters.



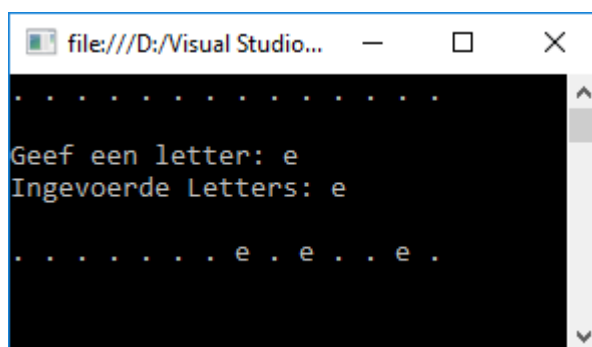
- h) Geef class GalgjeSpel een methode:

```
public bool RaadLetter(char letter)
```

Deze methode controleert of het geheime woord de opgegeven letter bevat, en retourneert **true** als dat het geval is, anders **false**. Ook voor strings kan je `Contains` gebruiken.

Als de letter in het geheime woord zit, wijzig dan het geraden woord door de letter op de juiste plaats(en) te zetten.

→ Test de methode in SpeelGalgje.



## Woord raden

De gebruiker krijgt maximaal 8 pogingen om het galgje-woord te raden. Het aantal (resterende) pogingen wordt verlaagd als de gebruiker een letter invoert die niet voorkomt in het galgje-woord. Het programma stopt als het woord geraden is, of als de gebruiker geen pogingen meer over heeft.

i) Geef class `GalgjeSpel` een methode:

```
public bool IsGeraden()
```

Deze methode retourneert `true` als het geraden woord gelijk is aan het geheime woord, anders `false`.

j) We gaan nu de methode `SpeelGalgje` definitief implementeren.

→ Maak een lus in de methode `SpeelGalgje` waarin:

- de gebruiker een nieuwe letter kan invoeren
- deze letter wordt toegevoegd aan de lijst met ingevoerde letters
- de ingevoerde letters worden getoond
- er wordt gekeken of de nieuwe letter in het woord voorkomt
- het aantal resterende pogingen wordt getoond
- het geraden woord wordt getoond

Je gebruikt hiervoor natuurlijk de methoden die je in de vorige vragen hebt gemaakt.

→ Zorg er voor dat de lus stopt als het woord geraden is of als het aantal resterende pogingen gelijk is geworden aan 0.

→ De methode retourneert of het woord geraden is (`true` of `false`). De `Start`-methode toont vervolgens een passende boodschap op het scherm. Als de gebruiker het geheime woord niet heeft geraden, toon het dan.

```
file:///D:/Visual Studio 2015/Pr...
. . . . .
Geef een letter: e
Aantal pogingen over: 8
Ingevoerde Letters: e
. . . . . e .
Geef een letter: a
Aantal pogingen over: 8
Ingevoerde Letters: e a
. . . a . . . e .
Geef een letter: i
Aantal pogingen over: 8
Ingevoerde Letters: e a i
. i . a . i . e .
Geef een letter: o
Aantal pogingen over: 7
Ingevoerde Letters: e a i o
. i . a . i . e .
```

```
file:///D:/Visual Studio 2015/Pr...
Aantal pogingen over: 6
Ingevoerde Letters: e a i o n l s
. i . a . i n e s
Geef een letter: v
Aantal pogingen over: 6
Ingevoerde Letters: e a i o n l s v
v i . a . i n e s
Geef een letter: t
Aantal pogingen over: 6
Ingevoerde Letters: e a i o n l s v t
v i t a . i n e s
Geef een letter: m
Aantal pogingen over: 6
Ingevoerde Letters: e a i o n l s v t m
v i t a m i n e s
Je hebt gewonnen!
```

N.B. Bewaar je uitwerking, want we gaan de opdracht later verder uitbreiden door b.v. de woorden uit een file te lezen.