

## Opdracht 0 – Handige methodes

Voor het invoeren van gegevens voor je programma's moet je telkens dezelfde soort code schrijven. In deze opdracht maak je een aantal methoden, die het schrijven van deze code makkelijker gaat maken.

Je gaat deze methoden ook in opdracht 1 en 2 gebruiken. Het loont dus de moeite om ze zo duidelijk mogelijk te schrijven en zo goed mogelijk te laten werken.

Deze week zullen we de methoden nog kopiëren, maar later maken we onze eigen code-bibliotheek.

- a) Maak een methode met signatuur:

```
int LeesInt(string vraag)
```

die de opgegeven vraag stelt en vervolgens een int inleest. De ingelezen int wordt geretourneerd.

→ Test de methode (met de code hieronder).

- b) Maak daarnaast ook een variant van LeesInt die controleert of de ingelezen int zich in een bepaald interval bevindt. Je hebt nu dus twee methoden LeesInt. Dit noemen we overloading.

De signatuur:

```
int LeesInt(string vraag, int min, int max)
```

Wanneer het ingevoerde getal zich niet in het interval [min..max] bevindt, wordt opnieuw om invoer gevraagd.

→ Maak hierbij gebruik van de methode LeesInt uit vraag a.

→ Test de methode (met de code hieronder).

- c) Maak ook een methode

```
string LeesString(string vraag)
```

Deze stelt wederom een vraag en retourneert het antwoord.

→ Test de methode (met de code hieronder).

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

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

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

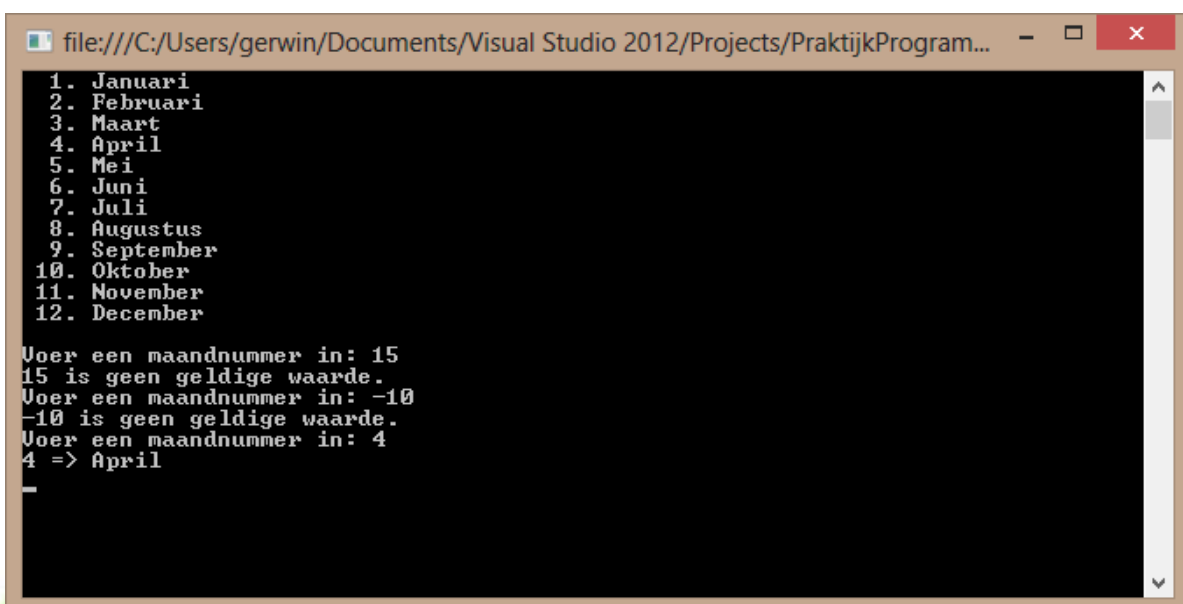
    string naam = LeesString("Hoe heet je? ");
    Console.WriteLine("Aangenaam kennis met je te maken, {0}.", naam);

    Console.ReadKey();
}
```

```
file:///C:/Users/Gerwin van Dijken/...
tik een getal: -123
Je hebt -123 ingetikt.
hoe oud ben je? 130
hoe oud ben je? 101
Je bent 101 jaar oud.
Hoe heet je? Robot
Aangenaam kennis met je te maken, Robot.
```

## Opdracht 1 – Enumeraties – Maand

- a) Maak een enumeratie type aan met als naam Maand (sla deze op in een apart bestand met de naam Maand.cs) en geef de enum Maand 12 opties: Januari, Februari, ..., November en December. Zorg er voor dat de eerste optie (Januari) de waarde 1 krijgt.
- b) Maak een methode met signatuur:  
`void PrintMaand( Maand maand )`  
die de opgegeven maand afdrukt (alleen de naam).  
→ Roep de methode PrintMaand aan vanuit de Start-methode om hem te testen.
- c) Maak een methode met signatuur:  
`void PrintMaanden()`  
die gebruik maakt van de methode PrintMaand uit vraag b om alle enum-opties van Maand onder elkaar te printen (gebruik een loop).  
Zorg dat de getallen (1..12) 'right-aligned'<sup>1</sup> zijn (zie voorbeeld hieronder).  
→ Roep methode PrintMaanden aan vanuit de Start-methode.
- d) Maak een methode met signatuur:  
`Maand VraagMaand( string vraag )`  
waarin de gebruiker een maandnummer kan invoeren dat vervolgens wordt geconverteerd van een int-waarde naar een enum-waarde. Retourneer deze Maand-waarde.  
In de Start-methode print vervolgens deze waarde.  
→ Hierbij maak je natuurlijk gebruik van de methode PrintMaand uit vraag b.
- e) Wat gebeurt er als je een ongeldig getal invoert (<1 of >12)?  
Wijzig de VraagMaand-methode zodat er een melding komt als er een ongeldig getal wordt ingevoerd; gebruik hiervoor de methode Enum.IsDefined(typeof(Maand), ...) die `true` of `false` teruggeeft. Lees net zolang een maandnummer totdat IsDefined `true` teruggeeft.



```
file:///C:/Users/gerwin/Documents/Visual Studio 2012/Projects/PraktijkProgram...  
1. Januari  
2. Februari  
3. Maart  
4. April  
5. Mei  
6. Juni  
7. Juli  
8. Augustus  
9. September  
10. Oktober  
11. November  
12. December  
  
Voer een maandnummer in: 15  
15 is geen geldige waarde.  
Voer een maandnummer in: -10  
-10 is geen geldige waarde.  
Voer een maandnummer in: 4  
4 => April
```

<sup>1</sup>zie o.a. <http://www.csharp-examples.net/align-string-with-spaces/>

## Opdracht 2 – Structures - Persoon

Je gaat in deze opdracht integers inlezen. Het is dus handig om de LeesInt-methoden uit opdracht 0 her te gebruiken. Kopieer deze methoden naar je Program.cs.

### De struct Persoon

- Maak een struct `Persoon` (in een apart bestand) met de velden `Voornaam`, `Achternaam`, `Leeftijd` en `Woonplaats`. Maak (in de `Start`-methode) een variabele van type `Persoon` en vul alle velden van deze struct.
- Maak een methode  
`Persoon` `LeesPersoon()`  
 waarmee je een `Persoon` kan inlezen.  
 → Gebruik hierbij de methoden `LeesInt` en `LeesString` uit opdracht 0.
- Maak een methode  
`void` `PrintPersoon(Persoon p)`  
 die de meegegeven `Persoon` print.

### De enum GeslachtType

- Geef de struct `Persoon` een veld `Geslacht` toe van het type `GeslachtType` (opties: `Man` en `Vrouw`).
- Maak ook een methode  
`GeslachtType` `LeesGeslacht(string vraag)`  
 die een geslacht kan inlezen. Gebruik deze in de methode `LeesPersoon` om het geslacht in te lezen.
- Maak ook een methode  
`void` `PrintGeslacht(GeslachtType geslacht)`  
 die 'm' of 'v' afdrukt afhankelijk van het opgegeven geslacht.  
 → Wijzig ook de `PrintPersoon`-methode zodat deze het geslacht afdrukt m.b.v. de methode `PrintGeslacht`.

### Personen in een Array

- Maak een `Persoon`-array aan met lengte 3. Vul deze array (in een loop) met gegevens van de gebruiker (lees alle informatie dus in). Je gebruikt hiervoor natuurlijk de methode `LeesPersoon`. Print vervolgens alle personen (weer in een loop) via de `PrintPersoon`-methode.

### Vier verjaardag

- Maak een methode  
`void` `VierVerjaardag(...)`  
 die een `Persoon` ontvangt en hiervan de leeftijd verhoogt. Print vervolgens deze persoon opnieuw (in de `Start`-methode) om er zeker van te zijn dat de leeftijd is aangepast. Uitvoer van de applicatie moet ongeveer zijn zoals hieronder is weergegeven.

N.B. Het vieren van de verjaardag lijkt niet goed te werken. Waardoor komt dat denk je? Zoek naar een oplossing, waarbij je nog steeds in de methode de leeftijd ophooft en in de `Start`-methode de leeftijd print!

```

file:///C:/Users/gerwin/Documents/Visual...
Geef achternaam: de Vries
Geef geslacht (m/v): m
Geef leeftijd: 27
Geef woonplaats: Beverwijk

Geef voornaam: Truus
Geef achternaam: Hofman
Geef geslacht (m/v): v
Geef leeftijd: 34
Geef woonplaats: Uitgeest

Karel Dijkstra (m)
56 jaar, Alkmaar

Peter de Vries (m)
27 jaar, Beverwijk

Truus Hofman (v)
34 jaar, Uitgeest

Verjaardag vieren van Karel Dijkstra...
Karel Dijkstra (m)
57 jaar, Alkmaar
  
```

## Opdracht 3 – Classes - Yahtzee

### De class Dobbelsteen

- a) Maak een class Dobbelsteen aan (in een apart bestand).

- b) Plaats een methode met signatuur:

```
public void Gooi()
```

in de class Dobbelsteen (ja, dat kan ook). In deze Gooi-methode wordt een nieuwe random-waarde gegenereerd tussen 1 en 6. Deze sla je op in het veld waarde.

- c) Geef de class Dobbelsteen ook een methode met signatuur:

```
public void ToonWaarde()
```

Deze methode toont de waarde van de dobbelsteen op het scherm.

- d) Test de class Dobbelsteen door een Dobbelsteen-variabele te maken en deze een aantal keer te gooien en telkens de waarde af te drukken.

```
class Dobbelsteen
{
    public int waarde;
    static Random rnd = new Random();
}
```

```
void Start()
{
    Dobbelsteen d1, d2;
    d1 = new Dobbelsteen(); // dobbelsteen 1
    d2 = new Dobbelsteen(); // dobbelsteen 2

    d1.Gooi();
    d2.Gooi();

    d1.ToonWaarde();
    d2.ToonWaarde();
}
```

### De class YahtzeeGame

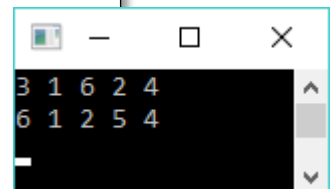
- e) Maak een class YahtzeeGame aan (in een apart bestand) met daarin een Dobbelsteen-array dobbelstenen. Voeg een methode `public void Init` toe om alle dobbelstenen aan te maken (via `new Dobbelsteen`).
- f) Plaats ook een methode in de class YahtzeeGame met signatuur `public void Gooi()`. In deze Gooi-methode worden alle 5 dobbelstenen gegooid (m.b.v. de methode Gooi van de class Dobbelsteen).
- g) Geef de class YahtzeeGame ook een methode met de signatuur `public void ToonWorp()`. die de waarde van alle dobbelstenen toont (zie voorbeeld). Je gebruikt hiervoor natuurlijk de methode ToonWaarde van de class Dobbelsteen.
- h) Test nu de class YahtzeeGame met de volgende code:

```
static void Main(string[] args)
{
    Program myProgram = new Program();
    myProgram.Start();
}

void Start()
{
    YahtzeeGame yahtzeeGame = new YahtzeeGame();
    yahtzeeGame.Init();

    yahtzeeGame.Gooi(); // gooï dobbelstenen
    yahtzeeGame.ToonWorp(); // toon resultaat

    yahtzeeGame.Gooi();
    yahtzeeGame.ToonWorp();
}
```



Checken op Yahtzee, ThreeOfAKind, FourOfAKind etc.

- i) Plaats een methode in de `YahtzeeGame` met signatuur:

```
public bool Yahtzee()
```

In deze Yahtzee-methode wordt bepaald of er Yahtzee is gegooid (5 dobbelstenen met hetzelfde aantal ogen); zo ja, dan wordt `true` teruggegeven, anders `false`.

- j) Gebruik de code hieronder (zie kader). Daarmee kun je testen of je de class `YahtzeeGame` goed hebt geïmplementeerd. Een mogelijke uitvoer van het programma zie je hieronder.

Hoeveel pogingen had je (ongeveer) verwacht als uitkomst?

*(hoeveel yahtzee-opties zijn er, en hoeveel variaties met 5 dobbelstenen?)*

- k) [optioneel] Breidt de class uit met de volgende Yahtzee-opties:

- ThreeOfAKind
- FourOfAKind

- l) [optioneel] Breidt de class uit met de volgende Yahtzee-opties:

- FullHouse
- KleineStraat
- GroteStraat

```

4 5 5 6 6
2 1 5 5 3
4 1 1 1 6
2 3 4 4 3
3 5 3 1 1
3 3 1 4 3
5 1 6 2 2
4 4 4 5 5
4 3 3 2 3
3 4 2 5 6
4 5 4 3 5
6 3 5 5 3
5 6 1 2 4
3 1 5 6 3
5 5 5 5 5
Aantal pogingen nodig voor yahtzee <5 dezelfde>: 1412
  
```

```

static void Main(string[] args)
{
    Program myProgram = new Program();
    myProgram.Start();
}

void Start()
{
    YahtzeeGame yahtzeeGame = new YahtzeeGame();
    yahtzeeGame.Init();
    SpeelYahtzee(yahtzeeGame); // speel het spel
}

void SpeelYahtzee(YahtzeeGame game)
{
    int aantalPogingen = 0;

    do
    {
        game.Gooi();           // gooi dobbelstenen
        game.ToonWorp();        // toon worp

        aantalPogingen++;

    } while (!game.Yahtzee());

    Console.WriteLine("Aantal pogingen nodig voor yahtzee: {0}", aantalPogingen);
}
  
```