Stack vs Heap

referenties en instanties

(http://www.youtube.com/watch?v=clOUdVDDzIM)

(lokale variabelen)

Heap

```
public class Persoon
{
    public string naam;
    public int leeftijd;
}
```

(lokale variabelen)

Er wordt alleen ruimte gealloceerd voor <u>een referentie</u> naar een Persoon object.

p 0x00000000

Heap

```
void Start()
{
    Persoon p;
}
```

(lokale variabelen)

```
0x00000000
p
```

```
void Start()
   Persoon p;
   p.naam = "Piet";
```

```
NullReferenceException was unhandled.
                                                                              ×
Object reference not set to an instance of an object.
Troubleshooting tips:
Check to determine if the object is null before calling the method.
                                                                             \wedge
Use the "new" keyword to create an object instance.
Get general help for this exception.
Search for more Help Online...
Exception settings:
Break when this exception type is thrown
Actions:
View Detail...
Copy exception detail to the clipboard
Open exception settings
```

(lokale variabelen)

Heap

(objecten / class instanties)

Persoon

```
naam = null
leeftijd = 0
```

Er wordt ruimte gealloceerd voor de data members van een Persoon object. De members (fields) krijgen default values.

```
void Start()
{
   new Persoon();
}
```

(lokale variabelen)

Heap

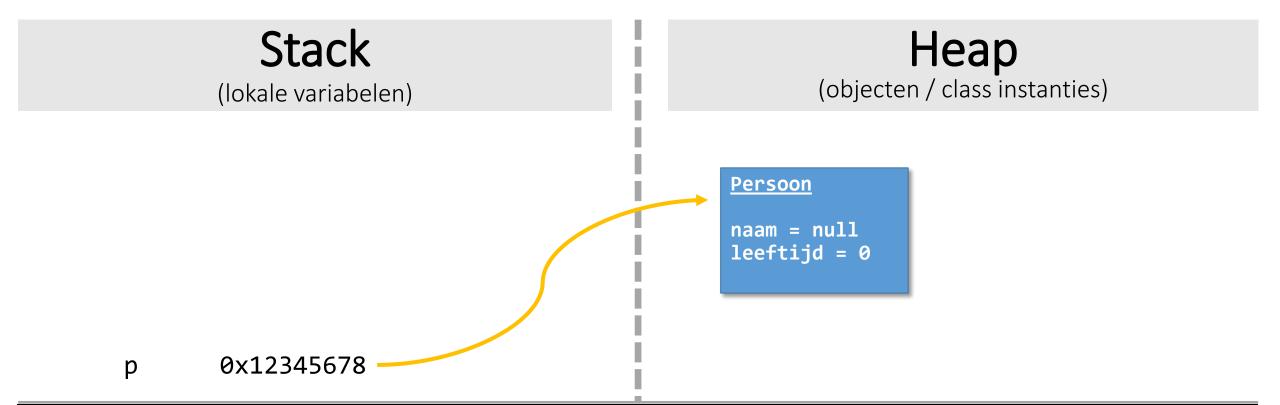
(objecten / class instanties)

Persoon

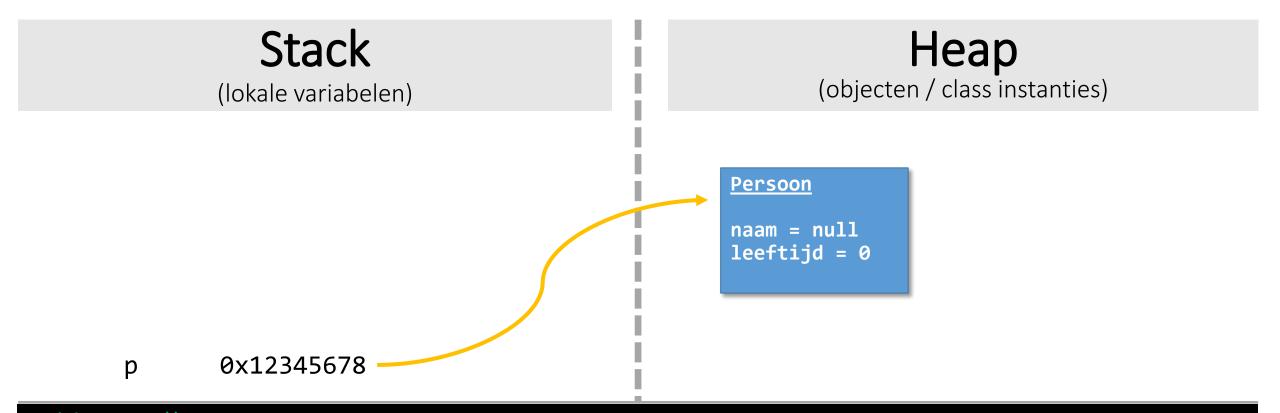
```
naam = null
leeftijd = 0
```

p 0x00000000

```
void Start()
{
    Persoon p;
    new Persoon();
}
```



```
void Start()
{
    Persoon p;
    p = new Persoon();
}
```



```
void Start()
{
    Persoon p = new Persoon();
}
```

Stack (lokale variabelen)

Heap

(objecten / class instanties)

```
Persoon

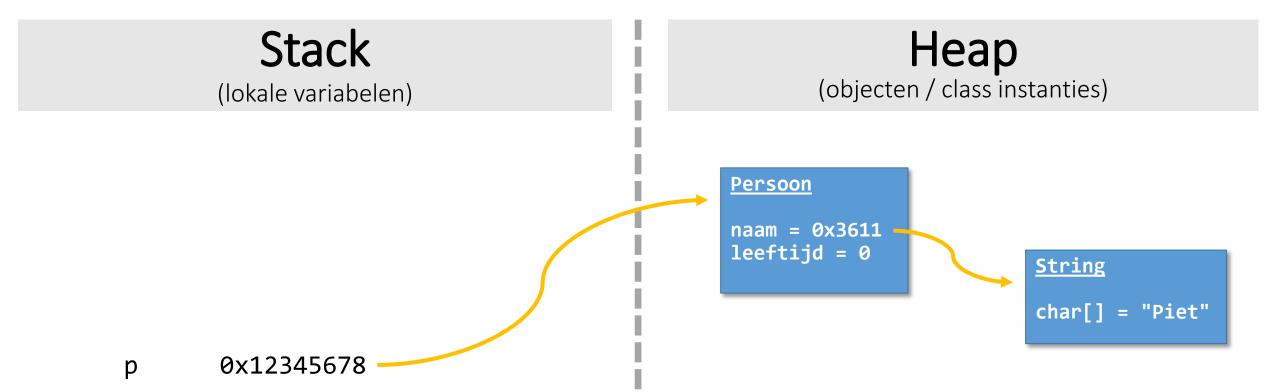
naam = "Piet"
leeftijd = 0
```

```
void Start()
```

p

```
Persoon p = new Persoon();
p.naam = "Piet";
```

0x12345678



```
void Start()
{
    Persoon p = new Persoon();
    p.naam = "Piet";
}
```

Stack (lokale variabelen)

Heap

```
Persoon

naam = "Piet"
leeftijd = 28
```

```
p 0x12345678
```

```
void Start()
{
    Persoon p = new Persoon();
    p.naam = "Piet";
    p.leeftijd = 28;
}
```

(lokale variabelen)

Heap

```
public class Persoon
{
    public string naam;
    public int leeftijd;

    public Persoon(string naam, int leeftijd)
    {
        this.naam = naam;
        this.leeftijd = leeftijd;
    }
}
```



p

Heap

(objecten / class instanties)

```
Persoon
```

```
naam = "Piet"
leeftijd = 28
```

```
void Start()
{
    Persoon p = new Persoon("Piet", 28);
```

0x12345678

Door de contructor met parameters kunnen we direct de gegevens meegeven bij het aanmaken van een Persoon.

(lokale variabelen)

Heap

(objecten / class instanties)

Persoon

naam = "Piet"
leeftijd = 28

Er is geen
verwijzing meer
naar dit object; de
Garbage Collector
ruimt dit object
op, ergens in de
nabije toekomst.

p 0x00000000

```
void Start()
{
    Persoon p = new Persoon("Piet", 28);
    p = null;
}
```

(lokale variabelen)

Heap

(objecten / class instanties)

<u>Persoon</u>

naam = "Piet"
leeftijd = 28

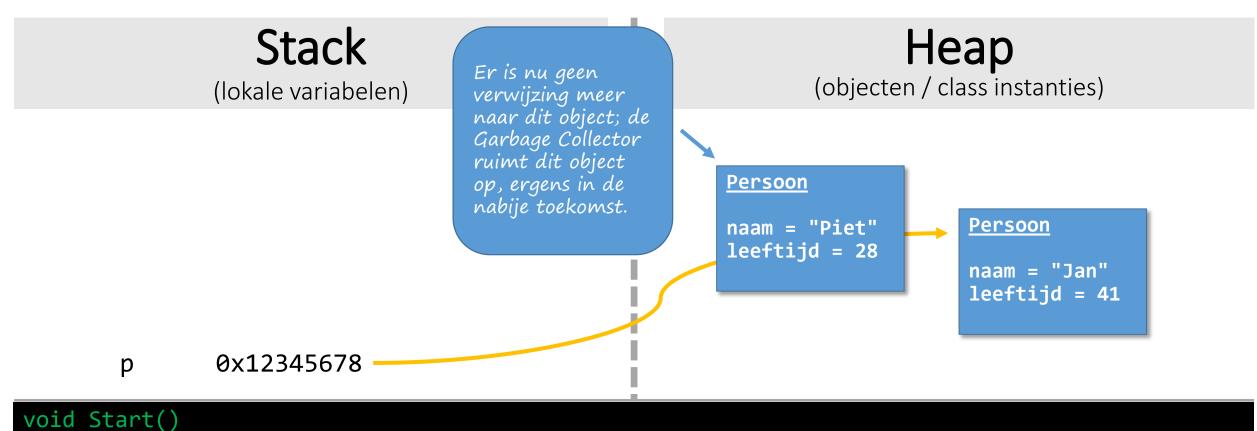
<u>Persoon</u>

naam = "Jan"
leeftijd = 41

```
void Start()
{
    Persoon p = new Persoon("Piet", 28);
    new Persoon("Jan", 41);
}
```

0x12345678

p



```
Persoon p = new Persoon("Piet", 28);
p = new Persoon("Jan", 41);
}
```

(lokale variabelen)

Heap

(objecten / class instanties)

```
p2  0x23456789
p1  0x12345678
```

```
<u>Persoon</u>
```

naam = "Jan"
leeftijd = 41

```
void Start()
{
    Persoon p1 = new Persoon("Piet", 28);
    Persoon p2 = new Persoon("Jan", 41);
}
```

(lokale variabelen)

Geen referentie maar een waarde, omdat int een 'value type' is.

leeftijd

0

Heap

```
void Start()
{
    int leeftijd;
}
```

(lokale variabelen)

Heap

```
Persoon

naam = "Piet"
leeftijd = 0
```

```
void Start()
{
   int leeftijd = 28;
   Persoon p = new Persoon();
   p.naam = "Piet";
}
```

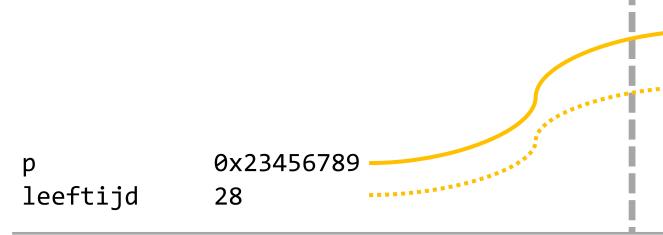
(lokale variabelen)

Heap

(objecten / class instanties)

Persoon

naam = "Piet"
leeftijd = 28



<u>Waarde</u> (28) wordt gekopieerd omdat int een 'value type' is.

Er is dus <u>geen connectie</u> tussen lokale variabele leeftijd en de 'member field' leeftijd.

```
void Start()
{
    int leeftijd = 28;
    Persoon p = new Persoon();
    p.naam = "Piet";
    p.leeftijd = leeftijd;
}
```

(lokale variabelen)

Heap

```
<u>Persoon</u>
```

```
naam = "Piet"
leeftijd = 28
```

```
void Start()
{
    int leeftijd = 28;
    Persoon p = new Persoon();
    p.naam = "Piet";
    p.leeftijd = leeftijd;
    leeftijd = 31;
}

Ook al verandert lokale variabele leeftijd, de 'member field' leeftijd blijft ongewijzigd.
```

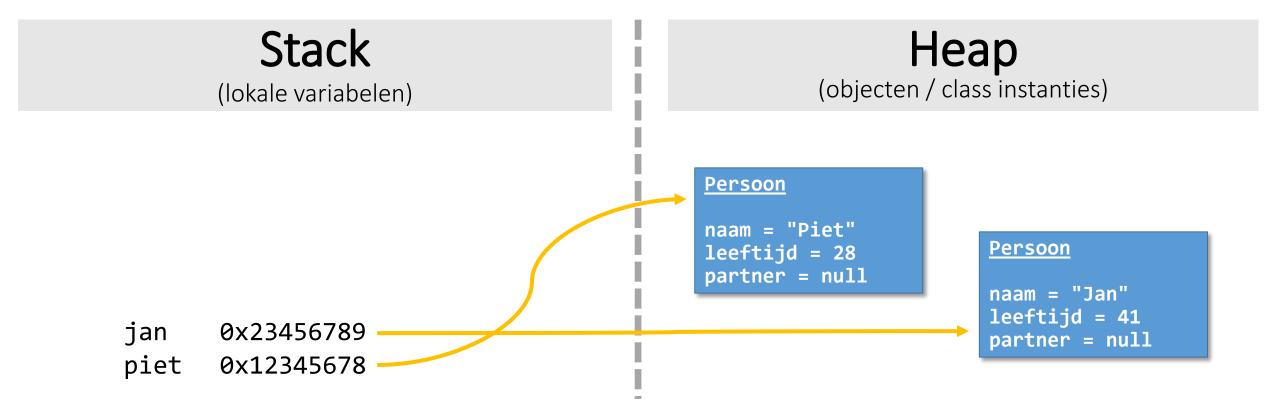
(lokale variabelen)

Heap

```
public class Persoon
{
    public string naam;
    public int leeftijd;

    public Persoon partner;
}

Class Persoon bevat nu dus
    ook zelf een referentie
    naar een object (een ander
    Persoon).
```



```
void Start()
{
    Persoon piet = new Persoon("Piet", 28);
    Persoon jan = new Persoon("Jan", 41);
}
```

Stack Heap (objecten / class instanties) (lokale variabelen) **Persoon** naam = "Piet" leeftijd = 28 Persoon partner = 0x23456789 naam = "Jan" leeftijd = 41 jan 0x23456789 partner = null piet 0x12345678

```
void Start()
{
    Persoon piet = new Persoon("Piet", 28);
    Persoon jan = new Persoon("Jan", 41);
    piet.partner = jan;
}
```

(lokale variabelen)

Heap

```
jan 0x0000000
piet 0x12345678
```

```
void Start()
{
    Persoon piet = new Persoon("Piet", 28);
    Persoon jan = new Persoon("Jan", 41);
    piet.partner = jan;
    jan = null;
}
```

Stack (lokale variabelen)

jan

piet

Heap

(objecten / class instanties)

```
Persoon
naam = "Piet"
leeftijd = 28
                           Persoon
partner =
                           naam = "Jan"
        0x23456789
                           leeftijd = 41
                           partner =
                                   0x12345678
```

```
void Start()
   Persoon piet = new Persoon("Piet", 28);
    Persoon jan = new Persoon("Jan", 41);
    piet.partner = jan;
    jan = null;
    piet.partner.partner = piet;
```

0x00000000

0x12345678

Wie is nu piet.partner.partner.partner.partner.partner????