

C# Advanced Sealed class & Interfaces

Koen Bloemen



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





Sealed class Interfaces

Sealed class

- Abstract class => **moet** overgeërfd worden
- Sealed class => **kan niet** overgeërfd worden

```
public sealed class Persoon  
{  
}
```

Persoon
Sealed Class

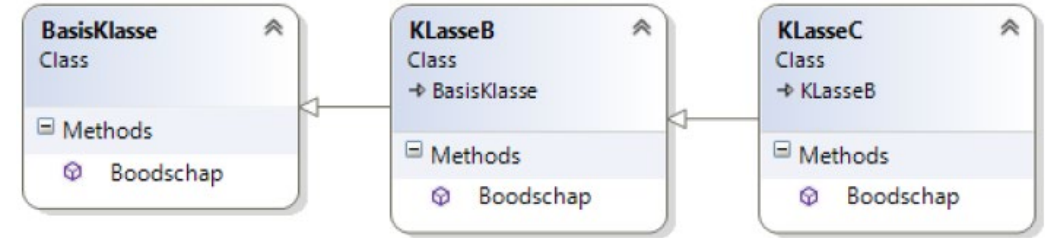
Bedienden
Class

```
public class Bedienden : Persoon // Gaat niet!!  
{  
}
```

 class sealedKlasse.Bedienden

'Bedienden': cannot derive from sealed type 'Persoon'

Sealed class



- Niet enkel op class niveau, kan ook op methods toegepast worden

```
public class BasisKlasse
{
    public virtual void Boodschap()
    {
        MessageBox.Show("Boodschap vanuit de basisklasse.", "Virtuele methode", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
public class KlasseB : BasisKlasse
{
    public sealed override void Boodschap()
    {
        MessageBox.Show("Boodschap vanuit de klasse die overerft van de basisklasse.",
            "Overschrijven en verzegelen van methode", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
public class KlasseC : KlasseB
{
    public override void Boodschap()
    {
        MessageBox.Show("Boodschap vanuit de klasse die overerft van klasse B.",
            "Overschrijven en verzegelen van methode", MessageBoxButtons.OK,
            MessageBoxIcon.Information);
    }
}
```

void KlasseC.Boodschap()

'KlasseC.Boodschap()': cannot override inherited member 'KlasseB.Boodschap()' because it is sealed

Type classes

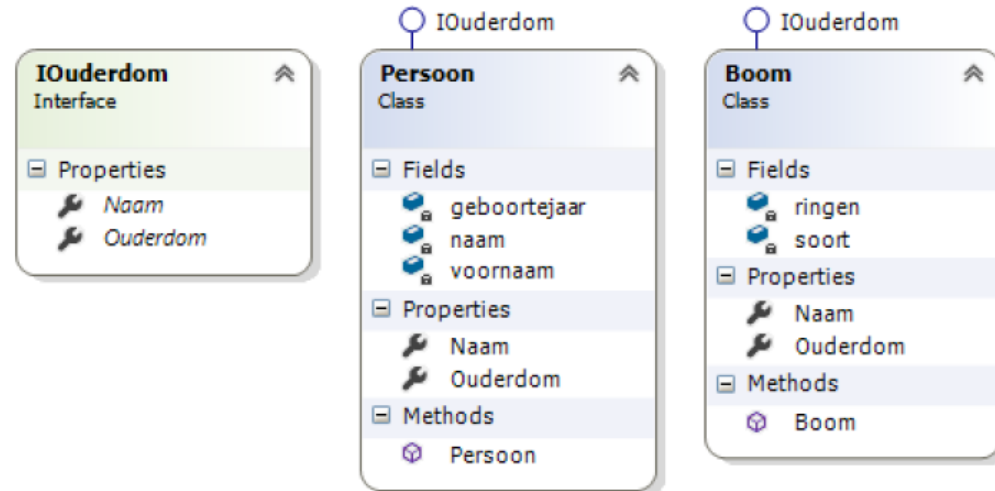
Type class	Normaal	Static	Sealed	Abstract
Kan geïntantieerd worden	Ja	Nee	Ja	Nee
Kan geërfd worden	Ja	Nee	Nee	Ja
Kan erven van anderen	Ja	Nee	Ja	Ja

Interface

- Definieert enkel de members (properties en methods)
- Geeft enkel een beschrijving, geen implementatie
- Soort blauwdruk voor de inherited classes
- Class die interface implementeerd moet **alles** implementeren zoals vastgelegd
- Grootste verschil met abstract class is dat deze laatste methods kan uitgewerkt hebben, een interface is **enkel** een beschrijving

Interface

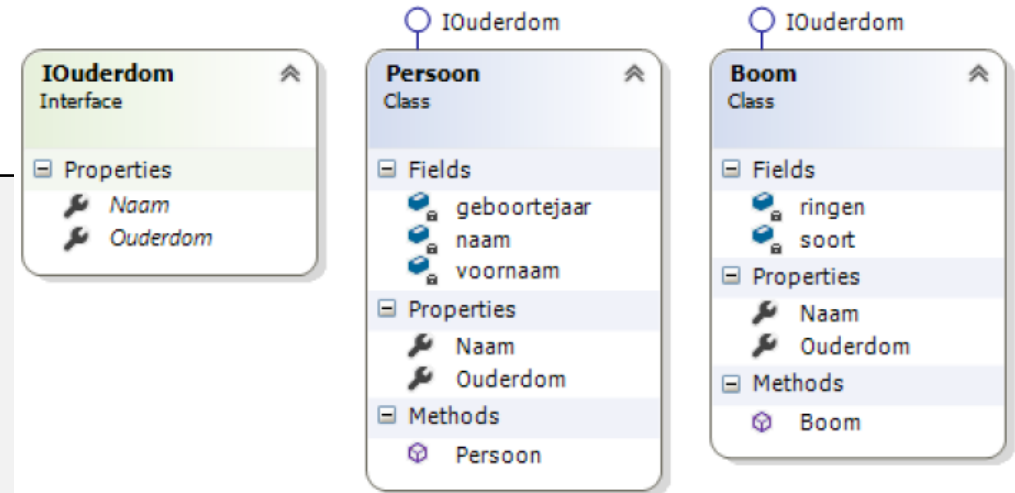
- Definieert enkel de members (properties en methods)



```
public interface IOuderdom
{
    // Klassen die de interface IOuderdom implementeren,
    // moeten de eigenschappen Ouderdom en Naam hebben.
    int Ouderdom {get;}
    string Naam {get;}
}
```

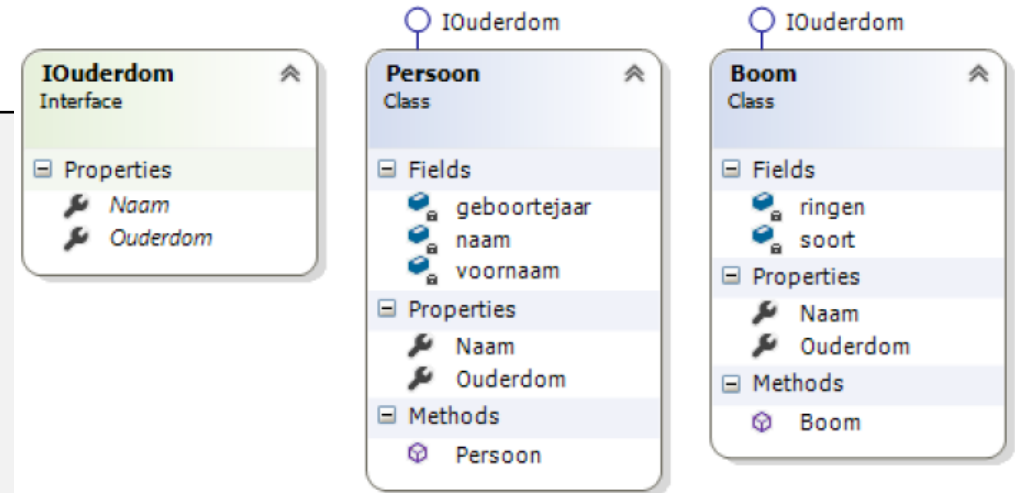

Interface

```
public class Boom : IOuderdom
{
    // Instantievariabelen.
    private int ringen;
    private string soort;
    // Constructor.
    public Boom(string type, int plantjaar)
    {
        ringen = DateTime.Now.Year - plantjaar;
        soort = type;
    }
    // Property Ouderdom - implementatie/interne werking van interface
    public int Ouderdom => ringen;
    // Property Naam - implementatie/interne werking van interface
    public string Naam => "Boom: " + soort;
}
```



Interface

```
public class Persoon : IOuderdom
{
    // Instantievariabelen.
    private int geboortejaar;
    private string voornaam;
    private string naam;
    // Constructor.
    public Persoon(string firstName, string familyName, int yearOfBirth)
    {
        voornaam = firstName;
        naam = familyName;
        geboortejaar = yearOfBirth;
    }
    // Property Ouderdom - implementatie van interface
    public int Ouderdom => DateTime.Now.Year - geboortejaar;
    // Property Naam - implementatie van interface
    public string Naam => voornaam + " " + naam;
}
```



Interface

```
private void BtnOuderdomPersoon_Click(object sender, EventArgs e)
{
    Persoon persoon = new Persoon(TxtVoornaam.Text, TxtFamiliennaam.Text,
    int.Parse(TxtGeboortejaar.Text));
    BerekenOuderdom(persoon);
}
private void BtnOuderdomBoom_Click(object sender, EventArgs e)
{
    Boom boom = new Boom(TxtSoort.Text, int.Parse(TxtPlantjaar.Text));
    BerekenOuderdom(boom);
}
public void BerekenOuderdom(IOuderdom age)
{
    MessageBox.Show((string) ("Ouderdom: " + age.Ouderdom), age.Naam,
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```