

leren. durven. doen.



# *C# FUNDAMENTALS*

STATIC  
GENERICs

# Inhoud

1. **Static Members van klasse**
2. **Statische klassen**



# Static sleutelwoord

## static versus Instantie members



# Static Members

- **Static members zijn geassocieerd met een type (class niveau) en niet met een instantie (object-niveau)**

Wordt aangeduid met **static** keyword bij declaratie

- **Static kan gebruikt worden bij**

Fields

Properties

Methods

Constructors

...



# Static vs. Non-Static

- **Static:**

Geassocieerd met type, niet met een instantie

- **Non-Static:**

Geassocieerd met een instantie

- **Static:**

Initialisatie gebeurt juist voordat het type voor de eerste keer wordt gebruikt

- **Non-Static:**

Initialisatie gebeurt wanneer de constructor wordt aangeroepen

# Static Members – Voorbeeld (1)

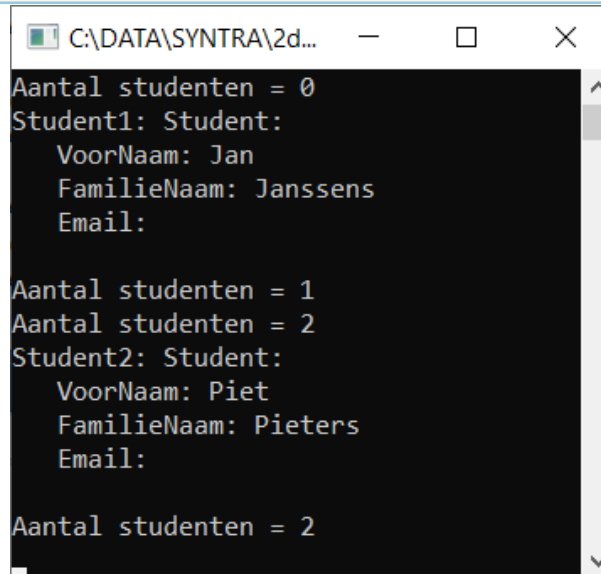


```
class Student
{
    private string voornaam;
    private string familienaaam;
    private string cursus;
    private Opleiding opleiding;
    private School school;
    private string email;
    private string telefoonNr;
    private static int aantalStudenten;
    public Student(string voornaam, string familienaaam, string cursus, Opleiding
opleiding=Opleiding.None, School school=School.None)
    {
        VoorNaam=voornaam;
        FamilieNaam=familienaaam;
        this.cursus =cursus;
        Opleiding = opleiding;
        this.School=school;
        Email =email;
        TelefoonNr = telefoonNr;
        aantalStudenten++;
    }
    public Student(string voornaam, string familienaaam):this(voornaam, familienaaam,null)
    {

    }
    public static int AantalStudenten {
        get { return aantalStudenten; }
    }
}
```

# Static Members – Voorbeeld (2)

```
public class Program
{
    public static void Main()
    {
        Console.WriteLine("Aantal studenten = " + Student.AantalStudenten);
        Student student1 = new Student("Jan", "Janssens");
        Console.WriteLine("Student1: " + student1.ToString());
        Console.WriteLine("Aantal studenten = " + Student.AantalStudenten);
        Student student2 = new Student("Piet", "Pieters");
        Console.WriteLine("Aantal studenten = " + Student.AantalStudenten);
        Console.WriteLine("Student2: " + student2.ToString());
        Console.WriteLine("Aantal studenten = " + Student.AantalStudenten);
    }
}
```



```
C:\DATA\SYNTRA\2d...
Aantal studenten = 0
Student1: Student:
  VoorNaam: Jan
  FamilieNaam: Janssens
  Email:

Aantal studenten = 1
Aantal studenten = 2
Student2: Student:
  VoorNaam: Piet
  FamilieNaam: Pieters
  Email:

Aantal studenten = 2
```





# Static Members

## Demo





# Static Members – Oefening

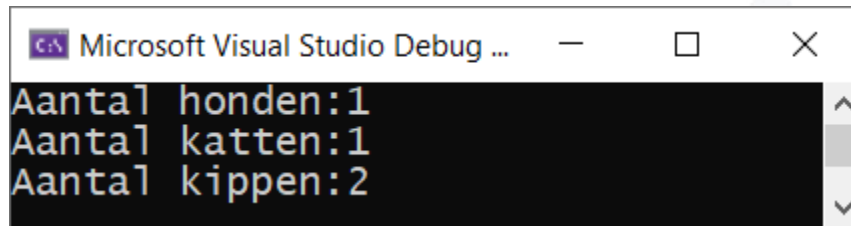
- **Neem het codevoorbeeld van de kinderboerderij uit de slides van Polymorfisme.**
- **Maak een public static property Aantal die enkel private schrijftoegang heeft die de waarde van het static field teruggeeft**
- **Voorzie een constructor aan elke afgeleide klasse van Dier die het static field aantal incrementeert**
- **Test met de code op volgende slide:**



# Static Members – Oefening (2)

```
class Program
{
    static void Main()
    {
        KinderBoerderij boerderij = new KinderBoerderij("Vierkantshoeve Gijzenzele");
        boerderij.VoegDierToe(new Dier() { Naam = "Beestje", Kleur = "zwart" });
        boerderij.VoegDierToe(new Hond() { Naam="Bobby", Ras="Labrador"});
        boerderij.VoegDierToe(new Kat() { Naam = "Minou", AantalMuizenPerDag=2});
        boerderij.VoegDierToe(new Kip() { Naam = "Sofie", EierenPerDag = 2 });
        boerderij.VoegDierToe(new Kip() { Naam = "Claire", EierenPerDag = 1 });

        Console.WriteLine("Aantal honden:" + Hond.Aantal);
        Console.WriteLine("Aantal katten:" + Kat.Aantal);
        Console.WriteLine("Aantal kippen:" + Kip.Aantal);
        // boerderij.MaakBoerderijGeluiden();
    }
}
```



Microsoft Visual Studio Debug ...

```
Aantal honden:1
Aantal katten:1
Aantal kippen:2
```



# Statische klassen

- Indien een klasse enkel **static** members bevat, kan de klasse zelf ook **static** gemaakt worden. Je vermijdt hiermee dat er instanties worden aangemaakt van de klasse
- Wordt aangeduid met **static class** keywords bij declaratie
- Wordt o.a. gebruikt voor utility klassen die waarvan geen instanties moeten worden gemaakt



# Voorbeeld Statische klasse

```
public static class Utils
{
    public static string MaakConnectionString(string dataSource, string username, string password, string database)
    {
        string connectionString = "Data Source=" + dataSource + ";";
        connectionString += "User ID=" + username + ";";
        connectionString += "Password=" + password + ";";
        connectionString += "Initial Catalog=" + database;
        return connectionString;
    }
}

class Program
{
    static void Main()
    {
        Console.WriteLine(Utils.MaakConnectionString(".", "ikke", "geheim", "myDb"));
    }
}
```



# Static Utility klasse

## Demo





# Vragen?