

# 3.Methoden en access modifiers

H9. OOP



# Voorbeeld van methode in klasse

- Merk op dat we **static** niet gebruiken bij klassen (later wel, zie H3)

```
class Mens
{
    public void Praat()
    {
        Console.WriteLine("Ik ben een mens!");
    }
}
```

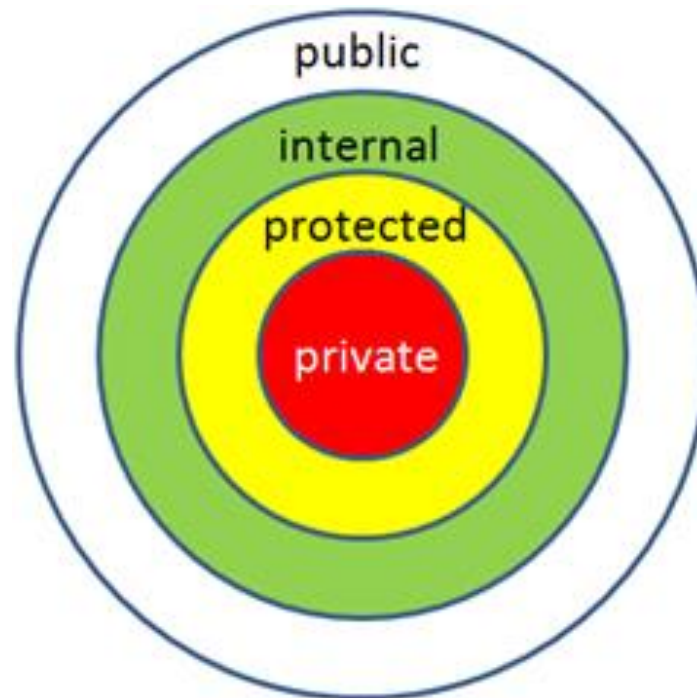
gebruik

```
Mens joske = new Mens();
Mens alfons = new Mens();

joske.Praat();
alfons.Praat();
```

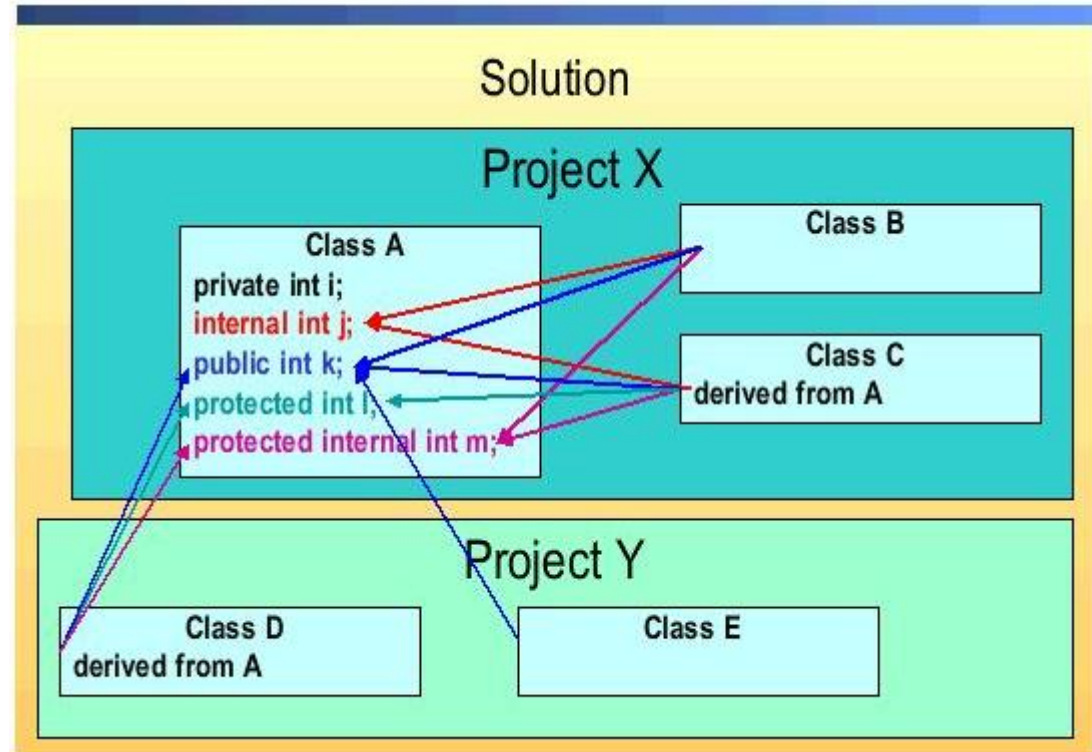
# Access modifier

- Keyword om aan te geven hoe 'zichtbaar' een methode of instantievariabele van een klasse is voor de buitenwereld:



# Access modifiers

- **Public:** overall zichtbaar
- **Internal:** enkel zichtbaar in assembly (project) van klasse
- **Protected:** in klasse zelf en klasse die van deze klasse overerven (zie later)
- **Private:** enkel in klasse zelf



# Public vs private

- Indien je geen access modifier voor method of field zet => **PRIVATE** (default)

```
class Gebouw
{
    private int aantalVerdiepingen;
    private string bouwMeester;
}
```

==

```
class Gebouw
{
    int aantalVerdiepingen;
    string bouwMeester;
}
```

- Maak enkel public indien dit echt moet.

# Naamgeving

- Private = start met kleine letter of underscore (\_)
- Public = start met hoofdletter

```
class Gebouw
{
    private int aantalVerdiepingen;
    private string _bouwMeester;

    public int BerekenKostPrijs()
    {
        return 2000;
    }
}
```

# Instantievariabelen NOOIT public

- **OPGELET:** instantievariabelen mogen NOOIT public gezet worden
  - C# laat het wel toe...maar het zit mee in de 'goto'-schemerzone
- **Gebruik properties of methoden indien je toegang tot instantievariabelen nodig hebt (zie verder)**



```
class Gebouw
{
    public int aantalVerdiepingen;
    public string bouwMeester;
}
```



```
class Gebouw
{
    private int aantalVerdiepingen;
    private string bouwMeester;
}
```

# Oldschool methoden

- 1 manier om aan instantievariabelen te komen is via GetX en GetY methoden

```
public class StaffMember
{
    private int age;
    public int GetAge()
    {
        return age;
    }
    public void SetAge( int inAge )
    {
        if ( (inAge > 0) && (inAge < 120) )
        {
            age = inAge;
        }
    }
}
```

```
StaffMember s = new StaffMember();
s.SetAge(21);
Console.WriteLine ( "Age is : " + s.GetAge() );
```

- Nog steeds gebruikt in Java
- In C# hebben we voorkeur voor **Properties** (zien we volgende les)



# Demo time

- Methoden en access modifiers

