

Referenz -> Superklasse -> Subklasse

```
preferences
public class Quadrat : Figur
{
    Properties and consts

    constructors (4 constructors available)

    public methods

private methods
}
```

Quadrat ERBT von Figur. Ein Quadrat ist also eine Figur.

Mit einer Referenz vom Typ Figur kann man auf ein Quadrat zeigen!



Referenz -> Superklasse -> Subklasse

```
9 references
public class Quadrat : Figur
{
    Properties and consts

    constructors (4 constructors available)

    public methods

    private methods
}
```

Quadrat ERBT von Figur. Ein Quadrat ist also eine Figur.

Mit einer Referenz vom Typ Quadrat kann man NICHT auf eine Figur zeigen!

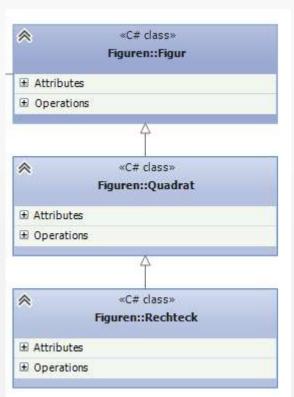
```
Quadrat andereReferenz;
andereReferenz = q1;
Console.WriteLine("ID: " + andereReferenz.ID);

// can not implicitly convert type "Figur" to "Quadrat"
andereReferenz = f1;
```



Beispiel: Flächenberechnung

Gegeben ist folgende Vererbungshierarchie:



```
public virtual double Flaeche()
   // Notlösung, denn eine (abstrakte)
                                                    virtual
   // Figur hat keine Fläche!
    return 0;
public override double Flaeche()
    // das Verhalten der Methode der
                                                   override
    // Basis-Klasse wird überschrieben!!!
   return s * s;
public override double Flaeche()
    // das Verhalten der Methode der
                                                   override
    // Basis-Klasse wird überschrieben!!!
    return s * s1;
```