



# PROPERTIES / EIGENSCHAFTEN

---

Vincent Uhlmann  
IT-Akademie Dr. Heuer GmbH

# PROPERTIES / EIGENSCHAFTEN

- Properties sind Mitgliedsvariablen einer Klasse mit speziellen Zugriffsmethoden
- Bieten Zugriff auf private Variablen mit Kontrolle über Lese- und Schreibzugriff
- Getter: Liefert den Wert der Property zurück
- Setter: Setzt den Wert der Property

```
public class Person
{
    private string _name;
    public string Name
    {
        get { return _name; }
        set { _name = value; }
    }
}
```

# GETTER

- Der Getter ist eine Methode, die den Wert einer Eigenschaft zurückgibt
- Berechnungen oder Formatierungen während des Lesens
- Ausdruckstextdefinitionen
- Erlaubt eine verkürzte Syntax für den Getter

```
public string FullName => $"{Forname} {Surname}";
```

# SETTER

- Der Setter ist eine Methode, die den Wert einer Eigenschaft setzt
- Validierungen oder Transformationen während des Setzens
- Das value-Schlüsselwort wird verwendet, um den Wert zu definieren, der von der set-Zugriffsmethode zugewiesen wird

```
public class Person
{
    private int _age;
    public int Age
    {
        get { return _age; }
        set {
            if(value > 0 && value < 120) {
                _age = value;
            }
        }
    }
}
```

# AUTOMATISCH IMPLEMENTIERTE EIGENSCHAFTEN

- Vereinfachte Syntax für den schnellen Zugriff auf Eigenschaften
- Der Compiler generiert automatisch einen privaten Feldspeicher
- Vorteile
  - Weniger Boilerplate-Code
  - Einfache Lesbarkeit und Wartung
- Nachteile
  - Begrenzte Kontrolle über Getter und Setter
  - Nicht geeignet für Validierung oder komplexe Logik

```
public class Person
{
    public string Name { get; set; }
    public int Age { get; set; }
}
```



# OBJEKTINITIALISIERER

---

Vincent Uhlmann  
IT-Akademie Dr. Heuer GmbH

# OBJEKTINITIALISIERER

- Objektinitialisierer ermöglichen das Zuweisen von Werten zu Attributen oder Eigenschaften während der Objekterstellung, ohne den Konstruktor aufzurufen
- Flexibilität bei der Auswahl, welche Attribute/Eigenschaften initialisiert werden sollen
- Kann mit parametrisierten Konstruktoren kombiniert werden
- Verwendung von geschweiften Klammern

# OBJEKTINITIALISIERER

```
class Person
{
    public string Name { get; set; }
}

static void Main()
{
    Person person = new Person { Name = "Peter" };
}
```



# OBJEKTINITIALISIERER

```
class Person
{
    public string Name { get; private set; }
    public int Age { get; set; }

    public Person(string name)
    {
        Name = name;
    }
}

static void Main()
{
    Person person = new Person("Hans") { Age = 10 };
}
```