OBJEKTORIENTIERUNG

Vincent Uhlmann
IT-Akademie Dr. Heuer GmbH

WAS SIND KLASSEN?

- Bauplan für Objekte (Referenztypen)
  - Definiert Struktur und Verhalten von Objekten
  - Analogien: Schablone, Architekturzeichnung, etc.
- Wiederverwendbarkeit
  - Kann für viele Objekte gleichen Typs verwendet werden
- Objektunterschiede
  - Unterschiedliche Werte für Eigenschaften wie Name, Farbe, Größe, Betrag, etc.

WAS IST EIN OBJEKT?

- Wird aus einem Bauplan (Klasse) erstellt
- Wird als Instanz bezeichnet
- Analogie: Ein konkreter VW Golf ist ein Objekt von der Klasse Car

KLASSE ANLEGEN

- Schlüsselword class
  - Klassenname beginnt immer mit einem großen Buchstaben
  - Klassenname ist immer Singular

- Klassendatei
  - Jede Klasse in einer eigenen Datei
  - Klassenname entspricht auch dem Dateinamen
  - Klassenname kann über mehrere Dateien verteilt werden (Schlüsselwort partial)

KLASSE ANLEGEN

- Eigenschaften (Attribute)
  - Beschreiben charakteristische Merkmale aller Objekte einer Klasse
  - Beispiel: Hersteller, Modell, Anzahl Räder

- Methoden (Verhalten)
  - Beschreiben, was ein Objekt tun kann
  - Beispiel: Fahren(), Ausgeben(), etc.

KLASSE ANLEGEN

```
public class Car
{
    public string Name;
    public int HorsePower;

    public string AsString()
    {
        return $"Name: {Name} HorsePower: {HorsePower}";
    }
}
```

KLASSE PARTIAL ANLEGEN

- Es ist möglich, die Definition einer Klasse in zwei oder mehr Dateien aufzuteilen
- Jede Datei enthält einen Abschnitt der Typdefinition
- Die Teile werden während der Kompilierung zusammengefügt
- Auch für Strukturen, Schnittstellen oder Methoden gültig
- Die Definitionen müssen hierfür denselben Namespace haben

KLASSE PARTIAL ANLEGEN

```
public partial class Date
{
    public int Day;
    public int Month;
    public int Year;
}

public partial class Date
{
    public string AsString()
    {
        return $"Day {Day} Month {Month} Year {Year}";
    }
}
```

OBJEKT ANLEGEN

- Klassenname als Datentyp
  - Klassenname wird zu einem Datentyp wie int oder string

- new Operator
  - Reserviert Speicher für das Objekt
  - Beispiel: Car golf = new Car();

- Objektvariable
  - Verweist auf den Speicherbereich, in dem das Objekt gespeichert ist
  - Daher Objekt-Referenz oder einfach nur Referenz

OBJEKT ANLEGEN

```
Car vw = new Car();
vw.Name = "Golf";
vw.HorsePower = 105;

Console.WriteLine(vw.AsString());
```

OBJEKT ANLEGEN

- Undefinierter Inhalt
  - Wird einer Objekt-Referenz kein Objekt zugewiesen, ist der Inhalt "nicht definiert" (lokale Variable) oder null (Objekt-Eigenschaft), nicht 0
- Mehrere Referenzen auf ein Objekt
  - Mehrere Referenzen können auf dasselbe Objekt verweisen
- Löschen einer Referenz
  - Zum "Löschen" einer Referenz kann diese auf null gesetzt werden
  - Beispiel: golf = null;

OBJEKT ANLEGEN

```
Car vw1 = new Car();
Car vw2 = vw1;
Car vw3 = vw2;

vw3 = null;

if(vw1 == null){
    Console.WriteLine("VW1 ist null");
} else {
    Console.WriteLine("VW1 ist nicht null");
}
```