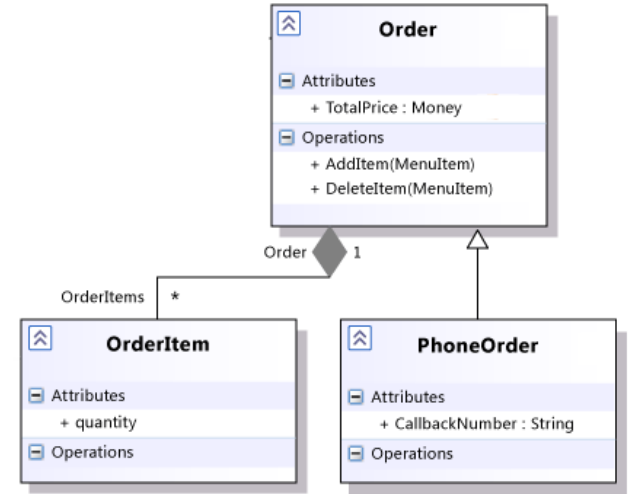


Objektorientierte Programmierung

Einführung



Lernziele

- Die Lernenden können in eigenen Worten erklären, was ein Objekt ist.
- Die Lernenden können in eigenen Worten den Unterschied zwischen einer Klasse und einem Objekt erklären.

Was ist ein Objekt?



object

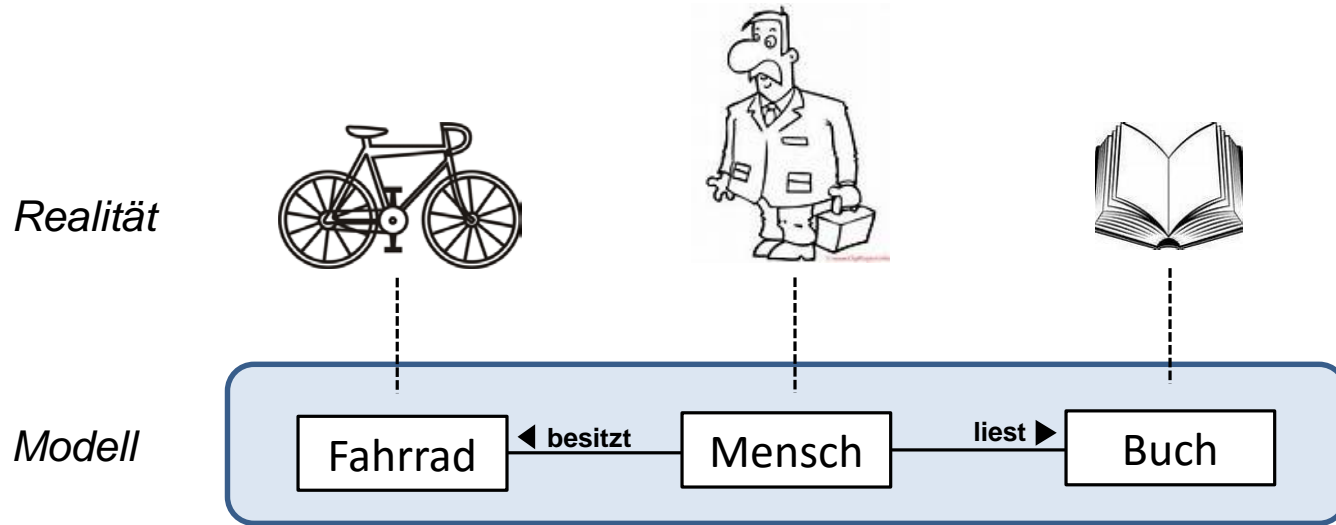


Was ist objektorientierte Programmierung?



Objektorientierte Programmierung

Abstraktion in der Softwareentwicklung



Objektorientierung

Was sagen die Experten?

Wikipedia: *Unter Objektorientierung, kurz OO, versteht man eine Sichtweise auf komplexe Systeme, bei der ein System durch das Zusammenspiel kooperierender Objekte beschrieben wird.*

ISO/IEC 2382-15 Standard: *Objektorientierung bezieht sich auf eine Technik oder Programmiersprache welche Objekte, Klassen und Vererbung unterstützt.*

Gruppenarbeit:

OOP Begriffe (er)klären



Objekt

Klasse

Eigenschaft

Methode

Instanziierung

Instanz

Konstruktor

Destruktor

Vererbung

Kapselung

Fürs Erste gilt



*Ein Objekt **ist eindeutig** identifizierbar (ist ein Unikat), **hat Eigenschaften**, die seinen Zustand bestimmen und verhält sich auf eine bestimmte Art und Weise indem es **Methoden ausführt**.*

```
public class Baum
```

```
{
```

```
    int _hoehe;
```

Konstruktor

```
1 reference
```

```
    public Baum(int hoehe)
```

```
    {
```

```
        _hoehe = hoehe;
```

```
    }
```

Eigenschaft

```
1 reference
```

```
    public int Hoehe
```

```
    {
```

```
        get { return _hoehe; }
```

```
        set { _hoehe = value; }
```

```
    }
```

```
1 reference
```

```
    public void Wachse(int Zuwachs)
```


```
    {
```

```
        _hoehe += Zuwachs;
```

```
    }
```

```
}
```

Methode

z.Bsp. C# 

```
Baum b = new Baum(20);  
b.Wachse(5);  
int h = b.Hoehe; // h = ????
```

Wie hoch ist der Baum nach
Aufruf der Methode Wachse () ?

Vorteile der Objektorientierten Programmierung?



Demo (KontoSample)

Eine Klasse in Aktion

- Eigenschaften
- Methoden
- Kapselung



Vorteile OOP

Wiederverwendbarkeit: Klassen modularisieren eine Anwendung in unabhängige Einheiten. Sie verwalten zusammengehörende Daten und gruppieren ähnliche Methoden. Klassen können von verschiedenen Programmen verwendet werden.

Wartungsaufwand: Klassen kapseln ihre Daten und ihre interne Logik (Methoden) gegenüber der Aussenwelt und können daher problemlos an neue Anforderungen angepasst werden. Programme, welche die Klasse nutzen, merken davon nichts. Jede Klasse kann als eigene, unabhängige Einheit getestet werden. Damit ist es ausreichend, die Klassenimplementierung nur einmal ausgiebig zu testen. Verläuft der Test positiv, wird die Klasse mit jeder anderen Anwendung ebenfalls korrekt zusammenarbeiten

Fehleranfälligkeit: Die Kapselung bewirkt, dass Objekte den Zugriff auf die Daten selbst kontrollieren und eine fehlerhafte Datenmanipulation und die daraus resultierende Inkonsistenz abwehren können.