

OOP

Objektorientierte Programmierung

Programmierstile

- Prozedurale Programmierung
- Strukturierte Programmierung (Funktionsbasierend)
- Objektorientierte Programmierung (OOP)

Vorteile der Objektorientierten Programmierung

- Code ist besser organisiert/ strukturiert, da alle Elemente auf der Seite kleine eigenständige Applikationen sind. Dies gilt vor allem bei großen Projekten.
- Wiederverwendbarkeit im selben oder in anderen Projekten
- Updatefähigkeit
- Es kann besser an unterschiedlichen Bereichen gleichzeitig gearbeitet werden, ohne Codeblöcke zu zerstören/überschreiben (Team programming)

Nachteile

- Bei kleinen Projekten muss in den meisten Fällen mehr Code geschrieben werden.
- Unter PHP kommt es zu Performance-Einbußen.

Qualität von Programmen

- Zweckerfüllung
- Bedienbarkeit
- Effizienz des Programmablaufs
- Zuverlässigkeit
- Wartbarkeit

Programmerstellung und Wartung

- Wichtigste Schritte in Softwareentwicklungsprozessen
 1. Entwurf
 2. Implementierung
 3. Validierung
- Ständige Anpassung von Entwicklungsprozessen wichtig!
- Wartung ist ein wesentlicher Kostenfaktor

Begriffe

- **Objekt**

- **Klasse** (Entwurfsmuster, Bauplan)

- **Methode**

Funktion die einem Objekt angehört

- **Eigenschaften** (Instanzvariablen, Attribut, Member)

Jedes Objekt besitzt eine oder mehrere Eigenschaften, die den Zustand oder die Zustände festlegen.

Zusammenfassung

Ein Objekt ist eine Instanz einer Klasse, in welcher Methoden und Eigenschaften festgelegt werden.

Vererbung

Ableitung neuer Klassen aus existierenden Klassen

Änderungsmöglichkeiten:

- Erweiterung um Methoden und Variablen
- Überschreiben von Methoden

Sichtbarkeit (Datenkapselung)

Sowohl Eigenschaften als auch Methoden kann eine Sichtbarkeitsstufe zugewiesen werden.

- **public** (Kompletter Zugriff)
- **protected** (Nur die eigene und geerbte Klassen haben Zugriff)
- **private** (Zugriff nur innerhalb der Klasse)

Finale Klassen/Methoden

Finale Klassen/Methoden bieten die Möglichkeit **Vererbung zu kontrollieren**, indem einzelne Methoden oder die ganze Klasse gesperrt werden kann.

Einsatzgebiet:

Wenn eine Methode/ Klasse Logik enthält die für einen reibungslosen Ablauf notwendig ist.

Fehlermeldung:

Fatal error: Class ChildClass may not inherit from final class (parentClass)

Abstrakte Klassen

- Dienen als Baupläne
- Ermöglichen die **kontrollierte Vererbung**
- Aus Abstrakten Klassen kann **keine Instanz** gebildet werden
- Wird eine als Abstrakt deklarierte Methode nicht Implementiert kommt es zu einem **Fehler**
- Sobald eine Klasse eine abstrakte Methode enthält gilt diese als abstrakt.

Interfaces / Schnittstellen

- Interfaces ermöglichen es zu spezifizieren, welche Methoden eine Klasse enthalten muss um dem Interface / der Schnittstelle zu entsprechen.
- Alle Methoden eines Interfaces müssen *public* sein
- Ein Interface kann von anderen Interfaces erben.
- Im Gegensatz zu abstrakten Klassen handelt es sich nicht um einen Bauplan sondern um eine Überprüfung

Konstruktor und Destruktoren

Konstruktor

werden bei Instanziierung einer Klasse automatisch gerufen

Destruktoren

werden bei Ende der Lebensdauer eines Objektes automatisch gerufen.

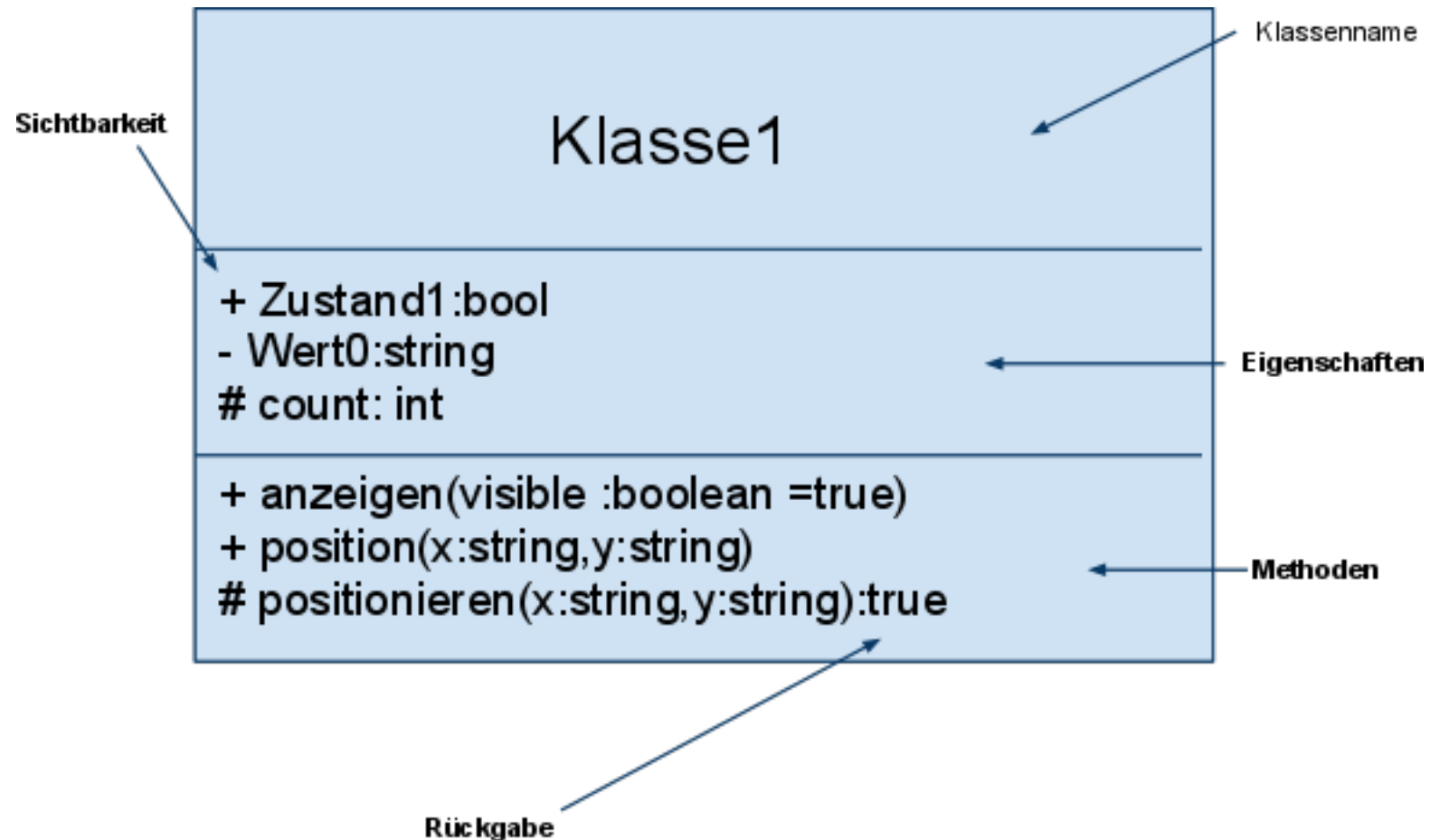
UML (Unified Modelling Language)

- Ein Standard der OMG (<http://www.omg.org/uml>)
- Grafische Modellierungssprache
- Definiert eine Notation und Semantik zur Visualisierung, Konstruktion und Dokumentation von Modellen
- Es existieren 14 verschiedene Diagramm-typen

Editoren :

- <http://www.gliffy.com/> (online)
- ArgoUML

Eine Klasse darstellen



Assoziationen / Beziehungen

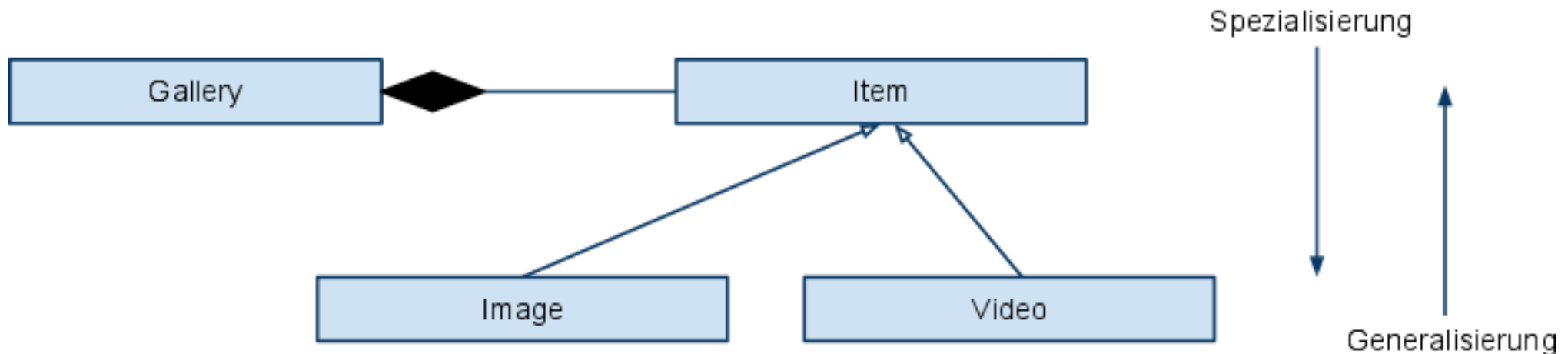
- Aggregation



- Composition



- Generalisierung & Spezialisierung



Multiplizität

Ein Blogeintrag hat mindestens 0 Maximal unendlich viele Kommentare.



Eine Kategorie benötigt mindestens 3 Bilder um angezeigt zu Werden, darf aber nicht mehr als 1000 beinhalten.

