**Default Werte von Funktionen werden**

* Im Funktionsprototyp angegeben, wenn er existiert. Dann dürfen die Werte nicht in der Funktionsdefinition stehen.
* In die Funktionsdefinition eingeschrieben, wenn es keinen Prototypen gibt.
* Nur von hinten weggelassen.

**Überladene Funktionen:**

Was sind überladene Funktionen?

Unterschiedliche Funktionen mit demselben Namen

Vorteile von überladenen Funktionen?

Mehrere Artverwandte Funktionen mit demselben Namen machen das Programm übersichtlicher, einfacher zu bedienen

Welche Funktion aufgerufen wird, wird anhand der Signatur entschieden, Anzahl, Typ der übergebenen Parameter

Dürfen sich nicht nur durch den Rückgabewert unterscheiden

Eindeutige Signatur achten, wird eine implizite Typkonversion durchgeführt, kann das u. U. zu falschen Ergebnissen führen.

Achtung bei Default Werten

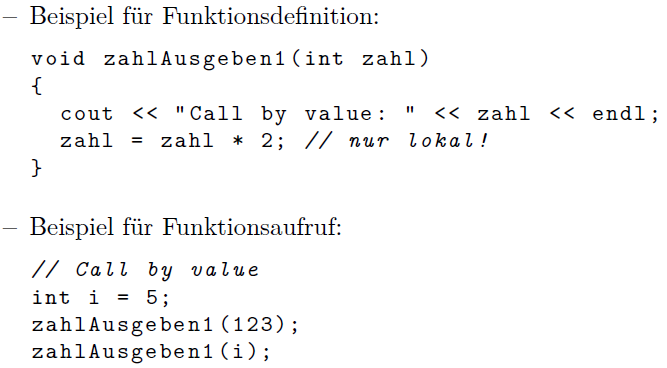
**Parameterübergabe an Funktionen:**

by value, by pointer, by reference, by const reference

Reference:

Werden mit & deklariert

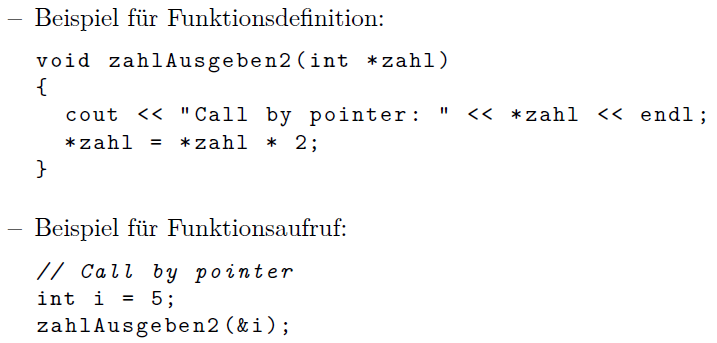
Müssen bei der Definition initialisiert werden -> Fest an eine Variable gebunden, kann nicht mehr verändert werden

**Call by value**

Einfach

Änderungen sind nur lokal wirksam

Kopieren von Werten ist aufwendig

**Call by pointer**

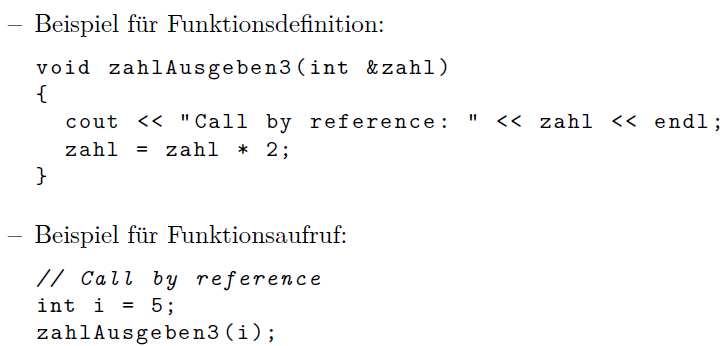
Änderungen auch außerhalb der Funktion wirksam, da Änderung des Werts an der Ardresse

Verwenden des AdressOperators

„Keine Übergabe“ als *nullptr*

Kein aufwändiges Kopieren

Adresse muss in Funktion dereferenziert werden



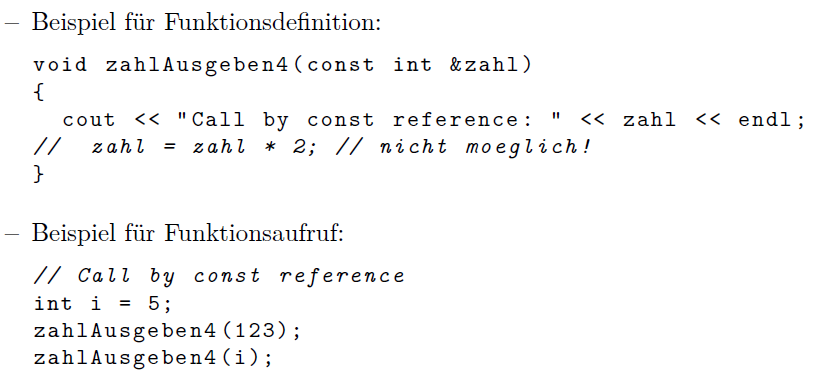
**Call by reference**

Änderung auch außerhalb der Funktion wirksam

Aufruf der Funktion nicht ersichtlich, ob Call by value, oder call by reference

Keine Dereferenzierung in Funktion nötig

Kein Kopieren

****

**Call by const reference**

Read-Only Übergabe

Keine Dereferenzierung nötig

Kein Kopieren nötig

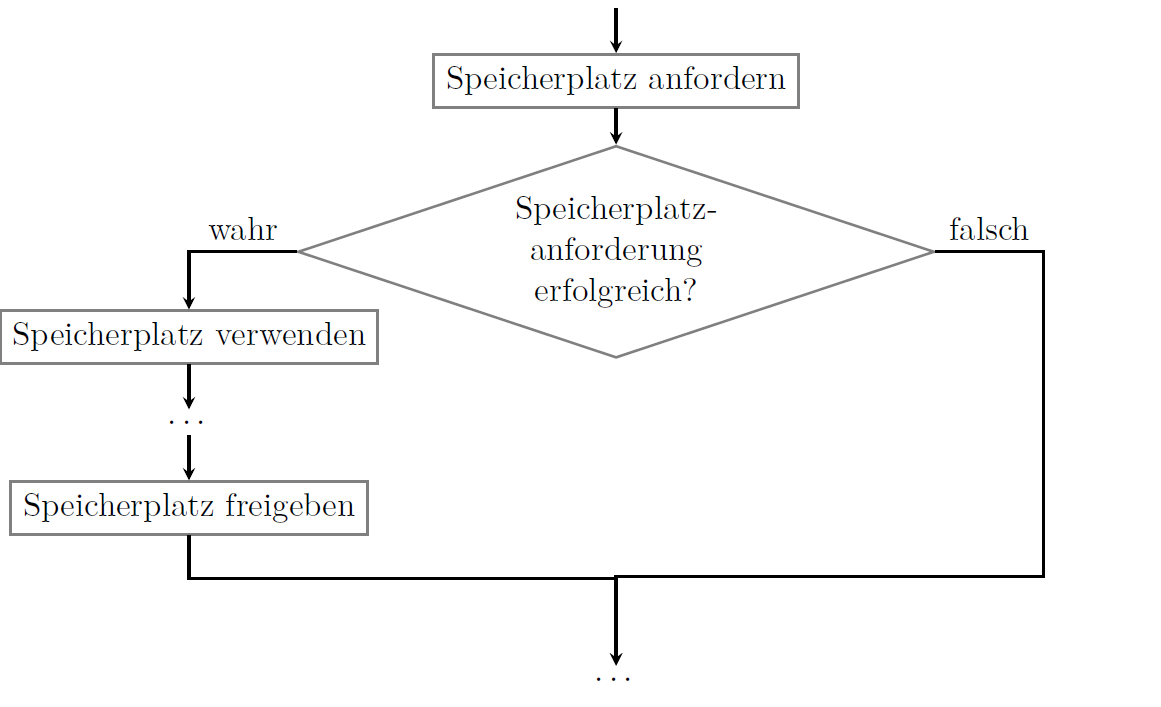
**Gründe für dynamische Speicherplatzreservierung:**

Ohne, sind Programme ineffizient, da Speicher im Voraus reserviert werden muss.

Speicherplatz bei Bedarf veränderbar

Kooperatives Prinzip: Speicherplatz wird an das System zurückgegeben, sobald weniger Speicherplatz benötigt wird.

Unkooperatives Prinzip: Programme können das System stark bremsen, wenn andere Programme auf virtuellen Speicher ausweichen müssen.

**Reservieren von Speicher**

double eineZahl = new double;

double zahlenfeld = new double[5]

Überprüfung, ob Speicher zugewiesen wurde mithilfe von Expections

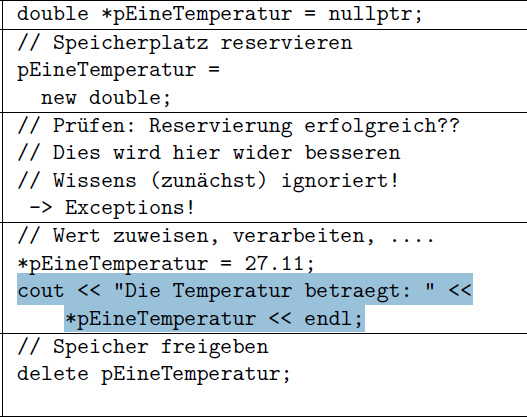
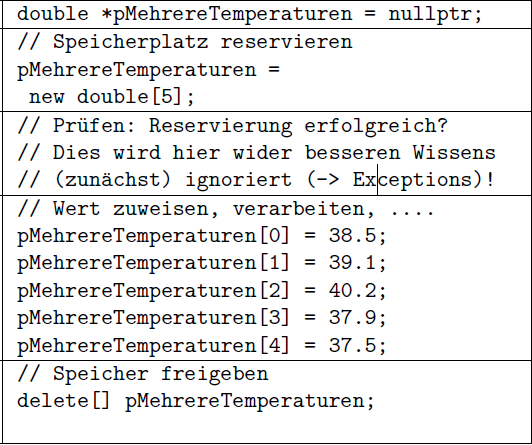
eineZahl = &zahl2;

\*eineZahl = 5;

zahlenfeld[0] = 5;

delete eineZahl;

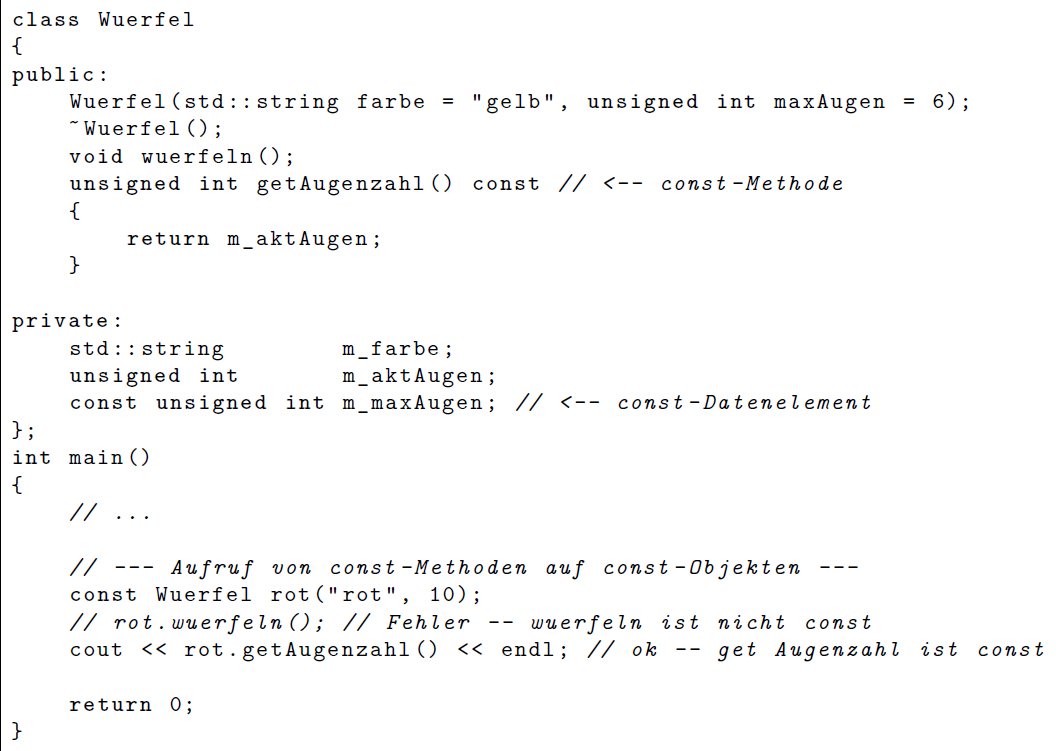
delete[] zahlenfeld;

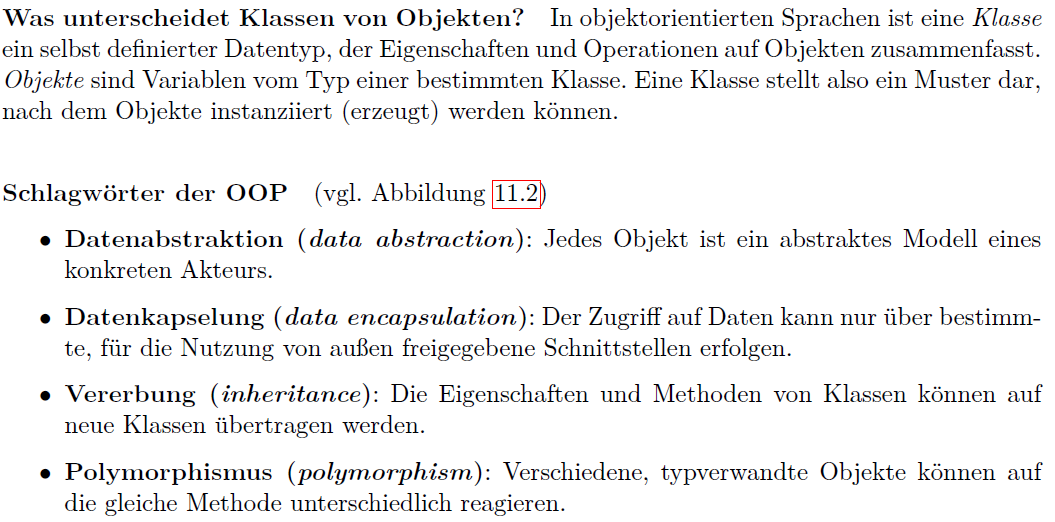


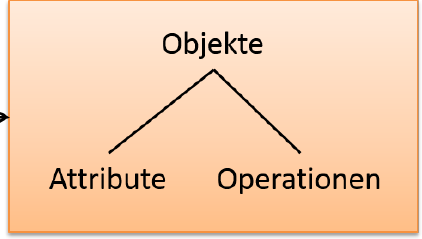
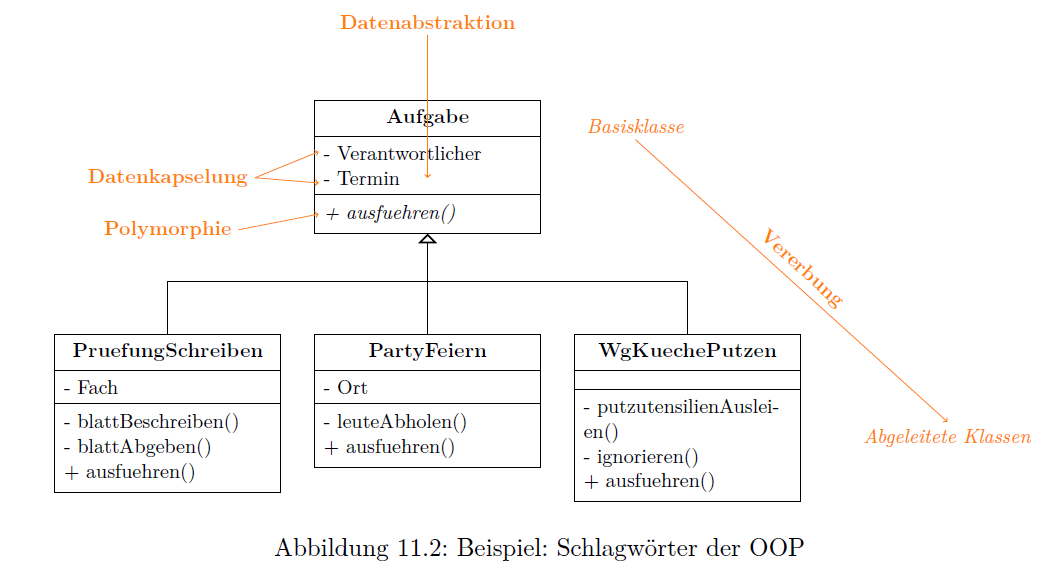
**const**

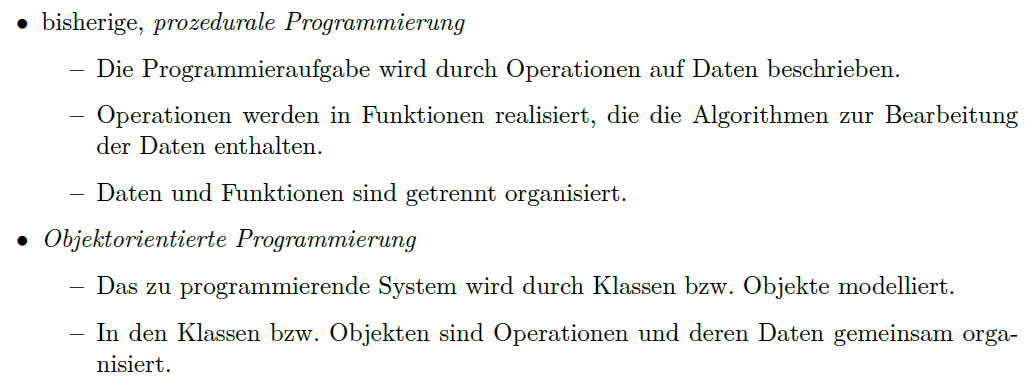
const Attribute werden in der Initialisierungsliste definiert

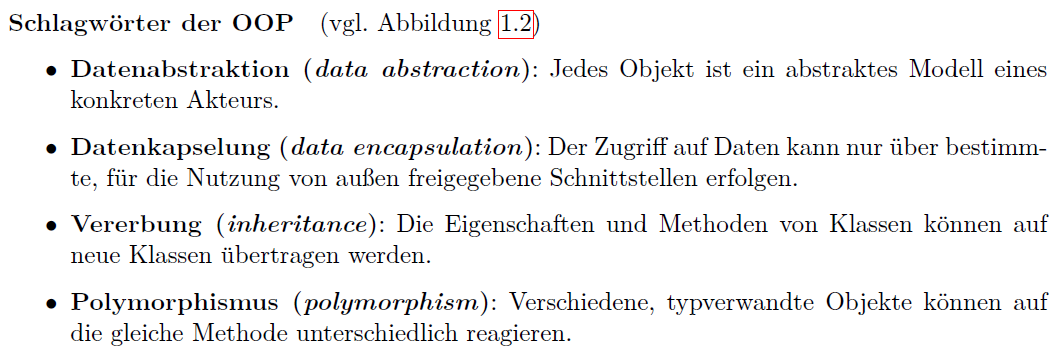
Konstante Objekte oder übgabe als const reference kann nur const deklarierte Funtionen ausführen

Methoden, die keine Daten verändern, sollen als const deklariert werden, nur dann können auch const Objekte die Funktion aufrufen

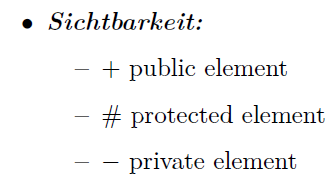
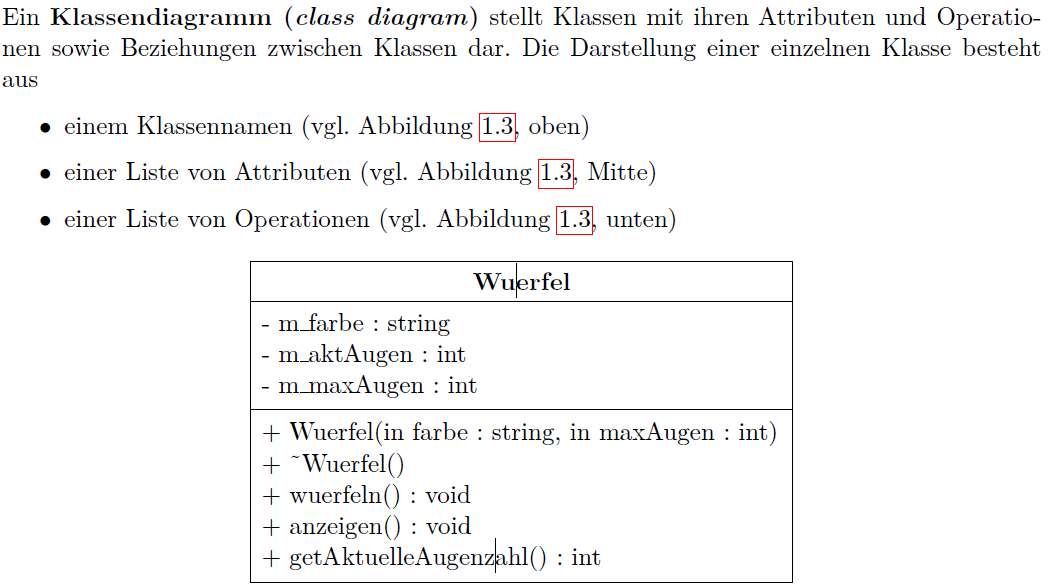








UML Klassendiagramm

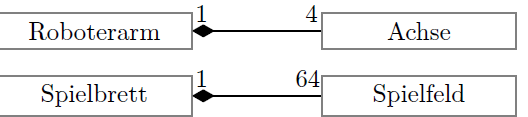


Abstrakte Klassen und Methoden werden kursiv gesetzt

Attribute und Methoden, die „statisch“ zu einer Klasse gehören, werden unterstrichen



Komposition:

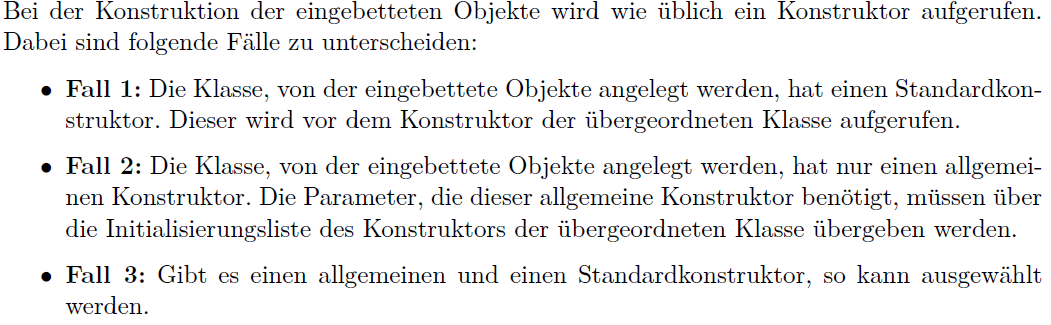
Setzt Objekte mehrere Klassen in Beziehung durch „besteht aus“

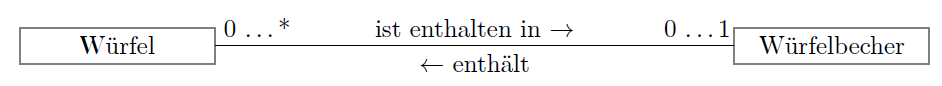
Sie sind existenzabhängig.

Lebensdauer von dem exisitenzabhängigem Teil ist mit Ganzen gekoppelt

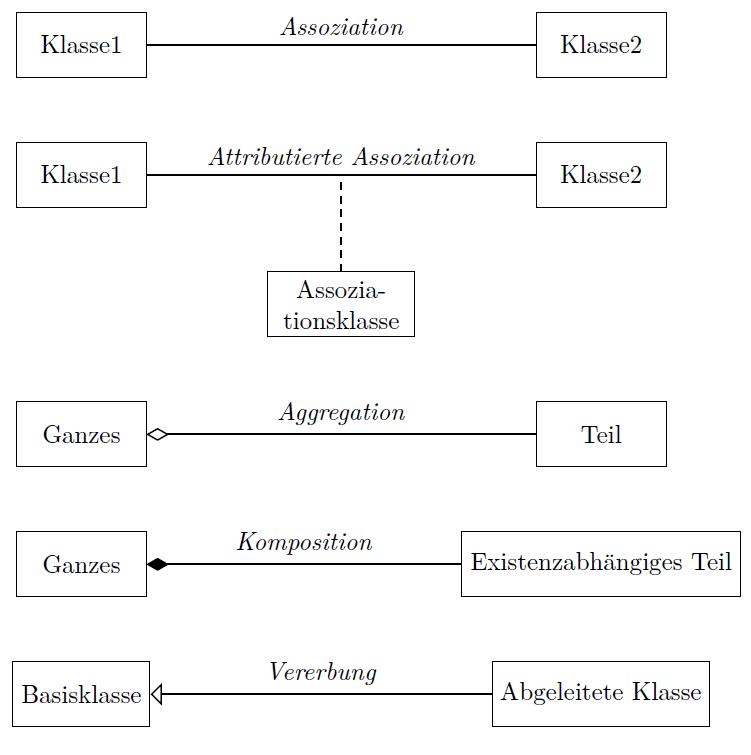
Jedes Teilobjekt darf zu jedem Zeitpunkt nur zu einem Ganzen gehören, ist Verantwortlich für das Erzeugen der Teile.

Die Konstruktionsfolge ist von innen nach außen, Die Destruktionsfolge von außen nach innen





Assoziation:



Sehr schwache Beziehung

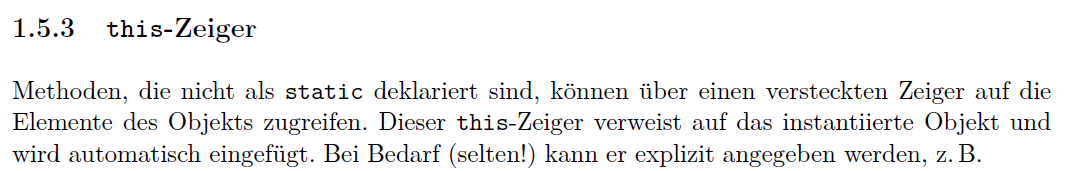
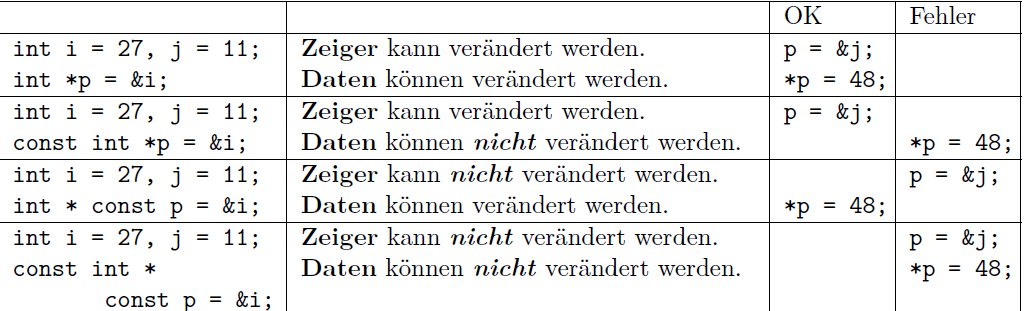
Kann während der Laufzeit auf und abgebaut werden.

Eigenschaften(Assoziationsname, Multiplizität, Richtung)

Ein Würfel kann sich in höchstens einem Wuerfelbecher befinden

Realisierung mit einem Feld aus Zeigern, die die Adresse von Wuerfel aufnimmt.

TODO Vererbung



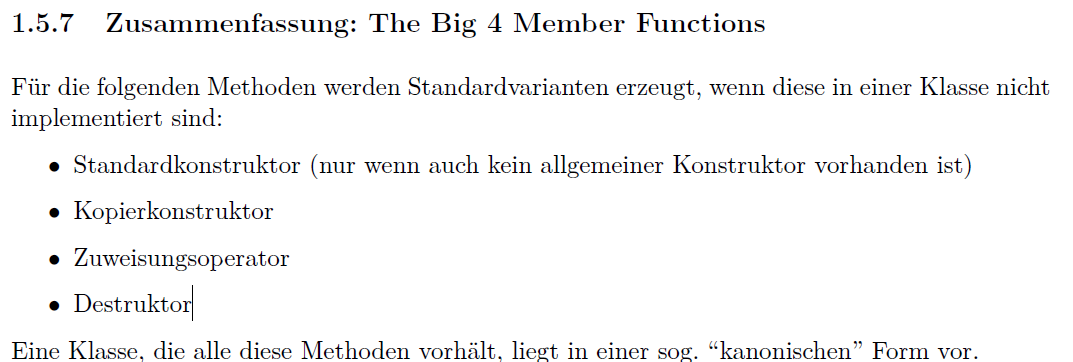
**Standard-Konstruktor**

Erwartet keine Übergabe oder hat für alle Werte Default-Werte, wird bei Bedarf vom Compiler automatisch erzeugt

**Allgemeiner Konstruktor**

Beliebige Parameterliste, welche oft überladen wird, wird aufgerufen, sobald man einen Wert dem Objekt übergibt

**TODO Kopierkonstruktor**



Vokabular:

Definition: Funktionsweise der Funktion

Deklaration: Funktionsdeklaration steht meist in der Header, Prototyp, StandardWerte stehen hier

Speicher zuweisen von Variablen

Rückgabewert: Rückgabewert von Funktion

Signatur: Variabeltypen, die eine Funktion erwartet

Objekt: Variablen vom Typ einer bestimmten Klasse

Klasse: Selbstdefinierter Datentyp, der Eigenschaften und Operationen auf Objekte zusammenfasst

Muster, wie ein Objekt erzeugt werden soll