

PGdP - Übung 9

Kira Nickel

- 1 Aufgabe 9.1
- 2 Aufgabe 9.2
- 3 Aufgabe 9.3
- 4 Aufgabe 9.4
- 5 Nützliche Dinge für die Hausaufgabe

Polymorphie

Info

Polymorphie könnte gut ein Thema in der Klausur sein

Statischer Typ

Typ, mit der eine Variable deklariert wurde

- Typ kann mit Casts (temporär) geändert werden

Signatur der Methode

Methodenname und Typen der Parameter mit denen sie aufgerufen wird

Polymorphie

Dynamischer Typ

Typ einer Variable, der erst zur Laufzeit bekannt ist (dieser kann vom statischen Typen abweichen)
- wird mit dem Begriff *new* erstellt

Superklasse

Alle Klassen, von denen die aktuelle Klasse erbt (in aufsteigender Reihenfolge)

Subklasse

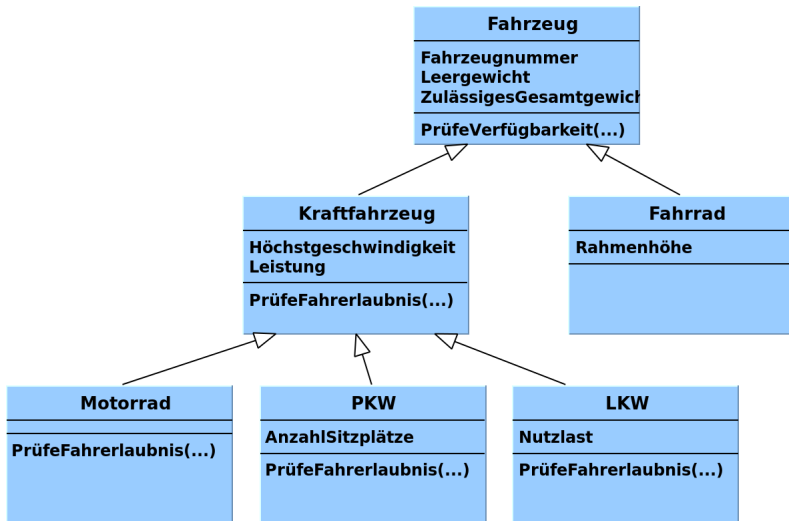
Alle Klassen, an die die aktuelle Klasse vererbt

Objektorientierte Programmierung

Vererbung

- Klassen erben von einer anderen Klasse durch Schlüsselwort *extends*
- **Subklassen** können public/protected Variablen oder Methoden der Superklasse aufrufen
 - Sie können eine Erweiterung oder Einschränkung der ursprünglichen Klasse sein
- Objekt einer Klasse wird über den **Konstruktor** der Klasse erstellt
$$A\ a = \text{new } A();$$
- Eine **Klassenhierarchie** beschreibt die Vererbung zwischen verschiedenen Klassen
- Methoden mit gleichem Namen und gleicher Signatur werden von einer Subklasse überschrieben

Beispiel für eine Klassenhierarchie



Visitor Pattern

Vorteile

Das Visitor Pattern erlaubt es, Funktionalitäten für verschiedene Klassen hinzuzufügen, ohne jede Klasse innerhalb einer Hierarchie ändern zu müssen. Dadurch erlaubt es sehr schnelle und einfache Änderungen

Visitor-Klasse

Bietet visit-Methode für alle gewünschten Klassen an

Visitor Pattern

TypeVisitor-Klasse

Bietet eine Funktionalität vom Typ `Type` für alle gewünschten Klassen an. Sie überschreibt außerdem die `visit`-Methoden der Klasse `Visitor`

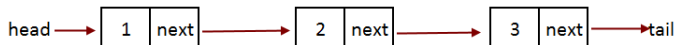
`accept`-Methode

- Wird in alle gewünschten Klassen eingefügt
- Bekommt ein Objekt der Klasse `Visitor` als Parameter
- Ruft die Methode `visit` der Klasse `Visitor` mit dem Objekt der Klasse (also sich selbst) mittels `this` auf

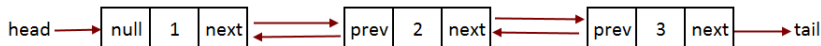
Doppelt verkettete Listen

Struktur

Jedes Element der Liste enthält sich selbst und einen Verweis auf das vorherige und nächste Element in der Liste.



Singly Linked List



Doubly Linked List

Links für Themen der Hausaufgaben

Beispiel und Erklärung Visitor Pattern

<https://www.youtube.com/watch?v=pL4mOUDi54o>

JavaDoc für Lists

Hier findet ihr alle Funktionen die euch die Klasse List bereits anbietet:

<https://docs.oracle.com/javase/8/docs/api/java/util/List.html>

JavaDoc für bitweise Operatoren

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/op3.html>

Alte Hausaufgaben aus der Besprechung Montag bzw. Samstag

Alternative Lösungen zu alten Hausaufgaben

Bitte Code aus diesen Klassen **nicht für Hausaufgaben verwenden!** Dies ist nur eine Hilfe für besseres Verständnis und um einen Ansatz zur Lösung von Hausaufgaben zu sehen.
<http://www.erikkynast.de/course/info1ws17/>