1. Was ist UML und für was wird es verwendet?

UML (Unified Modeling Language) ist eine standardisierte Modellierungsprache. Sie stellt geometrische Formen zur Verfügung, um Modelle von Software zu erstellen.

1. Welche verschiedenen Arten von Verhaltensdiagrammen gibt es?

Klassendiagramm  
Objektdiagramm (Objekte während der Laufzeit)  
Komponentendiagramm (nächsthöhere Ebene über den Klassen)  
Kompositionsstrukturdiagramm (Gegenstück zum Komponentendiagramm)  
Verteilungsdiagramm (Client-Server-Applikationen – verteilte Software)  
Paketdiagramm (Java oder C# Pakete)  
Use-Case Diagramm

Aktivitätsdiagramm (Ablauf von Aktivitäten

Zustandsautomat (beschreibt Zustandswechsel)

Sequenzdiagramm (Abläufen zw. Zum Beispiel Klassen oder Objekten)  
Kommunikationsdiagramm (ähnlich Sequenzdiagramm)  
Timing-Diagramm (wie viele Sekunden von einer Aktion auf eine andere folgt

Interaktionsübersichtsdiagramm (vereint mehrere Aktivitäts- und Sequenzdiagramme)

1. Was ist die das wichtigste Strukturdiagramm in UML und welche Aufgabe hat es?

Das Klassendiagramm ist mit Abstand das wichtigste Strukturdiagramm der UML. Es erklärt den Zusammenhang zwischen den Klassen und kommt dem Quellcode am nächsten.

1. Wie sind Klassendiagramme in Detail aufgebaut und welche Beziehungen zwischen den Klassen gibt es?

Siehe Folien Klassendiagramme im Detail!!!

1. Erkläre den Unterschied zwischen Aggretation und Komposition und Assoziation!  
   Eine Assoziation beschreibt eine Beziehung zwischen zwei oder mehr Klassen, im häufigsten Fall eine Verbindung zwischen genau zwei Klassen. Assoziationen definieren dabei eine Beziehung auf Typebene. Auf Instanzebene nennen sich die konkreten Ausprägungen einer Assoziation Link.

Aggregation = spezielle Assoziation

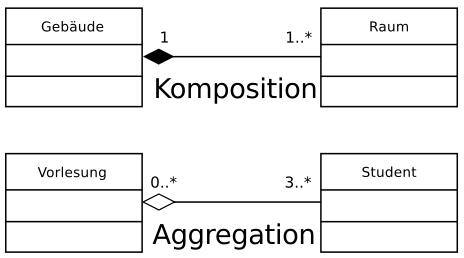
Stellt eine Teile-Ganzes-Beziehung dar

Komposition ist eine streng/spezielle Form der Aggregation mit folgenden Merkmalen:

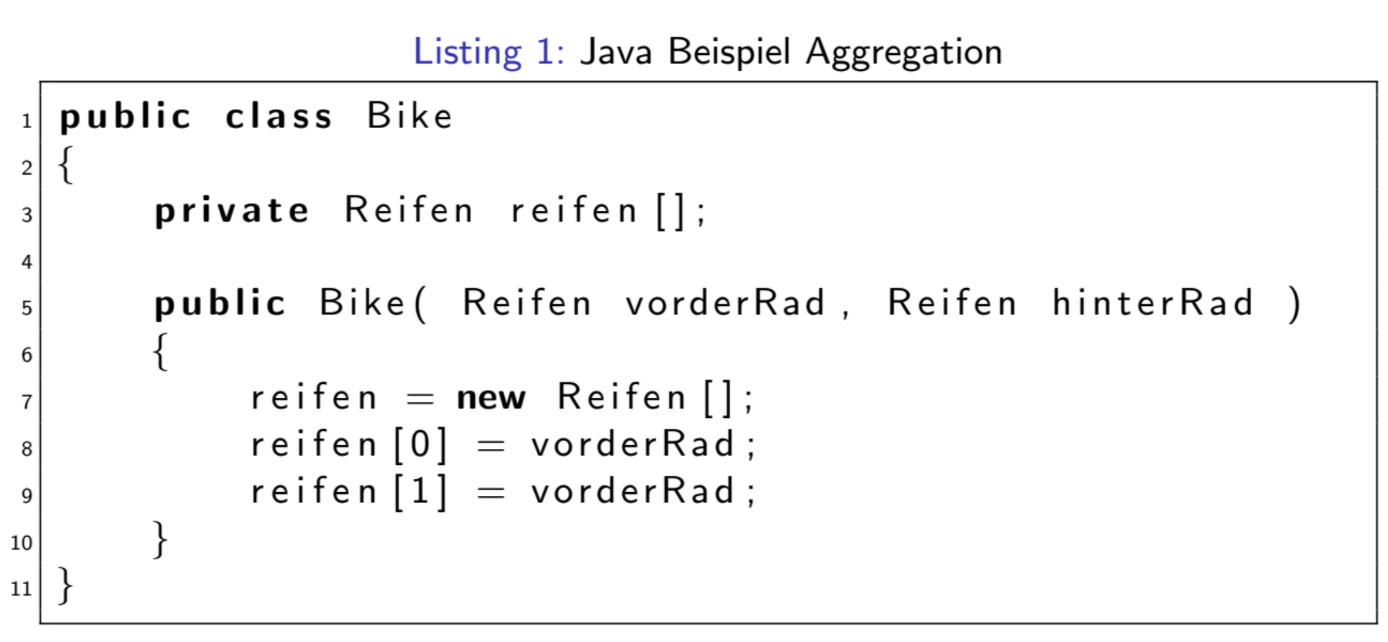
Ein Teil existiert nur solange wie sein Ganzes existiert(Existenzabhängigkeit)

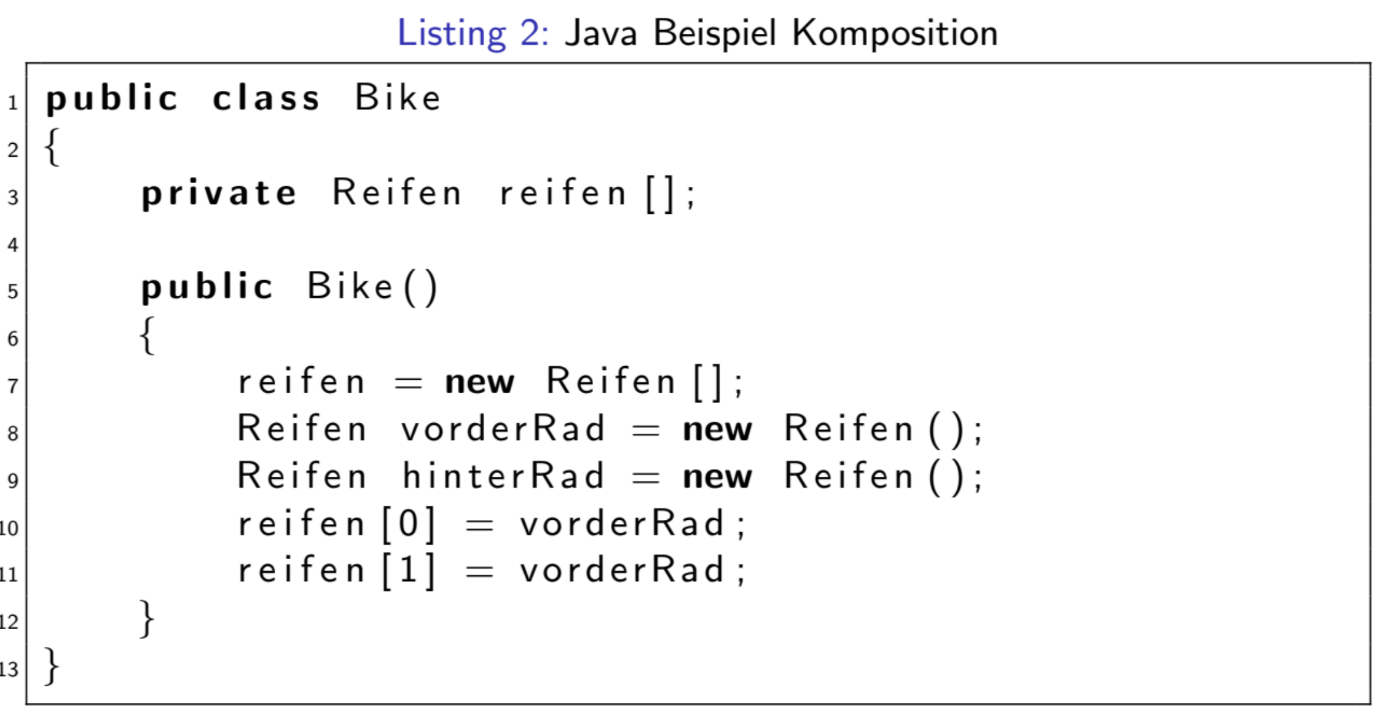
Ein Objekt kann immer nur Teil maximal eines Ganzen sein(Multiplizität 0…1

Oder 1). Also kann ein Objekt nicht Teil von mehreren verschiedenen Ganzen sein.



Raum kann nicht ohne Gebäude existieren aber Student kann ohne Vorlesung existieren.





1. Was ist SQL-Lite und warum sollte ich es verwendet?

SQLite ist eine Programmbibliothek, die mit einem eingebetteten Datenbanksystem arbeitet. Dadurch wird keine weitere Server-Software mehr benötigt und die Datenbank lässt sich problemlos in andere Applikationen integrieren.

SQL-Lite ist schneller und einfacher.

1. Vorteile-Nachteile SQL-Lite gegenüber MySQL!

Vorteile:

* Keine laufende Datenbankserveranwendung nötig
* Datenbanksystem vollständig in die Anwendung integrierbar
* Datenbankdatei lässt sich einfach sichern, austauschen und weitergeben
* Große Auswahl an verfügbaren Programmierschnittstellen
* Bedienung über SQL-Befehle

Nachteile:

* Fehlen der Benutzerrechtesteuerung machen den Einsatz im Mehrbenutzerbetrieb schwierig bis unmöglich
* es wird lediglich der veraltete SQL-92 Standard unterstützt
* nicht alle Möglichkeiten des SQL-92 Standards sind implementiert
  + Fremdschlüssel nur über den Umweg mit Triggern möglich
  + JOIN Funktionen nicht vollständig nutzbar
  + ALTER Funktionen nicht vollständig nutzbar

1. Welche wesentlichen Unterschiede gibt es zwischen MySQL und SQL-Lite?

SQL-Lite: Keine laufende Datenbankverbindung nötig

MySQL unterstützt viel mehr abgeleitete Datentypen (small Int z.B.)

SQLite transformiert alle angegebenen Datentypen in eine von 5 möglichen (z.B. small Int 🡪 Int), das bedeutet bei großen Datenbanken kann SQLite ineffizienter werden als MySQL.

Deshalb wird bei kleinen Datenbanken auch gerne SQLite verwendet da es dort sehr viel weniger Speicher als MySQL braucht.

1. Was sind Prepared Statements und Welchen Zweck haben Sie?

Ist eine vorbereitete Anweisung für ein Datenbanksystem. Im gegensatz zu gewöhnlichen Statements enthält es noch keine Parameterwerte. Stattdessen werden dem Datenbanksystem Platzhalter übergeben.

Mit Prepared Statements können SQL-Injections verhindert werden, da das Datenbaksystem die Gültigkeit von Parametern prüft, bevor diese verarbeitet werden

1. Was ist SQL Injection?
   1. Eine Technik welche von Hackern benutzt wird um Zugriff auf eine Datenbank zu erlangen.
   2. https://www.w3schools.com/sql/sql\_injection.asp
2. Was ist der grundlegende Aufbau von MySQL?
   1. MySQL ist eine Relationale Datenbank. Das heißt entweder es gibt eine Relation (Beziehung) zwischen zwei „Dingen“ oder nicht. Der Aufbau einer relationalen Datenbank sieht so aus:
   2. Sie besteht aus Tabellen.
      1. Diese Tabellen bestehen aus Zeilen und Spalten
         1. Die Zeilen stellen den sogenannten Datensatz dar
         2. Die Spalten beinhalten die einzelnen Komponenten eines Datensatzes
         3. Jede Spalte muss einen bestimmten Datentyp haben
      2. Über sogenannte Schlüssel werden die Tabellen mit einander verknüpft
3. Was ist ORM und wieso wird es verwendet?  
   *object-relational mapping, zu Deutsch Objektrelationale Abbildung, ist eine Technik, um Objekte, eines objektorientiert entwickelten Programms, in einer relationalen Datenbank abzubilden. Da objektorientierte Programmiersprachen und relationale Datenbanken grundlegend verschieden sind, wurden ORMs eingeführt um diesen Wiederspruch aufzuheben, jedoch wird bei Verwendung dieser die Datenbank nicht optimal genutzt, was Leistungseinbußen bedeutet.*
4. Was sind die Vorteile bzw. Nachteile von ORM gegenüber MySQL oder SQL-Lite?
   1. Vorteile:

* Man muss sich nicht mit Datenbankbefehlen auseinandersetzen 🡪 Grundvoraussetzung ist einfach simple Programmierung durch Einsatz der gegebenen Bausteine. DB’s bzw. Table werden durch das ORM generiert.
* Bei kleinen Projekten ist es sehr schnell zum erstellen
* (Vor/Nachteil) Spart sich Zeit um sich mit SQL-Statements auseinanderzusetzen
  1. Nachteile:
* Die Performance leidet je größer die Anwendung ist stark unter der zusätzlichen Ebene, die für das ORM notwendig sind.
* Bei sehr großen Datenbanken kann es leicht dazu kommen, dass man sich durch das ORM verwirrt 🡪 je größer desto unübersichtlicher.
* Bei Updates kann es oft zu Komplikationen kommen, da z.B. Version 7.7 des ORMs nicht 7.7 des „irgendwas“ unterstützt
* (Vor/Nachteil) Spart sich Zeit um sich mit SQL-Statements auseinanderzusetzen