

UML – Eine kurze Einführung

Programmierungsmethodik

Eva Zangerle
Universität Innsbruck

Handyvertrag.de: LTE 3GB Tarif

Für nur 4,99€¹ mtl. 3 GB LTE Highspeed - Wechselbonus - monatlich kündbar. J
sichern!

Handyvertrag.de: LTE 3GB Tarif

Für nur 4,99€¹ mtl. 3 GB LTE Highspeed - Wechselbonus - monatlich kündbar. J
sichern!

Überblick

Einführung

Java – Ein erster Überblick

Objektorientierung

Vererbung und Polymorphismus

Ausnahmebehandlung

Pakete und Javadoc

Spezielle Themen

Generische Programmierung

Java Collection-Framework

Streams

Unit-Tests

Entwurfsmuster - Eine Einführung

GUI-Programmierung

Java Virtual Machine

Ausblick



Allgemeine Grundlagen

Objekte und Klassen

Klassen, Objekte und Methoden in Java

String-Klasse

Wrapper-Klassen

UML – Eine kurze Einführung

Handyvertrag.de: LTE 3GB Tarif

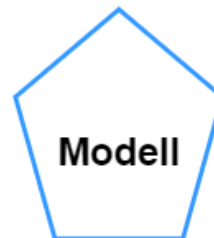
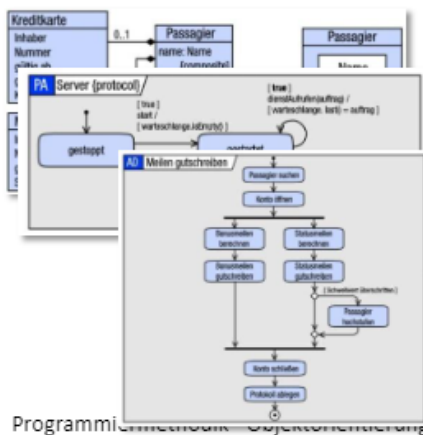
Für nur 4,99€¹ mtl. 3 GB LTE Highspeed - Wechselbonus - monatlich kündbar. J
sichern!

Handyvertrag.de: LTE 3GB Tarif

Für nur 4,99€¹ mtl. 3 GB LTE Highspeed - Wechselbonus - monatlich kündbar. J
sichern!

Modell und Diagramm

- Ein Modell stellt Abstraktion eines Realitätsausschnitts dar.
 - Um Informationen verständlicher darzustellen
 - Analog zum Erstellen von Bauplänen von Gebäuden
 - Um essentielle Systemaspekte aufzuzeigen
 - Zur Kommunikation
 - Mit Projektmitarbeitern
 - Mit Kunden
 - Um komplexe Architekturen darstellen zu können
- Ein Diagramm ist die grafische Repräsentation eines Modells.

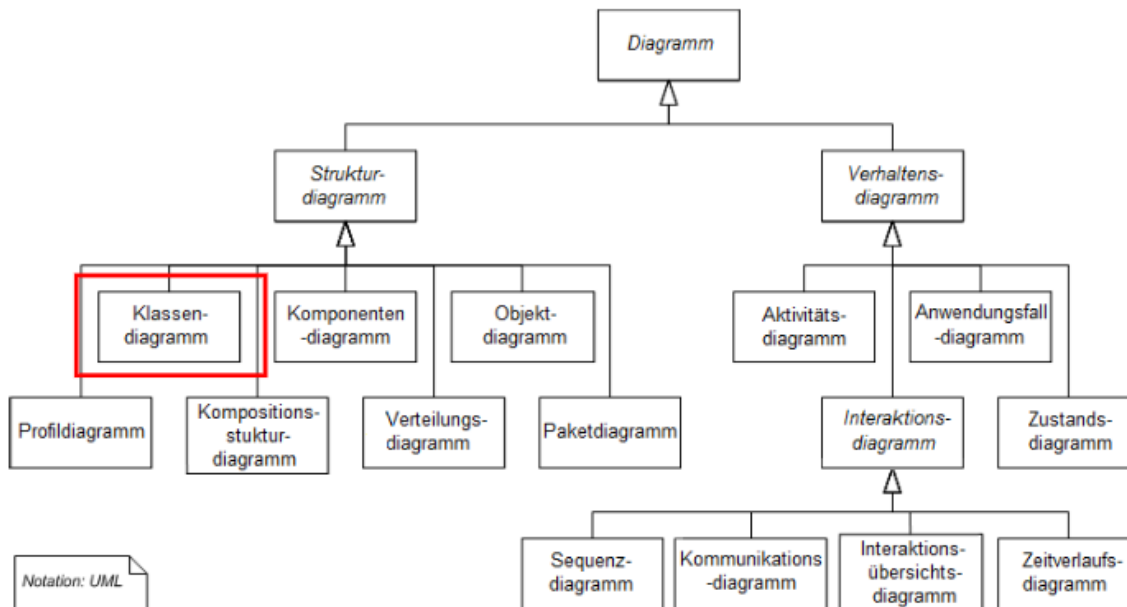


- Die Unified Modeling Language (UML) ist eine standardisierte ausdrucksstarke Modellierungssprache.
- Mit Hilfe der UML können Softwaresysteme besser **entworfen, analysiert** und **dokumentiert** werden.
- Begriff *Unified* bedeutet
 - Unterstützung des gesamten Entwicklungsprozesses.
 - Unabhängigkeit von Entwicklungswerkzeugen, sowie Programmiersprachen oder auch Anwendungsbereichen.
- Die UML ist aber nicht
 - ein Allheilmittel und vollständig,
 - ein vollständiger Ersatz für eine Textbeschreibung,
 - eine Methode oder Vorgehensmodell.

Diagrammarten (1)

- Etliche unterschiedliche Diagrammarten
- **Klassendiagramm** (in dieser Vorlesung)
 - Beschreibt den strukturellen Aspekt (Klassen, Interfaces, Beziehungen).
- **Sequenzdiagramm** (in der Vorlesung über Entwurfsmuster)
 - Beschreibt komplexe Interaktionen zwischen Objekten in bestimmten Rollen.
 - Beschreibt die zeitliche Abfolge dieser Interaktionen.
- viele mehr (z.B. Business Prozesse, Use Case-Diagramme, etc.)

Diagrammarten (2)



Grafik übernommen von http://de.wikipedia.org/wiki/Unified_Modeling_Language

Notation für Klassen (Übersicht)

Sprachkonzept	Notation
Klasse	
Klasse mit Abschnitten	

- Sichtbarkeit für Attribute und Operationen
 - + = public
 - # = protected
 - - = private
 - ~ = default
- Instanzattribute oder Klassenattribute
 - Klassenattribute werden unterstrichen
- Zusätzliche Eigenschaften angeben
 - <<readOnly>>
 - <<ordered>>
 - <<unique>>
 - <<redefines <Operationsname>>>

- Multiplizität von Attributen
 - Spezifiziert wie viele Werte ein Attribut aufnehmen kann.
 - `Attribut:Typ[a..b]`
 - `a..b`
 - mindestens a .. höchstens b
 - a und b sind natürliche Zahlen
 - * statt b bedeutet beliebig viele
 - Beispiele
 - `1..1` genau ein (entspricht 1)
 - `0..1` Null oder ein
 - `0..*` beliebig viele (entspricht *)
 - `1..*` beliebig viele aber zumindest ein