

Softwaretechnik 1 - 1. Tutorium

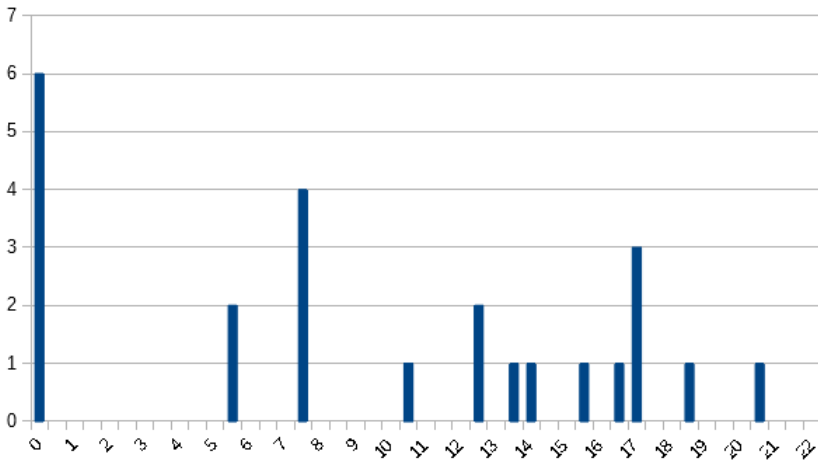
Tutorium 18

Felix Bachmann | 08.05.2018

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)



1. Übungsblatt Statistik



Allgemein

generell ohne Abzug:

- gleiche Abgabe bei allen Aufgaben

Allgemein

generell ohne Abzug:

- gleiche Abgabe bei allen Aufgaben

generell mit Abzug: (bis zu -2P)

- CheckStyle nicht beachtet
- JavaDoc !(vollständig && sinnvoll)
- Commits !(regelmäßig && aussagekräftig)

Aufgabe 1 (Altsoftware vorbereiten)

- vorgegebene .gitignore verwenden (mit IDE-Zeug), nicht nur target/

Aufgabe 1 (Altsoftware vorbereiten)

- vorgegebene .gitignore verwenden (mit IDE-Zeug), nicht nur target/
- fully.qualified.MainClass durch Paket-Struktur ersetzen (org.jis.Main)

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- `equals()` reicht nicht aus, um Gleichheit der Bilder zu prüfen
- `new File()` erstellt kein File, sondern nur einen “pointer” auf einen Pfad (siehe `File.createNewFile()` oder `File.mkdir()`)

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage
- @Test(expected=XYException.class) nutzen

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage
- @Test(expected=XYException.class) nutzen
- JUnit4 benutzen

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage
- @Test(expected=XYException.class) nutzen
- JUnit4 benutzen
- nicht throws Exception angewöhnen

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage
- @Test(expected=XYException.class) nutzen
- JUnit4 benutzen
- nicht throws Exception angewöhnen
- nicht Korrektheit der zu testenden Methoden annehmen

Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- auch bei Drehung um 0° ist Überprüfung des Bildes nötig (Dimensionen + Pixel)
- equals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- new File() erstellt kein File, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir())
- fügt Abhängigkeiten in die jmjrst.main-pom.xml ein, **nicht** in die von iMage
- @Test(expected=XYException.class) nutzen
- JUnit4 benutzen
- nicht throws Exception angewöhnen
- nicht Korrektheit der zu testenden Methoden annehmen
- Stil: Konstanten für Pfade benutzen

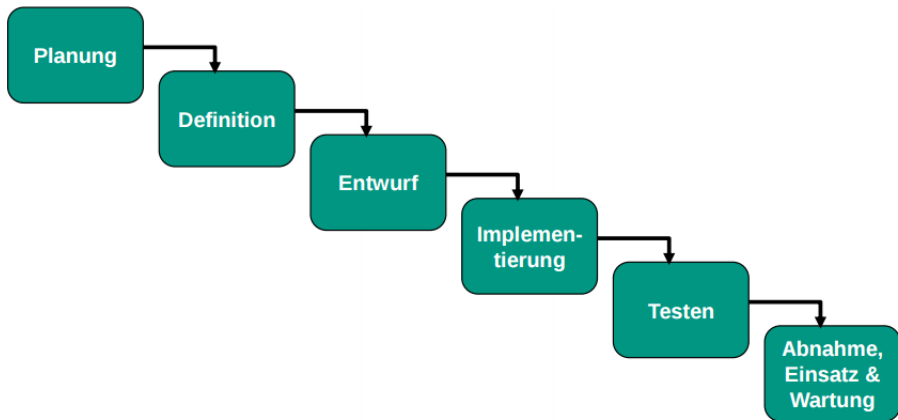
- 1 Orga
- 2 Wasserfallmodell
- 3 Durchführbarkeitsuntersuchung
- 4 Lastenheft
- 5 Pflichtenheft
- 6 UML-Klassendiagramm
- 7 \LaTeX
- 8 Tipps

■ Was ist das?

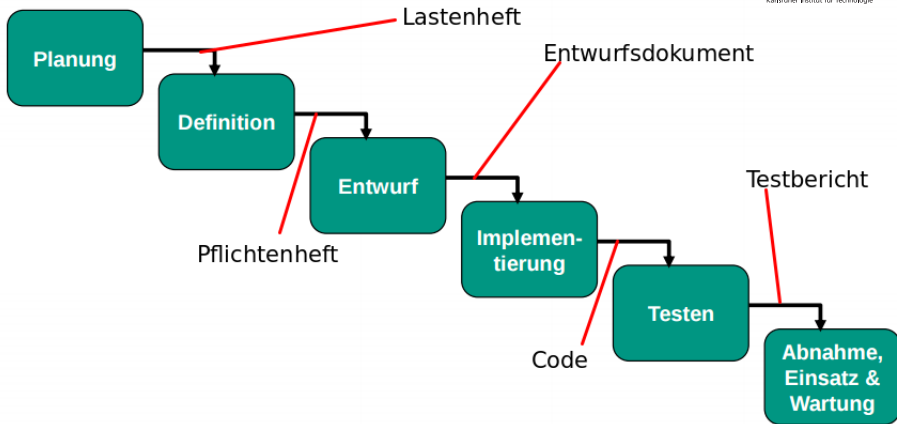
- dokumentengetriebenes Prozessmodell

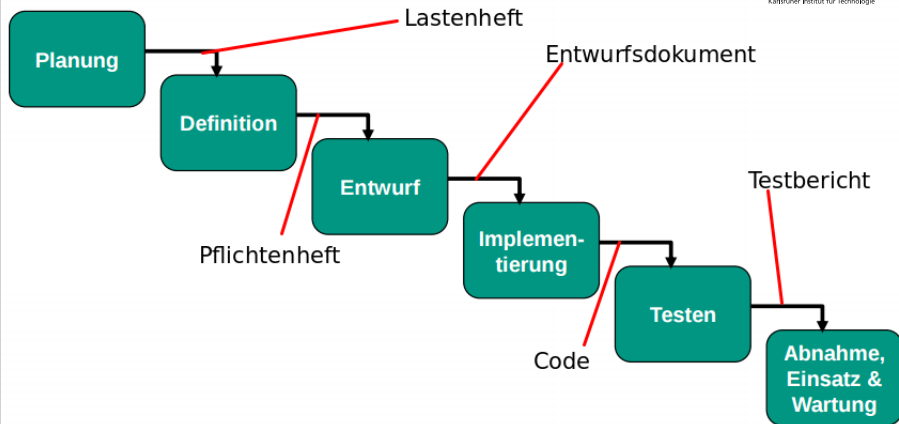
- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung

- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung



Wasserfallmodell





Dokumente für das 2. ÜB:

- Lastenheft
- Durchführbarkeitsuntersuchung (weiteres Artefakt der Planung)

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

1 Fachlich

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)
- 5 Ökonomisch

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)
- 5 Ökonomisch (wirtschaftlich? Termine?)

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)
- 5 Ökonomisch (wirtschaftlich? Termine?)
- 6 Rechtlich

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)
- 5 Ökonomisch (wirtschaftlich? Termine?)
- 6 Rechtlich (Datenschutz, Standards)

Durchführbarkeitsuntersuchung - Gliederung

Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- 1 Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- 3 Personell (genug qualifiziertes Personal?)
- 4 Risiken (Gibt es Risiken? :D)
- 5 Ökonomisch (wirtschaftlich? Termine?)
- 6 Rechtlich (Datenschutz, Standards)

Fürs Übungsblatt

Denkt euch was (plausibles) aus!

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- 1 Zielbestimmung (grobe Beschreibung)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- 1 Zielbestimmung (grobe Beschreibung)
- 2 Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- 1 Zielbestimmung (grobe Beschreibung)
- 2 Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)
- 3 Funktionale Anforderungen (feingranular: Funktionen des Produkts)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- 1 Zielbestimmung (grobe Beschreibung)
- 2 Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)
- 3 Funktionale Anforderungen (feingranular: Funktionen des Produkts)
- 4 Produktdaten (Welche Daten speichern?)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- 1 Zielbestimmung (grobe Beschreibung)
- 2 Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)
- 3 Funktionale Anforderungen (feingranular: Funktionen des Produkts)
- 4 Produktdaten (Welche Daten speichern?)
- 5 Nichtfunktionale Anforderungen (Meta-Anforderungen: Zeit, Zuverlässigkeit)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- ① Zielbestimmung (grobe Beschreibung)
- ② Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)
- ③ Funktionale Anforderungen (feingranular: Funktionen des Produkts)
- ④ Produktdaten (Welche Daten speichern?)
- ⑤ Nichtfunktionale Anforderungen (Meta-Anforderungen: Zeit, Zuverlässigkeit)
- ⑥ Systemmodelle
 - Szenarien (spezielles Beispiel)
 - Anwendungsfälle (allgemeiner Verwendungszweck)

Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer.

- ① Zielbestimmung (grobe Beschreibung)
- ② Produkteinsatz (Für wen? Zielgruppe, Anwendungsbereich)
- ③ Funktionale Anforderungen (feingranular: Funktionen des Produkts)
- ④ Produktdaten (Welche Daten speichern?)
- ⑤ Nichtfunktionale Anforderungen (Meta-Anforderungen: Zeit, Zuverlässigkeit)
- ⑥ Systemmodelle
 - Szenarien (spezielles Beispiel)
 - Anwendungsfälle (allgemeiner Verwendungszweck)
- ⑦ Glossar (technische Begriffe erklären)

Zielbestimmung vs. Funktionale Anforderungen

Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Funktionale Anforderungen: konkrete Auflistung von Funktionen

Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Funktionale Anforderungen: konkrete Auflistung von Funktionen

Funktionale Anforderungen vs. Nichtfunktionale Anforderungen

Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Funktionale Anforderungen: konkrete Auflistung von Funktionen

Funktionale Anforderungen vs. Nichtfunktionale Anforderungen

- Funktionale Anforderungen: Funktionen des Produkts
- Nichtfunktionale Anforderungen: "Meta"-Eigenschaften des Produkts

Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Funktionale Anforderungen: konkrete Auflistung von Funktionen

Funktionale Anforderungen vs. Nichtfunktionale Anforderungen

- Funktionale Anforderungen: Funktionen des Produkts
- Nichtfunktionale Anforderungen: "Meta"-Eigenschaften des Produkts

Zielbestimmung vs. Produkteinsatz

Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Funktionale Anforderungen: konkrete Auflistung von Funktionen

Funktionale Anforderungen vs. Nichtfunktionale Anforderungen

- Funktionale Anforderungen: Funktionen des Produkts
- Nichtfunktionale Anforderungen: "Meta"-Eigenschaften des Produkts

Zielbestimmung vs. Produkteinsatz

- Zielbestimmung: allgemeine Beschreibung, was das Produkt können soll
- Produkteinsatz: Rahmenbedingungen (Zielgruppe, Anwendungsbereiche)

Grundlegende Aufgabe

Erweiterung des Lastenheftes, sodass exakt abgebildet ist **was** (noch nicht **wie**) zu implementieren ist.

- ① Zielbestimmung
- ② Produkteinsatz
- ③ **Produktumgebung** (Hard-/Software in Einsatzumgebung)
- ④ Funktionale Anforderungen
- ⑤ Produktdaten
- ⑥ Nichtfunktionale Anforderungen
- ⑦ **Globale Testfälle** (“zu testende Abläufe”)
- ⑧ Systemmodelle
 - Szenarien
 - Anwendungsfälle
 - **Objektmodelle** \implies UML-Klassendiagramme (heute)
 - **Dynamische Modelle** \implies nächstes Mal
 - **Benutzerschnittstelle** \implies Zeichnungen/Screenshots
- ⑨ Glossar

Produkteinsatz vs. Produktumgebung

Produkteinsatz vs. Produktumgebung

- Produkteinsatz: Rahmenbedingungen (Zielgruppe, Anwendungsbereiche)
- Produktumgebung: Rahmenbedingungen bzgl. Software/Hardware

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes.

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase.

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte.

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte. falsch

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte. falsch
- Das Pflichtenheft beschreibt nur, was zu implementieren ist und nicht wie.

Wahr oder falsch?

- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte. falsch
- Das Pflichtenheft beschreibt nur, was zu implementieren ist und nicht wie. wahr

Wahr oder falsch?

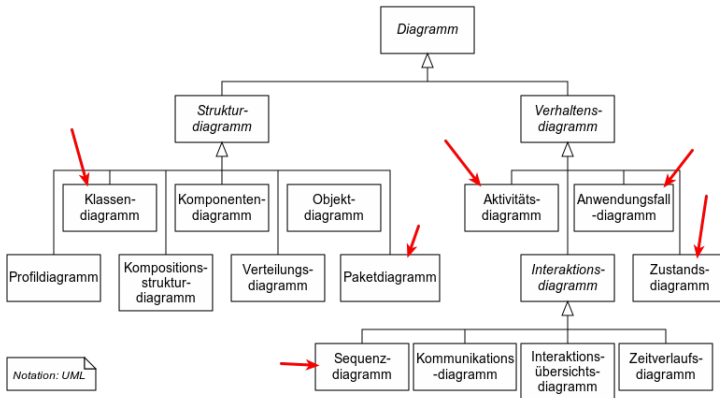
- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte. falsch
- Das Pflichtenheft beschreibt nur, was zu implementieren ist und nicht wie. wahr
- Nicht-funktionale Anforderungen sind sowohl Teil des Pflichtenhefts als auch des Lastenhefts.

Wahr oder falsch?

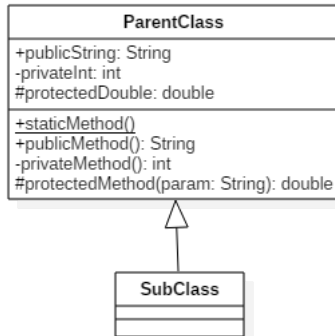
- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes. falsch
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte. falsch
- Das Pflichtenheft beschreibt nur, was zu implementieren ist und nicht wie. wahr
- Nicht-funktionale Anforderungen sind sowohl Teil des Pflichtenhefts als auch des Lastenhefts. wahr

UML? Kann man das essen?

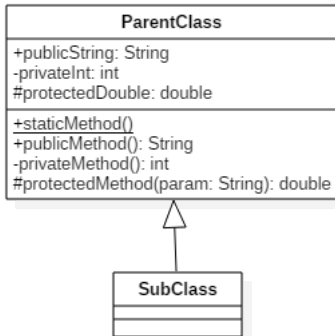
- UML = **U**nified **M**odeling **L**anguage
- grafische Modellierungssprache, strenge Syntax



UML-Klassendiagramm: Klassen + Vererbung

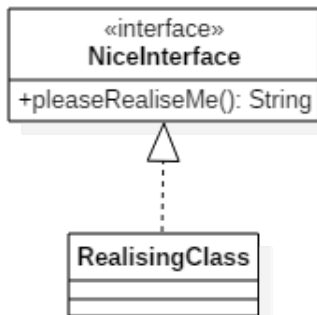


UML-Klassendiagramm: Klassen + Vererbung

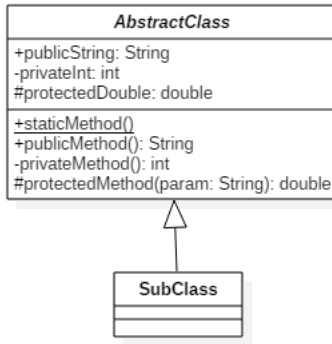


- - ist private: von Instanzen derselben Klasse sichtbar (**aber von allen!**)

UML-Klassendiagramm: Interface

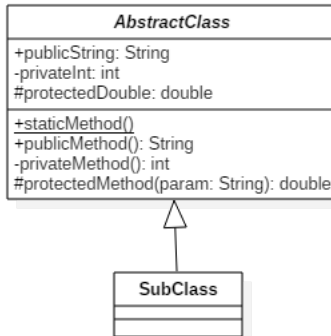


UML-Klassendiagramm: Abstrakte Klassen

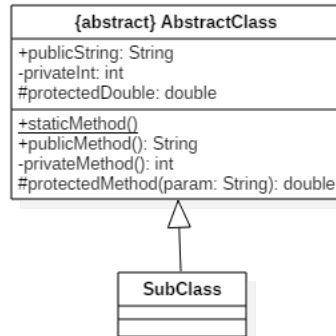


UML-Notation

UML-Klassendiagramm: Abstrakte Klassen



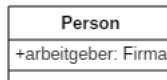
UML-Notation



Notation für Abgaben

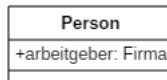
Kursiv schriftlich nicht erkennbar!

UML-Klassendiagramm: Assoziationen



so nicht,...

UML-Klassendiagramm: Assoziationen

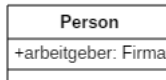


so nicht,...

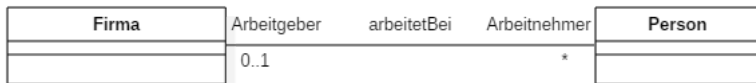


..., sondern so

UML-Klassendiagramm: Assoziationen



so nicht,...



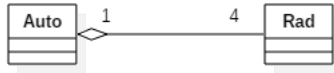
..., sondern so

UML

Beziehungen sollen direkt ersichtlich werden

⇒ nur primitive Typen als Felder

UML-Klassendiagramm: Aggregation + Komposition

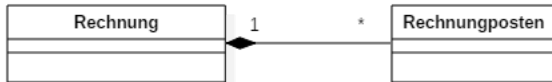


Aggregation

UML-Klassendiagramm: Aggregation + Komposition



Aggregation



Komposition

Unterschied

- Aggregation: Teil-Ganzes-Beziehung
- Komposition: Aggregation, Teil kann ohne Ganzes nicht existieren

Text \Rightarrow UML-Diagramm

Jeder Student hat eine Matrikelnummer und einen Namen. Ein fauler Student ist ein Student, der schlafen kann. Er hat dazu ein Bett. Ein fleißiger Student hingegen, kann lernen und hat dazu einen Computer, der aus Bauteilen besteht.

Text \Rightarrow UML-Diagramm

Jeder Student hat eine Matrikelnummer und einen Namen. Ein fauler Student ist ein Student, der schlafen kann. Er hat dazu ein Bett. Ein fleißiger Student hingegen, kann lernen und hat dazu einen Computer, der aus Bauteilen besteht.

UML-Diagramm?

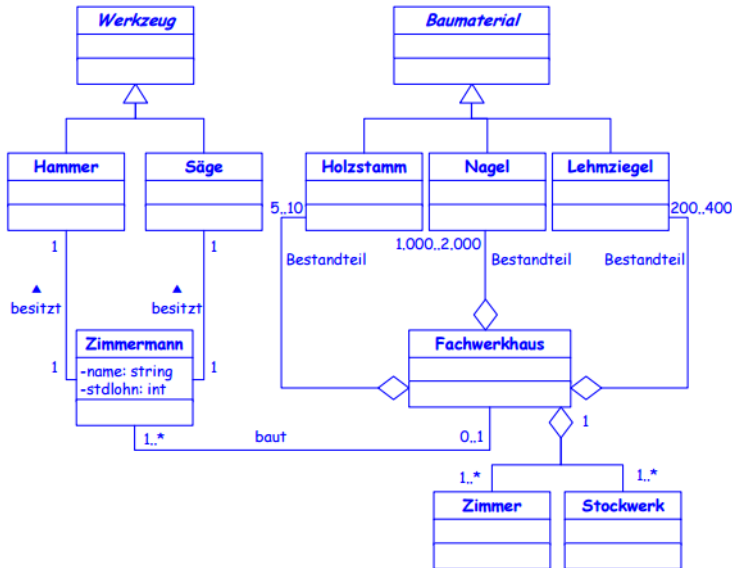
Text \Rightarrow UML-Diagramm

Jeder **Student** hat eine Matrikelnummer und einen Namen. Ein fauler Student **ist ein** Student, der schlafen kann. Er **hat** dazu ein Bett. Ein fleißiger Student hingegen, kann lernen und **hat** dazu einen **Computer**, der aus **Bauteilen besteht**.

Schlüsselwörter!

Modellieren Sie das Szenario möglichst vollständig als UML-Klassendiagramm. Modellieren Sie keine Methoden. Geben Sie Attribute, Multiplizitäten, Restriktionen, Assoziationsnamen sowie Rollen an.

Ein Fachwerkhaus besteht aus 5 bis 10 Holzstämmen, 200 bis 400 Lehmziegeln sowie 1.000 bis 2.000 Nägeln. Jedes Baumaterial, egal ob Holzstamm, Lehmziegel oder Nagel, ist Bestandteil in genau einem Fachwerkhaus. Jedes Fachwerkhaus hat eine bestimmte Anzahl an Zimmern und Stockwerken. Für den Bau eines Fachwerkhauses ist mindestens ein Zimmermann zuständig, welcher einen Namen sowie einen individuellen Stundenlohn besitzt. Zum Bau des Fachwerkhauses verwendet jeder Zimmermann sein eigenes Werkzeug, bestehend aus genau einem Hammer sowie genau einer Säge. Jeder Zimmermann kann an maximal einem Fachwerkhaus gleichzeitig bauen.



- auf dem Blatt müsst ihr L^AT_EX für die Dokumente benutzen
- nicht wie z.B. Word WYSIWYG, sondern WYSIWYAF / WYSIWYM
- wird euch an der Uni immer wieder begegnen, oft Pflicht

- auf dem Blatt müsst ihr L^AT_EX für die Dokumente benutzen
- nicht wie z.B. Word WYSIWYG, sondern WYSIWYAF / WYSIWYM
- wird euch an der Uni immer wieder begegnen, oft Pflicht
- Vorteile:
 - gut versionierbar
 - leicht Formeln erstellbar
 - nach Eingewöhnung recht intuitiv (vergleichbar mit HTML)
 - multifunktional (Dokumente, Präsentationen, ...)

- auf dem Blatt müsst ihr L^AT_EX für die Dokumente benutzen
- nicht wie z.B. Word WYSIWYG, sondern WYSIWYAF / WYSIWYM
- wird euch an der Uni immer wieder begegnen, oft Pflicht
- Vorteile:
 - gut versionierbar
 - leicht Formeln erstellbar
 - nach Eingewöhnung recht intuitiv (vergleichbar mit HTML)
 - multifunktional (Dokumente, Präsentationen, ...)
- Nachteile:
 - Einarbeitung notwendig :(

Installation einer Distribution notwendig, z.B.:

- MiKTeX für Windows
- TeX Live für Linux, Mac, Windows

Installation einer Distribution notwendig, z.B.:

- MikTeX für Windows
- TeX Live für Linux, Mac, Windows

Editoren machen das Schreiben von L^AT_EX-Dokumenten angenehmer

- Texmaker
- TeXstudio (erweiterter Texmaker, mein Favorit)
- TeXclipse (Plugin für Eclipse)
- ...

- Präambel: Includes von Paketen
 - `\documentclass{Klasse}` (z.B. book, letter)
 - `\usepackage[option1, option2,...]{Paket}`

- Präambel: Includes von Paketen
 - `\documentclass{Klasse}` (z.B. book, letter)
 - `\usepackage[option1, option2,...]{Paket}`
- Inhalt: Text setzen
 - Struktur: part, (chapter), section, subsection, subsubsection
 - Auflistungen: `\begin{itemize}` `\item Hello World!` `\end{itemize}`
 - Bilder: `\includegraphics[scale = 0.8]{PfadZumBild}`

- Präambel: Includes von Paketen
 - `\documentclass{Klasse}` (z.B. book, letter)
 - `\usepackage[option1, option2,...]{Paket}`
- Inhalt: Text setzen
 - Struktur: part, (chapter), section, subsection, subsubsection
 - Auflistungen: `\begin{itemize}` `\item Hello World!` `\end{itemize}`
 - Bilder: `\includegraphics[scale = 0.8]{PfadZumBild}`

Beispiel!

Aufgabe 1 + 3: Lastenheft + Durchführbarkeitsuntersuchung

- lasst euch was (sinnvolles) einfallen
- benutzt \LaTeX

Aufgabe 1 + 3: Lastenheft + Durchführbarkeitsuntersuchung

- lasst euch was (sinnvolles) einfallen
- benutzt \LaTeX

Aufgabe 2: Klassendiagramme

- achtet auf Schlüsselwörter (“ist ein“, “enthält ein“, “besteht aus“, ...)

Aufgabe 1 + 3: Lastenheft + Durchführbarkeitsuntersuchung

- lasst euch was (sinnvolles) einfallen
- benutzt \LaTeX

Aufgabe 2: Klassendiagramme

- achtet auf Schlüsselwörter (“ist ein“, “enthält ein“, “besteht aus“, ...)

Aufgabe 4 + 5: Shutterpile

- an einigen Stellen sind Aufgaben etwas vage
 \implies überlegt euch, was Sinn macht
- Zusammenhang der Klassen unklar? Vielleicht hilft Diagramm

Abgabe

- Deadline am 16.5 um 12:00
- Dokumente ausdrucken
- Klassendiagramme handschriftlich

