

### Softwaretechnik 1 - 1. Tutorium

Tutorium 17 Felix Bachmann | 14.05.2019

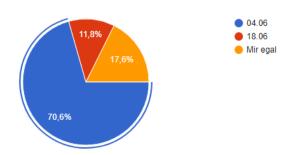
KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)

#### **Ersatztermin**



#### Wann soll das Ersatztut für den 11.06. stattfinden?

17 Antworten

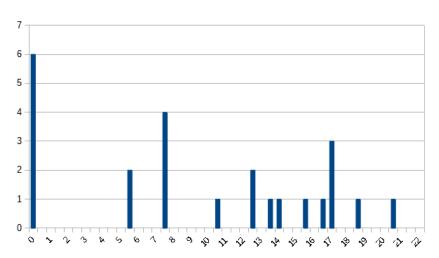


- Tut von 11.06 wird auf 04.06, verschoben
- ansonsten alles gleich (11:30-13:00, -119)

Orga	
•00000	

# 1. Übungsblatt Statistik





Lastenheft



#### 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/



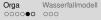
#### Aufgabe 1 (Altsoftware vorbereiten)

- 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)



6/41



### 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

6/41



- 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())



- 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



- 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





- 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



- 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



- 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



- 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



#### **Themenübersicht**



- Orga
- Wasserfallmodell
- Durchführbarkeitsuntersuchung
- Lastenheft
- Pflichtenheft
- UML-Klassendiagramm
- LAT⊨X
- Tipps

14.05.2019



Was ist das?



dokumentengetriebenes Prozessmodell



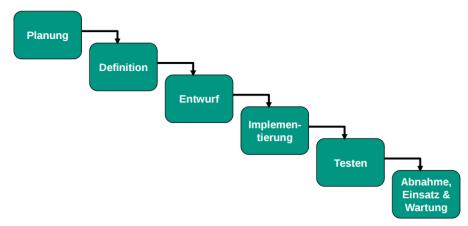


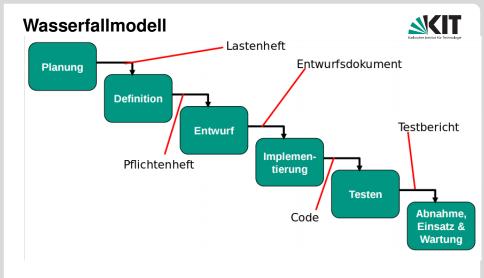
- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung

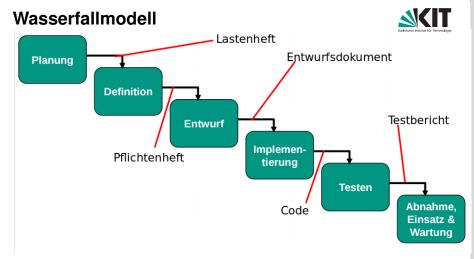
9/41



- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung







#### Dokumente für das 2. ÜB:

- Lastenheft
- Durchführbarkeitsuntersuchung (weiteres Artefakt der Planung)

ooooo oo● Felix Bachmann – SWT1

14.05.2019



#### Grundlegende Frage



### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

Fachlich



#### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

Fachlich (softwaretechnisch leicht realisierbar?)

Lastenheft

14.05.2019



#### Grundlegende Frage

- Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen



#### Grundlegende Frage

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



#### Grundlegende Frage

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



### Grundlegende Frage

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



### Grundlegende Frage

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



#### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- Fachlich (softwaretechnisch leicht realisierbar?)
- Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- Personell (genug qualifizertes Personal?)
- Alsiken (Gibt es Risiken? :D)

11/41



#### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- Fachlich (softwaretechnisch leicht realisierbar?)
- Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- Personell (genug qualifizertes Personal?)
- Alsiken (Gibt es Risiken? :D)
- Ökonomisch

Felix Bachmann - SWT1



#### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

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

Felix Bachmann - SWT1

14.05.2019



#### Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- Fachlich (softwaretechnisch leicht realisierbar?)
- Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- Personell (genug qualifizertes Personal?)
- Alsiken (Gibt es Risiken? :D)
- Ökonomisch (wirtschaftlich? Termine?)
- Rechtlich

Felix Bachmann - SWT1



#### Grundlegende Frage

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

# Durchführbarkeitsuntersuchung



## Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- Fachlich (softwaretechnisch leicht realisierbar?)
- Alternativen (lieber altes Projekt anpassen oder komplett neu entwickeln?)
- Personell (genug qualifizertes Personal?)
- Alsiken (Gibt es Risiken? :D)
- Ökonomisch (wirtschaftlich? Termine?)
- Rechtlich (Datenschutz, Standards)

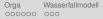
## Fürs Übungsblatt

Denkt euch was (plausibles) aus!



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.



12/41



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

Zielbestimmung (grobe Beschreibung)



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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

Felix Bachmann - SWT1



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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



### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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

000000



12/41

### Grundlegende Aufgabe

Das Lastenheft sammelt die Anforderungen des Auftraggebers an den Auftragnehmer. Theoretisch vom Kunden geschrieben.

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



## Zielbestimmung vs. Funktionale Anforderungen

14.05.2019



### Zielbestimmung vs. Funktionale Anforderungen

- Zielbestimmung: allgemeine Beschreibung, was das Produkt k\u00f6nnen 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

13/41



### 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 k\u00f6nnen 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)

000000

13/41

### Wozu ein Pflichtenheft?



### Grundlegende Aufgabe

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

### Wozu ein Pflichtenheft?



## Grundlegende Aufgabe

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

- man kann es sich so merken
  - erst werden uns "Lasten" vom Kunden auferlegt
  - daraus generieren wir dann "Pflichten"
  - "Pflichten" Grundlage für Entwurf

# Pflichtenheft - Gliederung



- Zielbestimmung
- Produkteinsatz
- **Produktumgebung** (Hard-/Software in Einsatzumgebung)
- Funktionale Anforderungen
- Produktdaten
- Nichtfunktionale Anforderungen
- Globale Testfälle ("zu testende Abläufe")
- Systemmodelle
  - Szenarien
  - Anwendungsfälle
  - **Objektmodelle**  $\Longrightarrow$  UML-Klassendiagramme (heute)
  - **Dynamische Modelle** ⇒ nächstes Mal
  - **Benutzerschnittstelle** ⇒ Zeichnungen/Screenshots
- Glossar

Wasserfallmodell



### 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.





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



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



#### Wahr oder falsch?

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

17/41



- 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 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.



- 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



- 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?

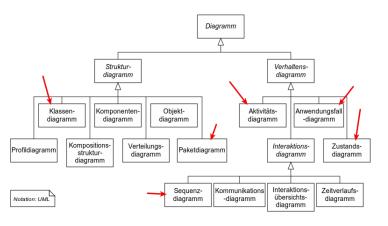
- Das Lastenheft ist eine Verfeinerung des Pflichtenheftes.
- Das Lastenheft ist das Ergebnis der Planungsphase. wahr
- Nicht-funktionale Eigenschaften beschreiben, was das Produkt nicht tun sollte.
- Das Pflichtenheft beschreibt nur, was zu implementieren ist und nicht wie. wahr
- Nicht-funktionale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

17/41

#### UML? Kann man das essen?



- UML = Unified Modeling Language
- grafische Modellierungssprache, strenge Syntax



14.05.2019

## **UML-Klassendiagramm: Syntax**



- Name der Klasse
- Attribute
- Methoden

- Klassenname
  - keine spezielle Syntax
  - einfach Namen hinschreiben
- Attribute
  - <modifier><name>:<type>
- Methoden
  - <modifier><name>(<parameters>):<type>
  - <parameters>
    - kann leer sein
    - oder komma-getrennte Liste von <name>:<type>
  - falls Rückgabe void, :<type> weglassen
- statische Methoden und Attribute unterstreichen

### **Modifier**



- UML
  - - private
    - von Instanzen derselben Klasse sichtbar (aber von allen!)
  - **#**
- protected (wie in Java)
- von Instanzen derselben Klasse, aller Unterklassen und Instanzen aus dem gleichen Paket sichtbar
- -+
- public (wie in Java)
- von Instanzen jeder Klasse sichtbar
- falls nichts angegeben implizit public

# **Beispiel**



#### Hund

- name: String
- rasse: Rasse
- gewicht: int
- + wiegen(): int
- + streicheln()
- + streicheln(intensität: int, ausruf: String)
- + füttern(ration: Nahrung)

# Vererbung



#### **ParentClass**

+publicString: String

-privateInt: int

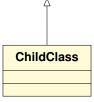
#protectedDouble: double

+staticMethod()

+publicMethod(): String

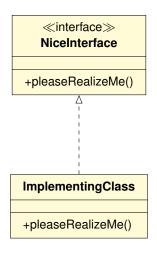
-privateMethod(): int

#protectedMethod(param: String): double



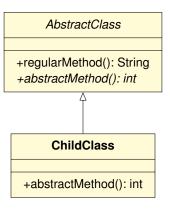
#### Interface





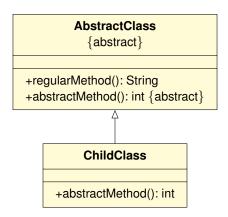
### Abstrakte Klassen





# Abstrakte Klassen: Abgaben





- für Übungsblätter und Klausur
  - kursiv nicht erkennbar, stattdessen {abstract} verwenden
  - laut VL unter Klassenname, hinter Methode

### **Assoziationen**



#### Firma

angestellte: List<Person>

#### **Person**

arbeitgeber: Firma

### Assoziationen



# Firma angestellte: List<Person>

# Person arbeitgeber: Firma

#### Probleme

- List<X> ist Java-Syntax und schreibt Datenstruktur vor
- Beziehungen sollen direkt ersichtlich werden
- Faustregel: nur primitive Typen als Attribute hinschreiben

Firma	Arbeitgeber	Arbeitnehmer	Person
	01	*	

26/41

# **Aggregation und Komposition**



Aggregation = Teil-Ganzes-Beziehung



# **Aggregation und Komposition**



Aggregation = Teil-Ganzes-Beziehung



- Komposition: Aggregation, aber Teil kann ohne Ganzes nicht existieren
  - wenn ganzes gelöscht wird, dann auch Teile!



# Klassischer Aufgabentyp



### Text ⇒ UML-Klassendiagramm

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.

# Klassischer Aufgabentyp



### Text ⇒ UML-Klassendiagramm

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?** 

# Klassischer Aufgabentyp



### Text ⇒ UML-Klassendiagramm

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!

# Klausuraufgabe SS11



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

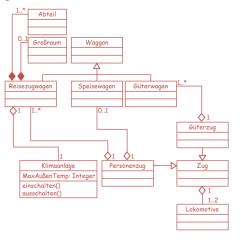
#### Szenario

Ein Güterzug ist ein Zug, dessen Waggons ausschließlich Güterwagen sind. Die Waggons eines Personenzugs sind mindestens ein Reisezugwagen und höchstens ein Speisewagen. Jeder Zug hat eine oder zwei Lokomotiven. Reisezugwagen setzen sich aus bis zu einem Großraum sowie einem oder mehreren Abteilen zusammen. Jeder Reisezugwagen hat eine Klimaanlage, die ein- und ausgeschaltet werden kann. Jede Klimaanlage darf nur bis zu einer bestimmten maximalen Außentemperatur betrieben werden.

ooooo ooo Felix Bachmann – SWT1

# Musterlösung





Klassen	11×	0,5 P
Methoden/Attribute "Klimaanlage"	1×	0,5 P
Aggregationen/Kompositionen	7x	0,5 P
Vererbung (alle 3 zusammen)	1×	0,5 P

Orga Wasserfallmodell

# LATEX- Basics



- auf dem Blatt müsst ihr LATEX für die Dokumente benutzen
- wird euch an der Uni immer wieder begegnen, manchmal Pflicht

Wasserfallmodell

# LATEX- Basics

Felix Bachmann - SWT1



LAT<sub>E</sub>X

32/41

14.05.2019

- auf dem Blatt müsst ihr LATEX für die Dokumente benutzen
- wird euch an der Uni immer wieder begegnen, manchmal Pflicht
- nicht WYSIWYG
  - What You See Is What You Get
  - Paradigma von Word etc.

#### (a) What You See

The second secon

#### (b) What You Get

Lover legan de deux de seus conseines calaptirique (els, vod dans messes e introd impose includes de debiter e delle mes que de seque mes con delle messes, de Avere me est enterme plane des delicions e se rebeta. Not chiu desa phagogene, con seus columnius externa en Laverin, possibilità del delicion e se rebeta. Del chiu desa phagogene, con seu columnius columnius contra con Laverin, possibilità del contra delicioni della messes delicioni della messigna. Al verse me del averina columnius co

Input mon cert debut angue des deliver de lenguir und Lacifas. Lerem iguem debut si men, conservatur adiptici, gel cui del dam menumey nibb existend discident si lovere cident magna aliquam este valujus.

Li mais retri ad minima conquira constand associ tation sillamosper uncipi loborità sisi at adiquip ce se controdole conseque. Deli associ vel com il lamo delor ai bestiendi in valquate vella adiquip ce se controdole conseque. Deli associ vel com il lamo delor ai bestiendi in valquate vella

Namilber temper com solata noble elekted option compare mbil imperdier doming al-quod maxin placeral faire position same. Levers i peam delare el amer, consenten ellipsicaleg elle, sel dissen nonumme mbil ecishmol decichtor al moren delare magna aliquam ente volugar. Un visi cent an aminin venium, quis notical eversi inton allamonaper savaight laborité nels si aliquip es en commodo commegnat.

Data autom vol eum inture dolor in bendrerk in vulpatate velit euer emlende comoquat, vel filom dolore en fragiat nalla facilities.

At some one or assume the passe described to a relative for the last of passenges, we see admission of the contract of the last of passenges, we see the contract of the last owners greatly of the last of the last owners are last of the l

### LATEX- Basics



- stattdessen WYSIWYAF bzw. WYSIWYM
  - What You See Is What You Ask For / Mean
  - Paradigma von LaTEX HTML und CSS

#### (a) What You See

\subsection{Pflichtenheft - Gliederung} \begin{frame}{Pflichtenheft - Gliederung} \begin{enumerate} \item Zielbestimmung \item Produkteinsatz \item \underline{\textbf{Produktumgebung}} (Hard-/Software in Einsatzumgebung) \item Funktionale Anforderungen \item Produktdaten \item Nichtfunktionale Anforderungen \item \underline{\textbf{Globale Testfälle}} (\enquote{zu testende Abläufe}) \item Systemmodelle \begin{itemize} \item Szenacien \item Anwendungsfälle \item \underline{\textbf{Qbjektmodelle}} \$\implies\$ UML-Klassendiagramme (heute) \item \underline{\textbf{Dynamische Modelle}} \$\implies\$ n\u00e4chstes Mal \item \underline{\textbf{Benutzerschnittstelle}} \$\implies\$ Zeichnungen/Screenshots \end{itemize} \item Glossar \end{enumerate} \end{frame}

#### (b) What You Mean



Wasserfallmodell





- Vorteile
  - gut versionierbar
    - "Quellcode" ist normales Textdokument
    - Word etc. verwenden oft XML-Formate mit Metadaten
  - leicht Formeln erstellbar
  - nach Eingewöhnung recht intuitiv
  - multifunktional (Bücher, Dokumente, Präsentationen, ...)
  - open source, kostet nix :)
  - viele Erweiterungen, Pakete, . . .
- Nachteile
  - Einarbeitung notwendig :(

# LATEX- Installation



35/41

Installation einer Distribution notwendig, z.B.:

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

# LATEX- Installation



Installation einer Distribution notwendig, z.B.:

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

Editoren machen das Schreiben von LATEX-Dokumenten angenehmer

- Texmaker
- TeXstudio (erweiterter Texmaker, mein Favorit)
- TeXclipse (Plugin für Eclipse)
- Plugins für Visual Studio Code
- . . . .



Felix Bachmann - SWT1

# LATEX- Dokumentaufbau



### Präambel: Pakete laden, Dokumenttyp festlegen

```
\documentclass{Klasse} \usepackage[option1,option2,...]{Paket}
```

- nützliche Klassen: book, beamer, scrartcl
- nützliches Paket z.B. csquotes (ermöglicht \enquote{...})

# LATEX- Dokumentaufbau



LATEX

36/41

### Präambel: Pakete laden, Dokumenttyp festlegen

```
\documentclass{Klasse}
\usepackage[option1,option2,...]{Paket}
```

- nützliche Klassen: book, beamer, scrartcl
- nützliches Paket z.B. csquotes (ermöglicht \enquote{...})

### Inhalt: Text setzen, Bilder, Graphiken, Formeln,...

- Struktur: part, (chapter), section, subsection, subsubsection
- Auflistungen: \begin{itemize} \item Hello World! \end{itemize}
- Bilder: \includegraphics[scale = 0.8]{PfadZumBild}

# LATEX-Beispiel!



### Aufgabe 1 + 3: Lastenheft + Durchführbarkeitsuntersuchung

- lasst euch was (sinnvolles) einfallen
  - 6+3 Punkte für < 5 Seiten sinnvollen Text!</p>
    - besser wirds nicht
    - aber: nicht zu allgemein werden, Bezug zu Szenario, Form
- Anwendungsfalldiagramm: Syntax beachten
- Durchführbarkeitsuntersuchung: Fragen beantworten, nicht stellen!



Felix Bachmann - SWT1



### Aufgabe 1 + 3: Lastenheft + Durchführbarkeitsuntersuchung

- lasst euch was (sinnvolles) einfallen
  - 6+3 Punkte für < 5 Seiten sinnvollen Text!</p>
    - besser wirds nicht
    - aber: nicht zu allgemein werden, Bezug zu Szenario, Form
- Anwendungsfalldiagramm: Syntax beachten
- Durchführbarkeitsuntersuchung: Fragen beantworten, nicht stellen!

### Aufgabe 2: Klassendiagramme

- Form, Syntax
- achtet auf Schlüsselwörter ("ist ein", "enthält ein", "besteht aus",...)



### Aufgabe 4 + 5: HDrize

- HDR implementieren
  - aus Belichtungsreihe
- Zusammenhang der Klassen unklar? Vielleicht hilft Diagramm



### Aufgabe 4 + 5: HDrize

- HDR implementieren
  - aus Belichtungsreihe
- Zusammenhang der Klassen unklar? Vielleicht hilft Diagramm

### Aufgabe 6: Schnittstellen für Filter

- nur Schnittstellen definieren
- JavaDoc!

### Denkt dran!



### Abgabe

- Deadline am 22.5 um 12:00
- Dokumente ausdrucken
- Klassendiagramme handschriftlich
- Deckblatt!



Felix Bachmann - SWT1

40/41

# Bis dann! (dann := 28.05.18)



