

Softwaretechnik 1 - 1. Tutorium

Tutorium 17 Felix Bachmann | 14.05.2019

KIT - INSTITUT FÜR PROGRAMMSTRUKTUREN UND DATENORGANISATION (IPD)

Themenübersicht



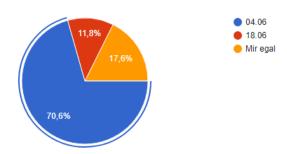
- 1 Orga
- Wasserfallmodell
- 3 Durchführbarkeitsuntersuchung
- 4 Lastenheft
- Opening the second of the s
- UML-Klassendiagramm
- 7 LATEX
- 8 Tipps

Ersatztermin



Wann soll das Ersatztut für den 11.06, stattfinden?

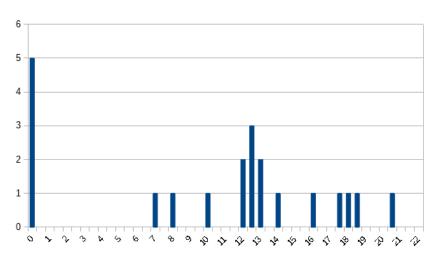
17 Antworten



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

1. Übungsblatt Statistik







Allgemeine Anmerkungen



für Feedback zu eurem Code Deckblatt einwerfen!

Punktevergabe

generell ohne Abzug

gleiche Abgabe bei allen Aufgaben



Allgemeine Anmerkungen



für Feedback zu eurem Code Deckblatt einwerfen!

Punktevergabe

generell ohne Abzug

gleiche Abgabe bei allen Aufgaben

generell mit Abzug

- Code nicht CheckStyle-konform (auch Tests!)
 - Save Actions bei Eclipse! (IntelliJ ähnlich)
- JavaDoc nicht (vollständig && sinnvoll)
- Git-Commits nicht (regelmäßig && aussagekräftig)

Allgemeine Anmerkungen - pom.xml



Abhängigkeiten deklarieren

- iMage dient dazu jmjrst.main zu verpacken
- wenn ihr Abhängigkeiten in iMage-pom hinzufügt funktioniert es bei euch
 - aber es wird jmjrst.main-pom in zip gepackt
 - bei mir explodiert es, da mein iMage die Abhängigkeit nicht hat

Lösung: Abhängigkeiten in dem Untermodul, das ihr gerade abgebt, deklarieren

Felix Bachmann - SWT1

Allgemeine Anmerkungen - pom.xml



Abhängigkeiten deklarieren

- iMage dient dazu jmjrst.main zu verpacken
- wenn ihr Abhängigkeiten in iMage-pom hinzufügt funktioniert es bei euch
 - aber es wird jmjrst.main-pom in zip gepackt
 - bei mir explodiert es, da mein iMage die Abhängigkeit nicht hat

Lösung: Abhängigkeiten in dem Untermodul, das ihr gerade abgebt, deklarieren

Parent

- als parent von Untermodulen iMage, nicht das remote uebungsparent benutzen!
- als Version des Parents 0.0.1-SNAPSHOT

Allgemeine Anmerkungen - pom.xml



Abhängigkeiten deklarieren

- iMage dient dazu imirst.main zu verpacken
- wenn ihr Abhängigkeiten in iMage-pom hinzufügt funktioniert es bei euch
 - aber es wird jmjrst.main-pom in zip gepackt
 - bei mir explodiert es, da mein iMage die Abhängigkeit nicht hat

Lösung: Abhängigkeiten in dem Untermodul, das ihr gerade abgebt, deklarieren

Parent

- als parent von Untermodulen iMage, nicht das remote uebungsparent benutzen!
- als Version des Parents 0.0.1-SNAPSHOT

ab nächstem ÜB Punktabzug, wenn ich noch irgendwas an der pom ändern muss, damit es bei mir läuft

14.05.2019



Aufgabe 1 (Altsoftware vorbereiten)

- falsche oder keine .gitignore in die zip verpackt
 - Konfiguration in src/assembly/src.xml
 - wenn ihr nur .gitignore hinschreibt, wird .gitignore aus jmjrst-Ordner genommen
 - die existiert bei euch vielleicht garnicht, oder ist default
 - in zip liegt keine oder falsche .gitignore
 - fix: ../../.gitignore statt .gitignore

Felix Bachmann - SWT1



Aufgabe 1 (Altsoftware vorbereiten)

- falsche oder keine .gitignore in die zip verpackt
 - Konfiguration in src/assembly/src.xml
 - wenn ihr nur .gitignore hinschreibt, wird .gitignore aus jmjrst-Ordner genommen
 - die existiert bei euch vielleicht garnicht, oder ist default
 - in zip liegt keine oder falsche .gitignore
 - fix: ../../.gitignore statt .gitignore
- LICENSE sollte in das Wurzelverzeichnis
 - siehe https://maven.apache.org/guides/introduction/ introduction-to-the-standard-directory-layout.html

14.05.2019



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

CheckStyle (und JavaDoc) auch in Tests benutzen





Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- CheckStyle (und JavaDoc) auch in Tests benutzen
- sinnvolle Definition der Gleichheit zweier Bilder
 - gleiche Dimensionen
 - UND gleiche Pixel-Werte



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- CheckStyle (und JavaDoc) auch in Tests benutzen
- sinnvolle Definition der Gleichheit zweier Bilder
 - gleiche Dimensionen
 - UND gleiche Pixel-Werte
- auch bei Drehung um 0° ist Überprüfung von Gleichheit nötig



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- CheckStyle (und JavaDoc) auch in Tests benutzen
- sinnvolle Definition der Gleichheit zweier Bilder
 - gleiche Dimensionen
 - UND gleiche Pixel-Werte
- auch bei Drehung um 0° ist Überprüfung von Gleichheit nötig
- assertEquals() reicht nicht aus, um Gleichheit der Bilder zu pr
 üfen

Lastenheft

Felix Bachmann - SWT1



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- CheckStyle (und JavaDoc) auch in Tests benutzen
- sinnvolle Definition der Gleichheit zweier Bilder
 - gleiche Dimensionen
 - UND gleiche Pixel-Werte
- auch bei Drehung um 0° ist Überprüfung von Gleichheit nötig
- assertEquals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- Ordner target/test wurde nicht erstellt
 - new File() erstellt keine Datei, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir() oder File.mkdirs())



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- CheckStyle (und JavaDoc) auch in Tests benutzen
- sinnvolle Definition der Gleichheit zweier Bilder
 - gleiche Dimensionen
 - UND gleiche Pixel-Werte
- auch bei Drehung um 0° ist Überprüfung von Gleichheit nötig
- assertEquals() reicht nicht aus, um Gleichheit der Bilder zu prüfen
- Ordner target/test wurde nicht erstellt
 - new File() erstellt keine Datei, sondern nur einen "pointer" auf einen Pfad (siehe File.createNewFile() oder File.mkdir() oder File.mkdirs())
- @Test(expected=XYException.class) nutzen, wenn Exception erwartet, sonst @Ignore

Orga



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- Stil: Konstanten für Pfade benutzen
 - magic Strings sind ähnlich böse wie magic numbers



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- Stil: Konstanten f
 ür Pfade benutzen
 - magic Strings sind ähnlich böse wie magic numbers
- Generator soll in @Before-Methode vor jedem Test neu erstellt werden
 - Objekt soll "frisch" sein
 - Zustand des Objektes kann sich durch Methoden-Aufrufe verändern



Aufgabe 2 + 3 (Modultests + Testüberdeckung)

- Stil: Konstanten f
 ür Pfade benutzen.
 - magic Strings sind ähnlich böse wie magic numbers
- Generator soll in @Before-Methode vor jedem Test neu erstellt werden
 - Objekt soll "frisch" sein
 - Zustand des Objektes kann sich durch Methoden-Aufrufe verändern
- niemals absolute Pfade verwenden
 - ich habe bei mir keinen Ordner
 - C:/Users/Manfred/Desktop/Uni/bloedes_SWT/eclipseWorkspace/
 - besser: relative Pfade
 - am besten: Resourcen über eingebaute Methoden suchen
 - this.getClass().getResource(<Datei-Name>)

 - siehe https://stackoverflow.com/questions/3861989/ preferred-way-of-loading-resources-in-java



- Was ist das?
- Weiß jemand die Phasen? :)



dokumentengetriebenes Prozessmodell



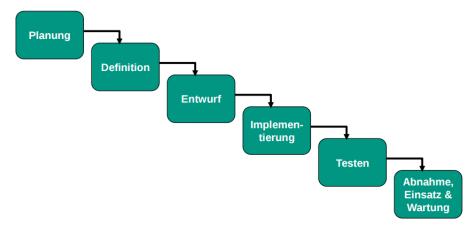


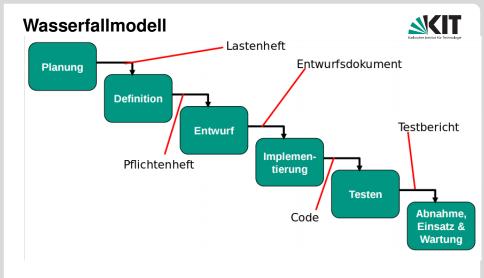
- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung

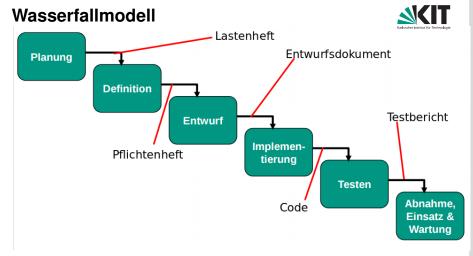




- dokumentengetriebenes Prozessmodell
- mögliche Phasen der Softwareentwicklung







Dokumente für das 2. ÜB:

- Lastenheft
- Durchführbarkeitsuntersuchung (weiteres Artefakt der Planung)

LAT⊨X

12/43

Orga Wasserfallmodell Durchführbarkeitsuntersuchung Lastenheft OO 0000 UML-Klassendiagramm



Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?



Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

Fachlich

Felix Bachmann - SWT1



Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

Fachlich (softwaretechnisch leicht realisierbar?)



Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

- Fachlich (softwaretechnisch leicht realisierbar?)
- 2 Alternativen



Grundlegende Frage

Ist das Projekt in dem jeweiligen Szenario überhaupt durchführbar?

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



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



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?)



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



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?)
- A Risiken (Gibt es Risiken? :D)



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

Pflichtenheft



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?)



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



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?)
- Rechtlich (Datenschutz, Standards)



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)

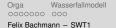
Fürs Übungsblatt

Denkt euch was (plausibles) aus!



Grundlegende Aufgabe

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





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)
- Funktionale Anforderungen (feingranular: Funktionen des Produkts)
- Produktdaten (Welche Daten speichern?)



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)
- Produktdaten (Welche Daten speichern?)
- Nichtfunktionale Anforderungen (Meta-Anforderungen: Zeit, Zuverlässigkeit)

14.05.2019



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)
- Produktdaten (Welche Daten speichern?)
- Nichtfunktionale Anforderungen (Meta-Anforderungen: Zeit, Zuverlässigkeit)
- Systemmodelle
 - Szenarien (spezielles Beispiel)
 - Anwendungsfälle (allgemeiner Verwendungszweck)



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)
- 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

Lastenheft



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 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 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)

0000000

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

Felix Bachmann - SWT1

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

14.05.2019



Produkteinsatz vs. Produktumgebung

Lastenheft



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.



Lastenheft



Wahr oder falsch?

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

19/43

Lastenheft



Wahr oder falsch?

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

Lastenheft



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.



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. falsch



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.

14.05.2019



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

14.05.2019



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.



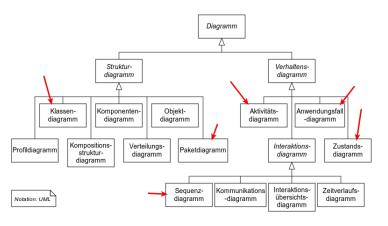
Wahr oder falsch?

- 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 = Unified Modeling Language
- grafische Modellierungssprache, strenge Syntax



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: Rassegewicht: int
- + wiegen(): int
- + streicheln()
- + streicheln(intensität: int, ausruf: String)
- + füttern(ration: Nahrung)

Vererbung



ParentClass

+publicString: String

-privateInt: int

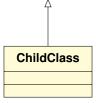
#protectedDouble: double

+staticMethod()

+publicMethod(): String

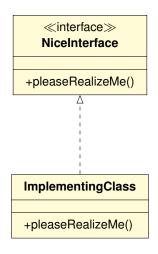
-privateMethod(): int

#protectedMethod(param: String): double



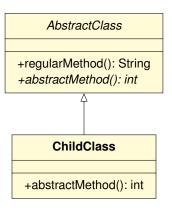
Interface





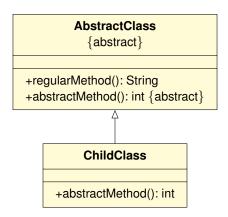
Abstrakte Klassen





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	*	

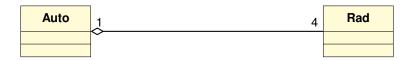
0000000

28/43

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!

Felix Bachmann - SWT1

Klausuraufgabe SS11



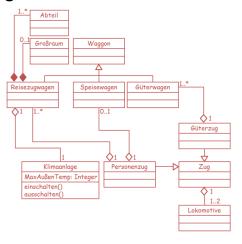
Modellieren Sie das Szenario möglichst vollständig als UML-Klassendiagramm. Geben Sie Methoden, Attribute, Multiplizitäten, Restriktionen, Assoziationsnamen, Aggregationen und Kompositionen sowie Rollen an.

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.

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

Felix Bachmann - SWT1

LATEX- Basics



- 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

Loven ipoun dolor at anys, conservar sadiporing elits; sed dism conseny signed tempor invidual hipsatum and delentangue data delere in frequit mills find bit. Lovers ignore deler sits merc, consecutation adjacating dit, sed dam necumenty raths extended in citizen at factored deleter magna. Ut whi mins of minim venion, quis notend exert inton olimo representații (obortă nial er alușuje et ea commodu consequet. Dan anem vel eun trius delar in headert ur valganes esti-cee moletie renorquet, vel il un dolore ea fragate sola fecilio at ven-ero et acrumom et intoi-cido diginism qui blandir present lupram ural delent seque duit dalars se fragat nial a facilia. Numbber rempor cum substancins eletiend opton conque salul ampenire donung al quod maxim From their tempor case solute motor elettered option compare shift inspection downing at quod maxims places the fore position assume. Leaves forum dokes six meet, consecutaves ellipsicing whit, and claim nonamany and the submood translates at housest dokes mangas adaptates near volumps. It waste reals maxim versions, quin moused on seri auton submoorages suscepti loborate sixil at adaptate or ea Date states well sum intere delor in bendwerk in valgenate wells esse molecule consequet, vel (Burn dolors en females suille facilisis. At veso sec et accusam et jum duo delares et su rebum. Suo clim kasel gubergren, na sea micina success set Lorent (prime dolor isk amer. Loom (prime dolor isk amer, conservine subprime sile, set dam normer ektered impre involute is labore et dolore magne allegoven ent, sed dam vollagiene. Al vero ent et accusates et prime duo dolores et en rebur. I bet chis had gabergen, en sea sadon au sancius est Lorent (prime dolor isk amer. Loores (prime dolor isk amer, conserver cultiporine et a...). accusan surgiciam cum maim nome nomes nomes no recording recording part of set briefdant junio aldron Sier, chia e et gibbrighten, kind mag na no odrom, senticio no se el sidimente set veco rologiana, set Loreis poum dolor di sinet. Loreis guaran dolor di mete, romaneste subjusting chia, sed chian nominare desino di sespe i involució si above et dolorei magos alloquena este. Comercus sudiporing elles, sed disen nommery circund tempor invident at labore et dolors magna alsays server, sed disen volupusa. As versi eco et acrossos et junca dos dolors et es relass. Ser class land euberose, no seu sidairon aucross et l'across insum dolor si name. Lores insum dolor si

(b) What You Get

Larven ipseum delier sit anner, consenter sudipseing eller, und diam nonamy eirmod tempor invident at labore et deliere magne altepyam esse, und diam vellupsas. As vers ens et accusans et jonis diam deliere et un teleste sect cita land giberpepts, on on taliantate saccase out. Larven (poum delier de serve. Loren ipseum cirice si a med, consenter sedelering elle, un deliam noname crimed serger.

Date autom vel eum inturr dolor in benderek in volgutate velit eue melende comoquer, vel illem dolore eu freutat nella facilisie.

Controlled Stage of the Contro



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







- 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 :(

Felix Bachmann - SWT1

14.05.2019

LATEX- Installation



Installation einer Distribution notwendig, z.B.:

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

Lastenheft

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
- ...

14.05.2019

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{...})

Lastenheft

38/43

LATEX- Dokumentaufbau



38/43

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,...

```
% preamble
\begin{document}
      content
\end{document}
```

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

LATEX-Beispiel!

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

Lastenheft

- 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",...)

Lastenheft

0000000



Aufgabe 4 + 5: HDrize

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



Durchführbarkeitsuntersuchung

Pflichtenheft

Lastenheft

UML-Klassendiagramm



Aufgabe 4 + 5: HDrize

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

Aufgabe 6: Schnittstellen für Filter

- nur Schnittstellen definieren
- JavaDoc!

Felix Bachmann - SWT1

Lastenheft

Denkt dran!



Abgabe

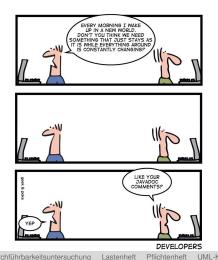
- Deadline am 22.5 um 12:00
- Dokumente ausdrucken
- Klassendiagramme handschriftlich
- Deckblatt!

Lastenheft

14.05.2019

Bis dann! (dann := 28.05.19)





Orga