

# Tutorium Softwaretechnik I

Moritz Klammler | 30. Mai 2017

Fakultät für Informatik, IPD Tichy



Titelfoto: Copyright (C) 2010 Multimond, CC BY-SA 3.0

- In Zukunft bei Mail-Abgaben bitte immer die Übungsleiter (`swt1@ipd.kit.edu`) auf Cc setzen. (Und damit rechnen, dass sie Eure Abgabe nicht akzeptieren werden.)

- In Zukunft bei Mail-Abgaben bitte immer die Übungsleiter (swt1@ipd.kit.edu) auf Cc setzen. (Und damit rechnen, dass sie Eure Abgabe nicht akzeptieren werden.)
- Beim Klassendiagramm (in der Klausur) immer alle drei Kästen zeichnen (auch die leeren).

- In Zukunft bei Mail-Abgaben bitte immer die Übungsleiter (swt1@ipd.kit.edu) auf Cc setzen. (Und damit rechnen, dass sie Eure Abgabe nicht akzeptieren werden.)
- Beim Klassendiagramm (in der Klausur) immer alle drei Kästen zeichnen (auch die leeren).
- Wie schreibe ich (in der Klausur) handschriftlich kursiv?
  - ✗ *AbstractWidget*
  - ✗ /AbstractWidget/
  - ✓ {abstract} AbstractWidget

- Übungsblatt 1
- Definitionsphase (*analysis*)
- UML-Diagramme
- Objektmodellierung

# Übungsblatt 1 – Häufige Probleme

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`

# Übungsblatt 1 – Häufige Probleme

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).

# Übungsblatt 1 – Häufige Probleme

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- ✍ Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.



- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- ✍ Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- ✍ Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- ✍ Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- ✍ Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.
- Instanzvariablen explizit mit `null` zu initialisieren ist unnötig.

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- 📖 Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- 📖 Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.
- Instanzvariablen explizit mit `null` zu initialisieren ist unnötig.
- 📖 Wie verwende ich Branches in Git?

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/.m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- 📖 Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- 📖 Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.
- Instanzvariablen explizit mit `null` zu initialisieren ist unnötig.
- 📖 Wie verwende ich Branches in Git?
- 📖 Wie erstelle und applye ich einen Patch?

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/.m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- 📖 Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- 📖 Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.
- Instanzvariablen explizit mit `null` zu initialisieren ist unnötig.
- 📖 Wie verwende ich Branches in Git?
- 📖 Wie erstelle und applye ich einen Patch?
- Wie teste ich Identitäten intelligent?

- Niemand hat das Parent-Projekt korrekt definiert.  
→ Maven-Cache is evil! Hint: `~/ .m2/`
- Auf Ressourcen immer mittels `ClassLoader` zugreifen (am Besten via `getResourceAsStream`).
- 📖 Unit-Test-Framework ausnutzen. Nicht selbst Control-Flow implementieren.
- 📖 Exception-Handling: Bitte nur `catchen`, wenn man auch etwas Sinnvolles zur Fehlerbehandlung beitragen kann. Ansonsten eskalieren lassen. `printStackTrace()` ist in aller Regel *keine* sinnvolle Fehlerbehandlung.
- Instanzvariablen explizit mit `null` zu initialisieren ist unnötig.
- 📖 Wie verwende ich Branches in Git?
- 📖 Wie erstelle und applye ich einen Patch?
  - Wie teste ich Identitäten intelligent?
  - Wie teste ich undokumentierten Code sinnvoll?

# Definitionsphase (*analysis*) – Wahr oder Falsch?

Das Lastenheft ist ein Artefakt der Definitionsphase.

# Definitionsphase (*analysis*) – Wahr oder Falsch?

✗ Das Lastenheft ist ein Artefakt der Definitionsphase.



# Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.  
Das Pflichtenheft kann UML-Diagramme enthalten.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.

# Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.  
Von einer Klasse kann es Null oder mehr Instanzen geben.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.

# Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.  
Die Instanzen einer Klasse unterscheiden sich nicht.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.  
„Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.



## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.  
Eine Klasse legt fest, welche Attribute ihre Instanzen haben.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
  - ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
  - ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
  - ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
  - ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
  - ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.

Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.
- ✗ Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.
- ✗ Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.

Wenn sich zwei Objekte im selben Zustand befinden, sollte ihr nach außen sichtbares Verhalten nicht unterscheidbar sein.

# Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.
- ✗ Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.
- ✓ Wenn sich zwei Objekte im selben Zustand befinden, sollte ihr nach außen sichtbares Verhalten nicht unterscheidbar sein.



## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.
- ✗ Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.
- ✓ Wenn sich zwei Objekte im selben Zustand befinden, sollte ihr nach außen sichtbares Verhalten nicht unterscheidbar sein.  
Der Zustand eines Objekts sollte von außen nur Methoden verändert werden können.

## Definitionsphase (*analysis*) – Wahr oder Falsch?

- ✗ Das Lastenheft ist ein Artefakt der Definitionsphase.
- ✓ Das Pflichtenheft kann UML-Diagramme enthalten.
- ✓ Von einer Klasse kann es Null oder mehr Instanzen geben.
- ✗ Die Instanzen einer Klasse unterscheiden sich nicht.
- ✓ „Instanzen“ werden in der Vorlesung auch „Exemplare“ genannt.
- ✓ Eine Klasse legt fest, welche Attribute ihre Instanzen haben.
- ✗ Wenn zwei Instanzen derselben Klasse in allen Attributen übereinstimmen, sind sie identisch.
- ✗ Der Zustand eines Objekts ist ein Element aus dem kartesischen Produkt der Wertebereiche seiner Attribute.
- ✓ Wenn sich zwei Objekte im selben Zustand befinden, sollte ihr nach außen sichtbares Verhalten nicht unterscheidbar sein.
- ✓ Der Zustand eines Objekts sollte von außen nur Methoden verändert werden können.

# Diskussion: Sinnvolle Vererbungsbeziehungen

- UML-Klassendiagramme (*class diagram*)
- UML-Aktivitätsdiagramme (*activity diagram*)
- UML-Sequenzdiagramme (*sequence diagram*)
- UML-Zustandsdiagramme (*state machine*)
- UML-Paketdiagramme

Asterix bemerkt, dass sein Dorf von einer römischen Legion angegriffen wird, und macht sich bereit, es zu verteidigen. Dazu will er sich zunächst an einem Schluck Zaubertrank stärken. Er besucht daher den Druiden Miraculix, der als Einziger diesen magischen Trank brauen kann. Miraculix bittet Asterix in seine Hütte und flößt ihm eine Schöpfkelle des Zaubertranks ein, den er noch in einem Kessel neben dem Feuer warm gehalten hat. Asterix nimmt den Trank zu sich. Sodann vergewissert er sich anhand eines Hinkelsteins, dass der Trank seine gewünschte Wirkung nicht verfehlt hat. Nachdem er diesen mühelos mit einer Hand stemmen kann, verschwendet er keine Zeit mehr und stürmt vor das Dorf, um die Römer zu empfangen.