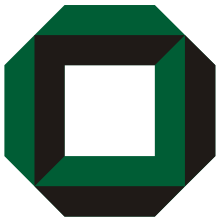


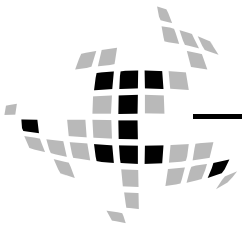
Universität Karlsruhe (TH)

Forschungsuniversität · gegründet 1825

Softwaretechnik 1 Tutorium

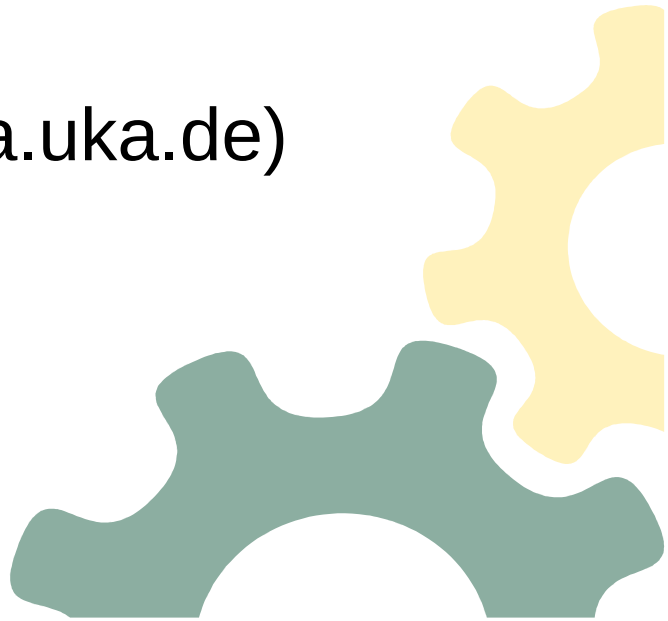
11. Mai 2009

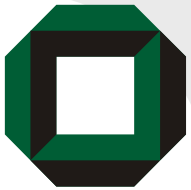
Matthias Thoma (s_thoma@ira.uka.de)



Fakultät für **Informatik**

Lehrstuhl für Programmiersysteme

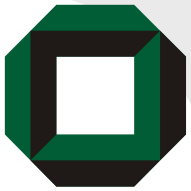




Wer ist der komische Kerl?

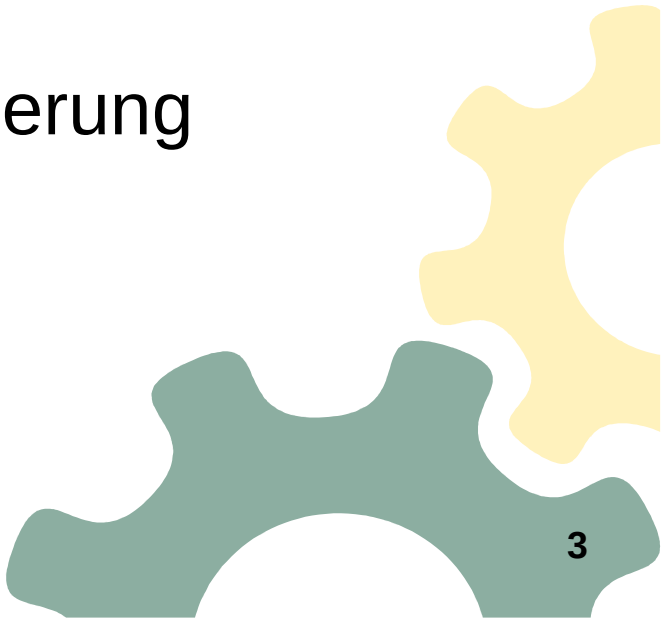
- Matthias Thoma (s_thoma@ira.uka.de)
- Vertiefungsgebiete:
  - Softwaretechnik und Übersetzerbau
  - Parallelverarbeitung
- Diplomarbeit (Forschungszentrum Informatik)

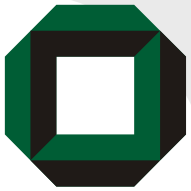
  „Methodik zur Modellierung paralleler eingebetteter Software unter Berücksichtigung der zugrundeliegenden Architektur“



Heute

- Organisatorisches
- Rückgabe der Übungsblätter
- Fragen
- Namenskonventionen
- JAR und ZIP
- Internationalisierung und Lokalisierung
- UML Anwendungsfalldiagramme

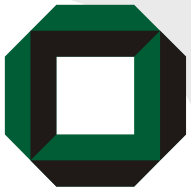




Organisatorisches

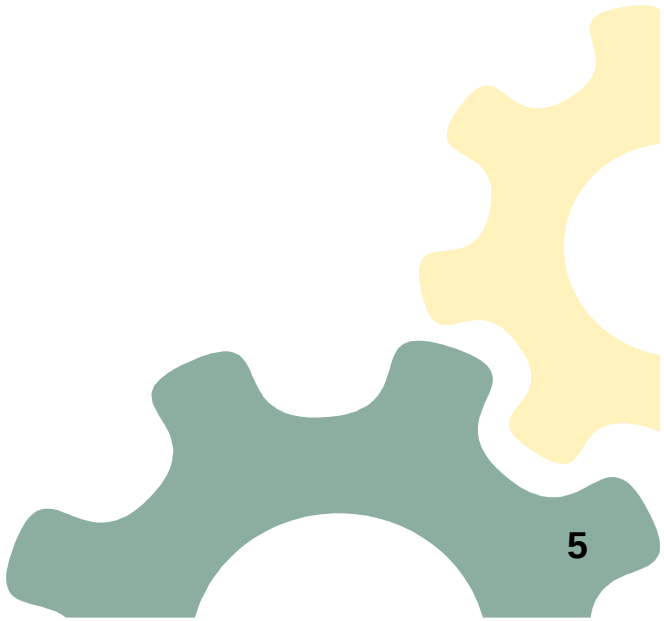
- Abgabe nur einzeln.
- Abgabe von getippten Blättern erlaubt und erwünscht.
- Es gibt auch kostenlose UML Programme (z. B. http://jude.change-vision.com)
- Die Folien gibt’s unter

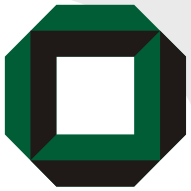
  http://www.stud.uni-karlsruhe.de/~ucys/swttut



Mehr Organisatorisch

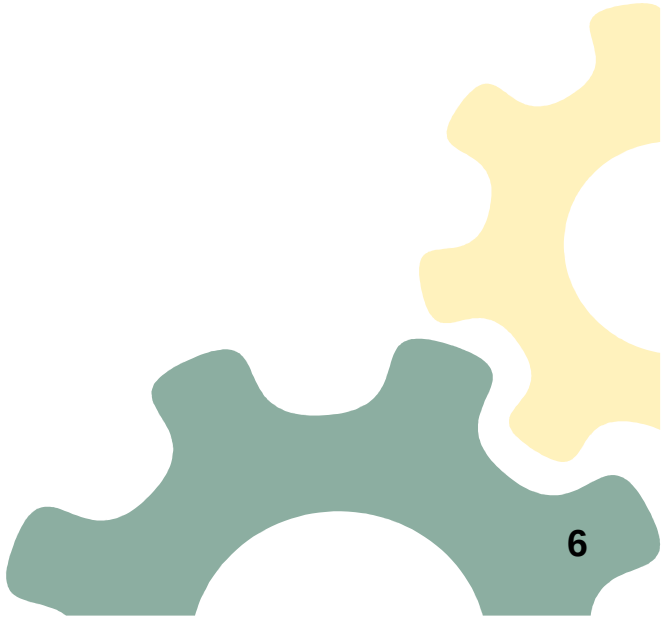
- Täuschungsversuch wird mit Abzug der Punktzahl des Übungsblattes geahndet.
- Verstöße gegen die Richtlinien für die Einsendung von Lösungen führen zu 0 Punkten in der entsprechenden Aufgabe.

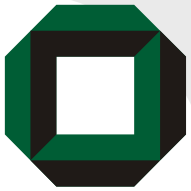




Unerfreuliches

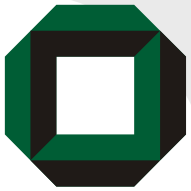
- 12 Betrugsversuche beim ersten Übungsblatt (geahndet mit -24 Punkten)
- 11 mal 25% Abzug wegen falschem Format (RAR, tar etc. statt ZIP!)
- 7 mal 0 Punkte wegen nicht übersetzbaren Programmen





Noch mehr Organsiatorisches

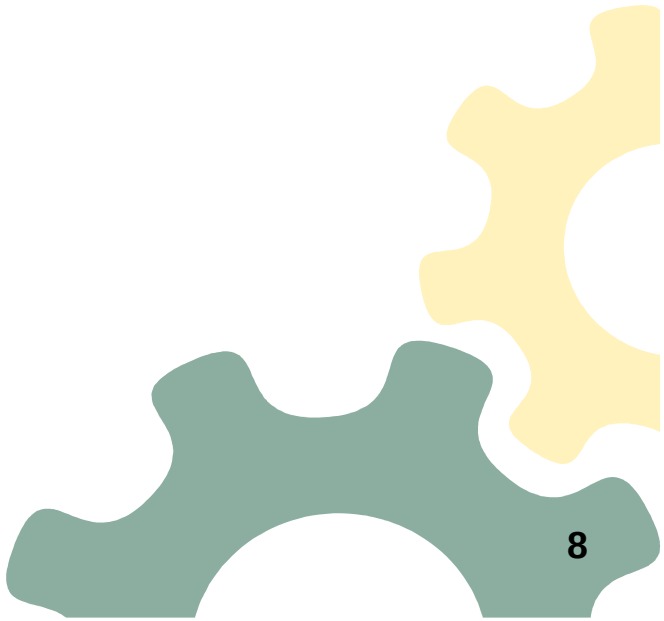
- Abgabe von Programmieraufgaben zukünftig:
  - Als ausführbares JAR (mehr dazu gleich)
  - Quellcode zusätzlich als ZIP Datei (und damit ist wirklich ZIP gemeint, nicht RAR oder 7zip)
  - Bitte nur ASCII verwenden
  - Sich ganz genau an die Namenskonventionen für die Dateinamen des JAR und des ZIP halten!
  - Sonst: Punktabzug bis zu 100%

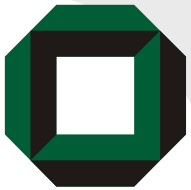


Voraussichtliche Termine 2009

Alle 14-Tage, immer in der Woche nach der großen Donnerstag-Saalübung (Feiertage beachten!)

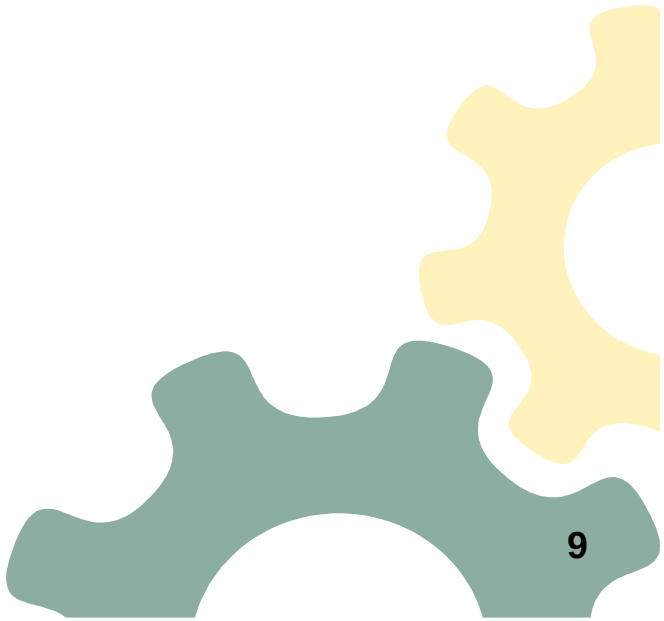
- 11. Mai
- 25. Mai
- 08. Juni
- 22. Juni
- 06. Juli
- 20. Juli

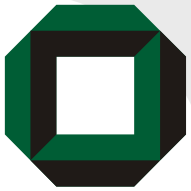




Fragen?

Fragen zum Übungsblatt?



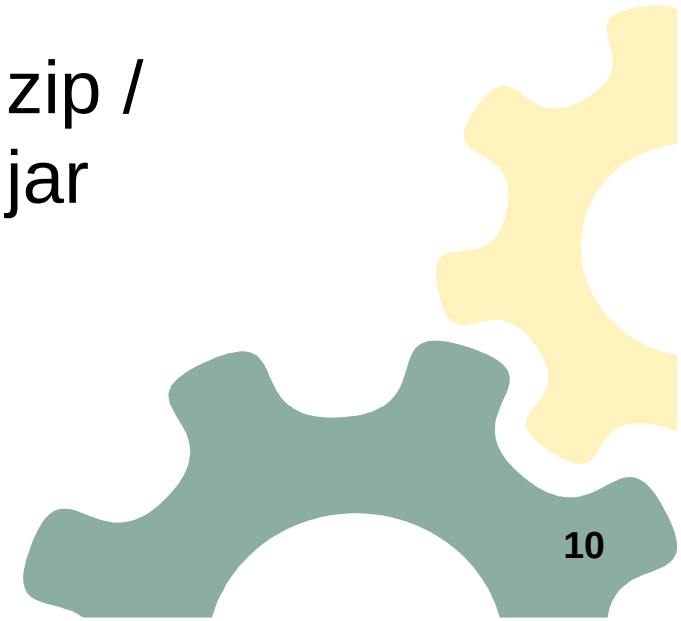


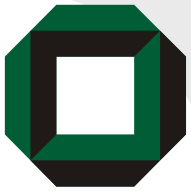
Namenskonventionen

- Bitte zukünftig alle JAR und ZIP Dateien wie folgt benennen! Fehlerhafte Dateinamen führen zum Punktabzug (bis zu 100%).

```
<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.jar
<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.zip
```

01_1234567_Max_Mustermann.zip /
01_1234567_Max_Mustermann.jar



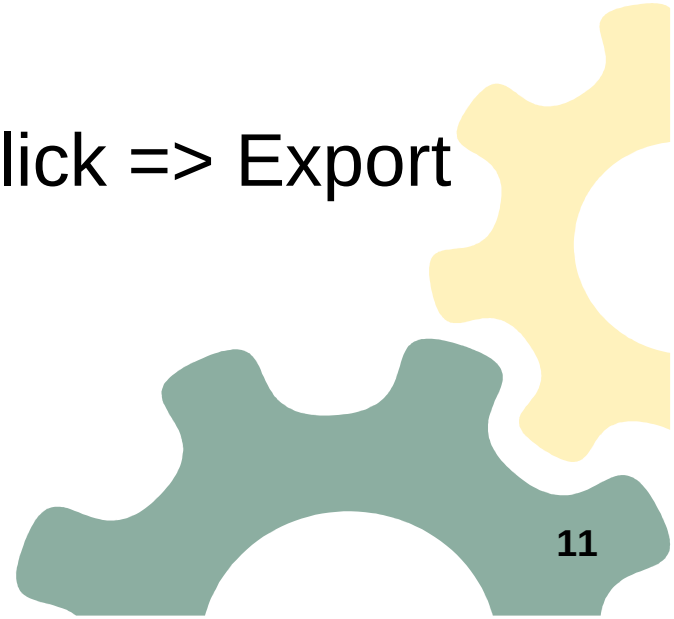


Zu einem ausführbaren JAR und einem ZIP in 7 einfachen Schritten

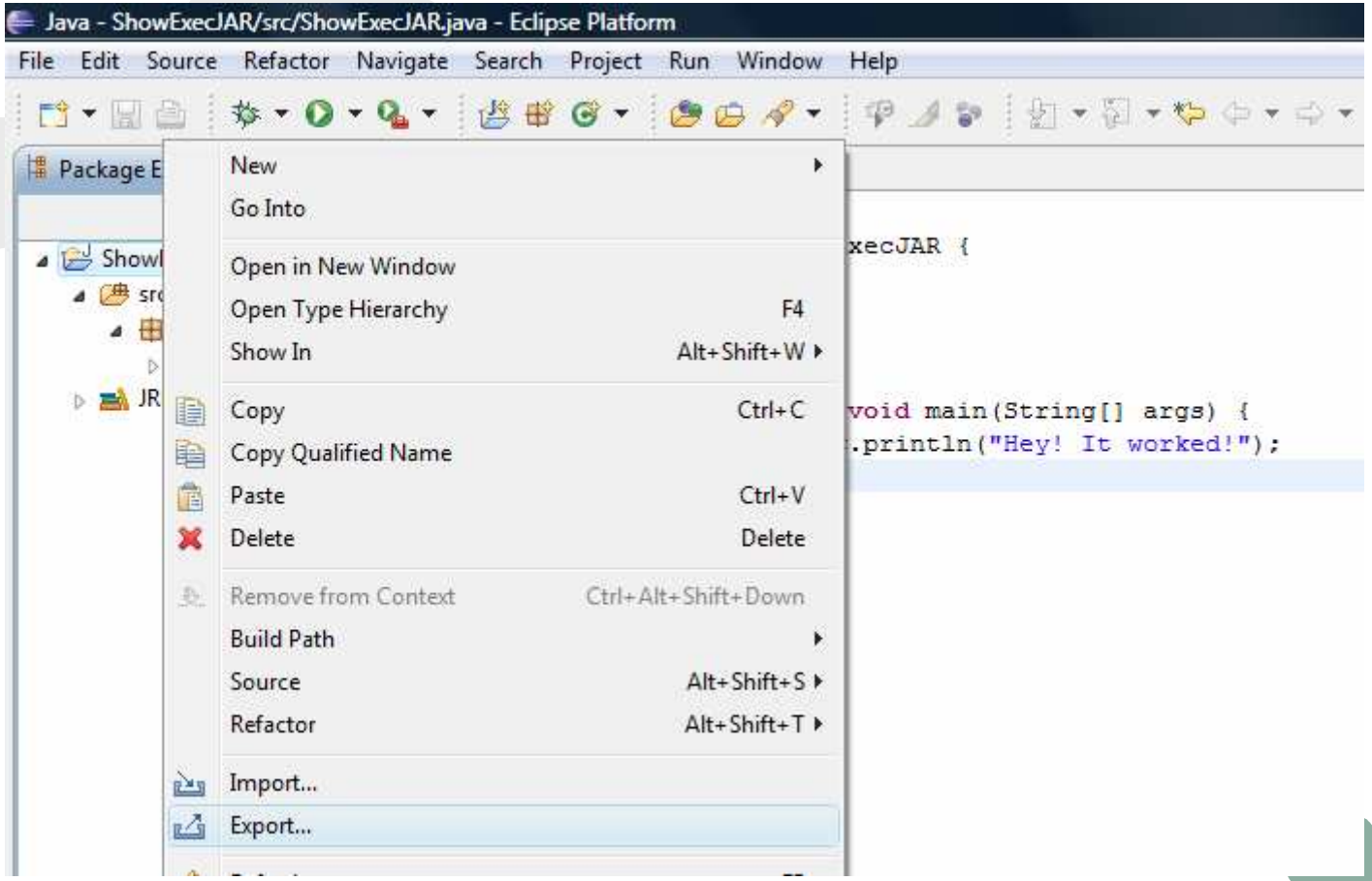
Alle Erläuterungen beziehen sich auf das letzte Stable-Release von Eclipse: „Ganymede“

Schritt 1: Programm mindestens einmal ausführen, damit Eclipse eine mögliche Launch Configuration erstellt.

Schritt 2: Projektname => Rechtsklick => Export

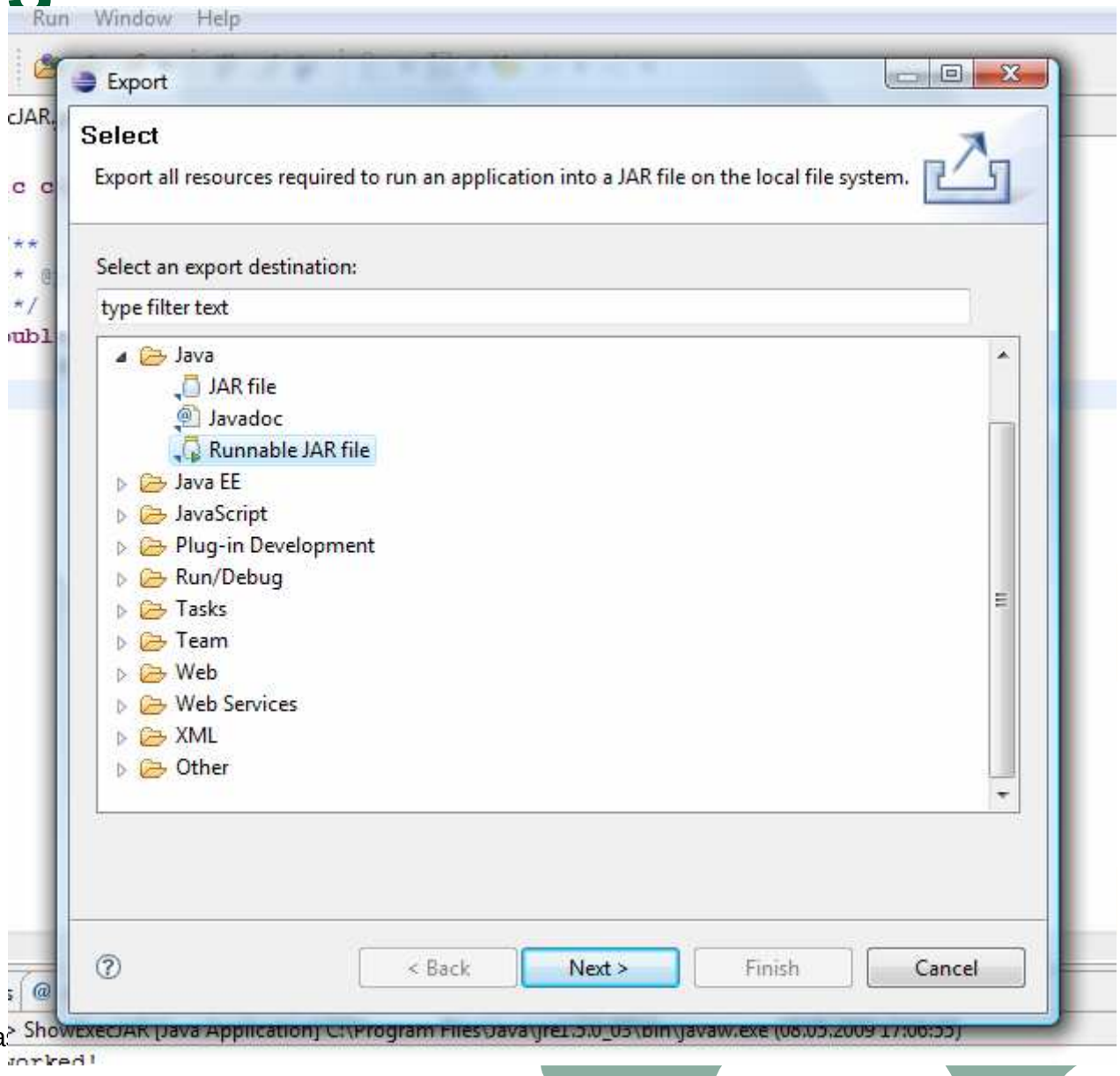


Schritt 2

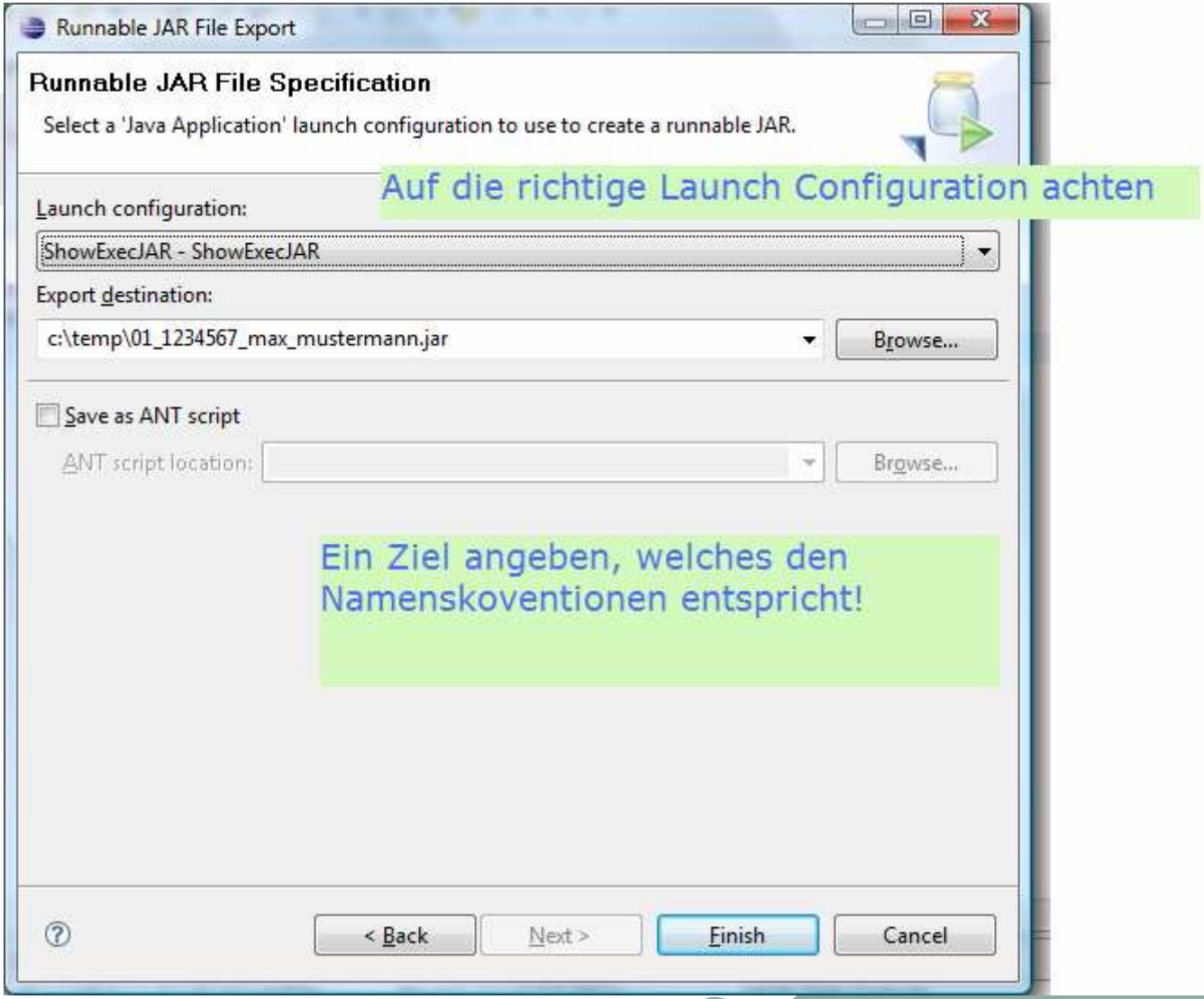

Java - ShowExecJAR/src/ShowExecJAR.java - Eclipse Platform
File Edit Source Refactor Navigate Search Project Run Window Help
Package E
New
Go Into
Open in New Window
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
xecJAR {
void main(String[] args) {
.println("Hey! It worked!");

Schritt 3

- Runnable JAR file wählen
- Next

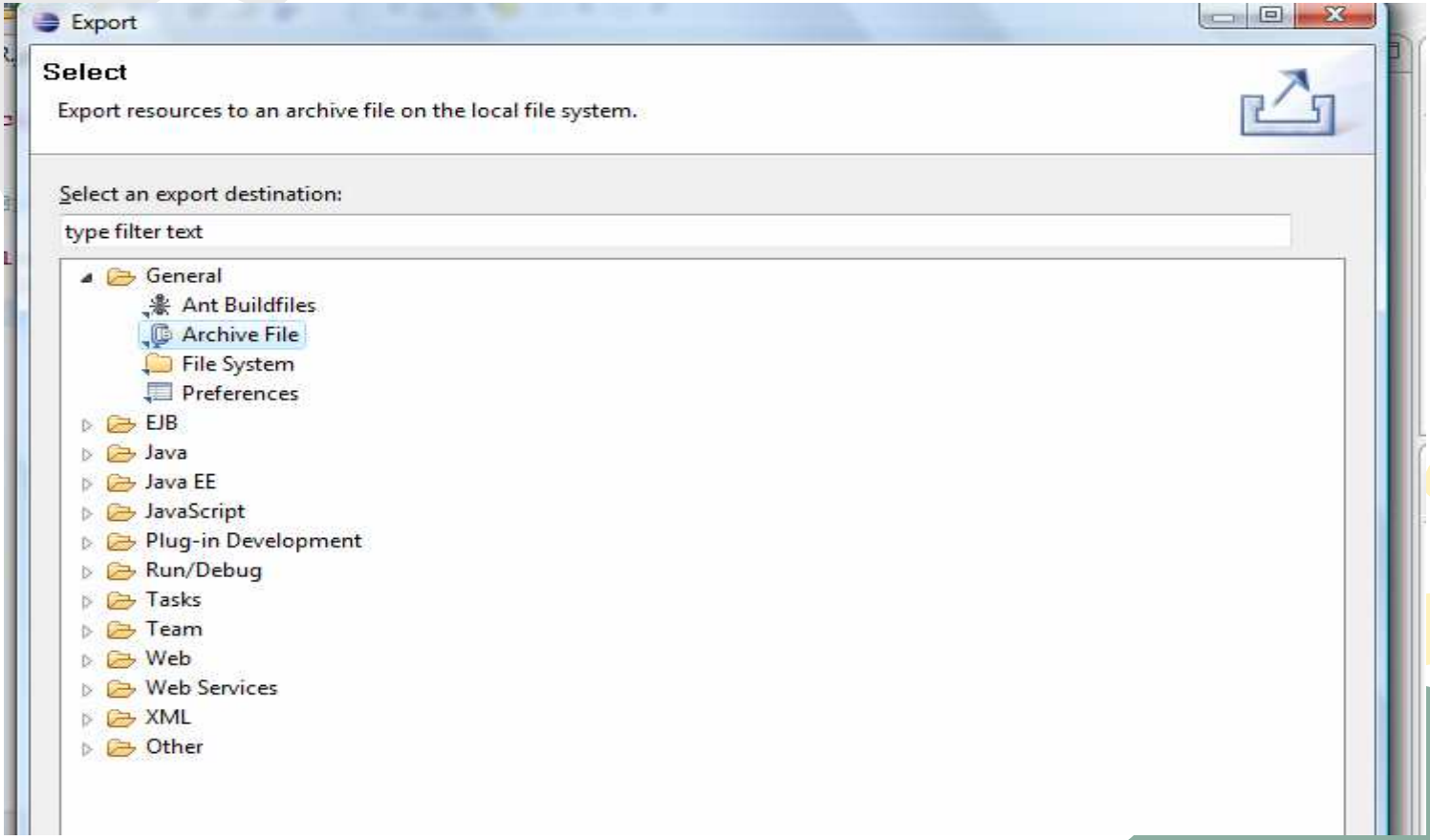


Schritt 4

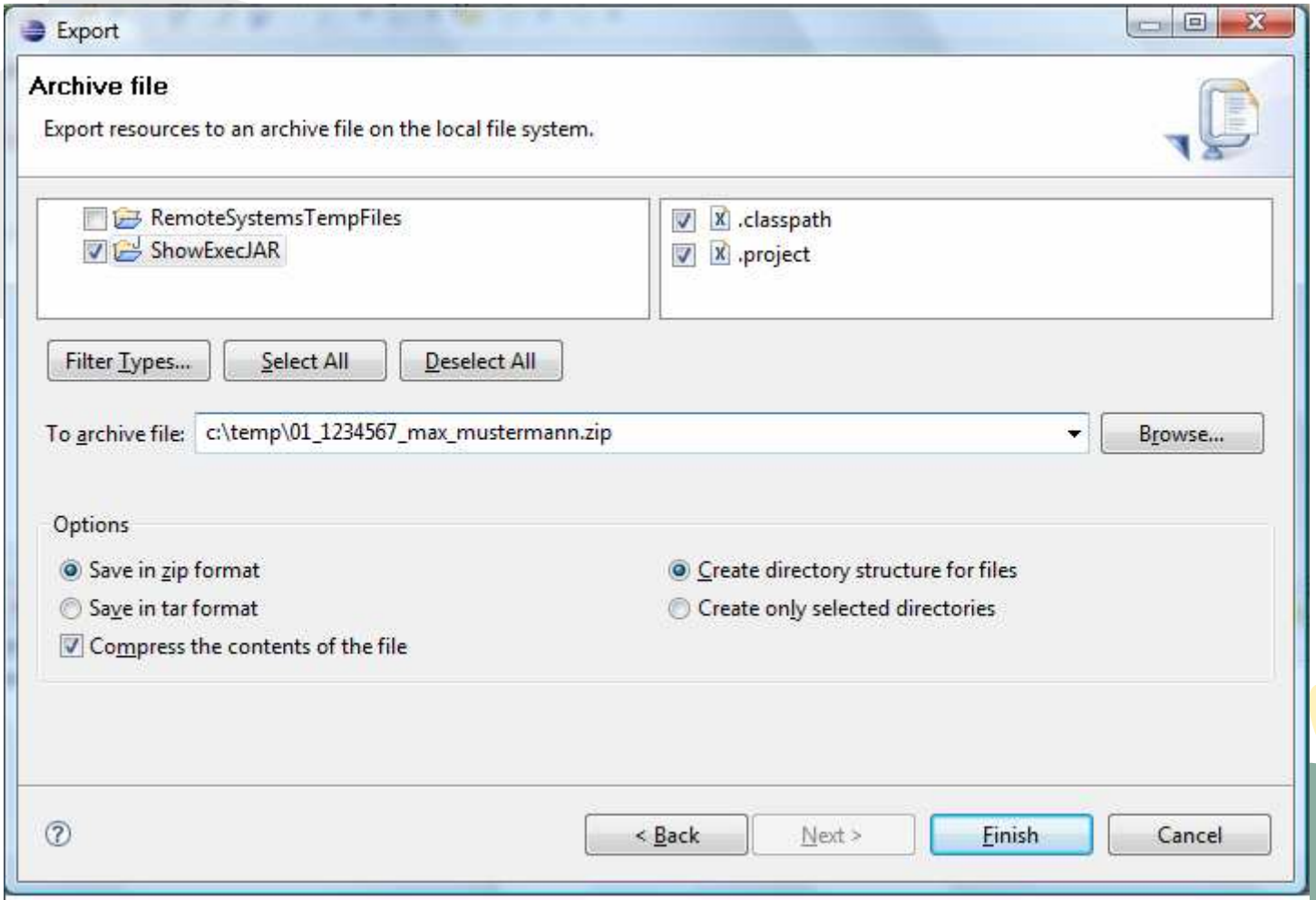

Runnable JAR File Export
Runnable JAR File Specification
Select a 'Java Application' launch configuration to use to create a runnable JAR.
Auf die richtige Launch Configuration achten
Launch configuration:
ShowExecJAR - ShowExecJAR
Export destination:
c:\temp\01_1234567_max_mustermann.jar
Browse...
Save as ANT script
ANT script location:
Browse...
Ein Ziel angeben, welches den Namenskoventionen entspricht!
< Back
Next >
Finish
Cancel

Schritt 5

- Projektname => Rechtsklick => Export

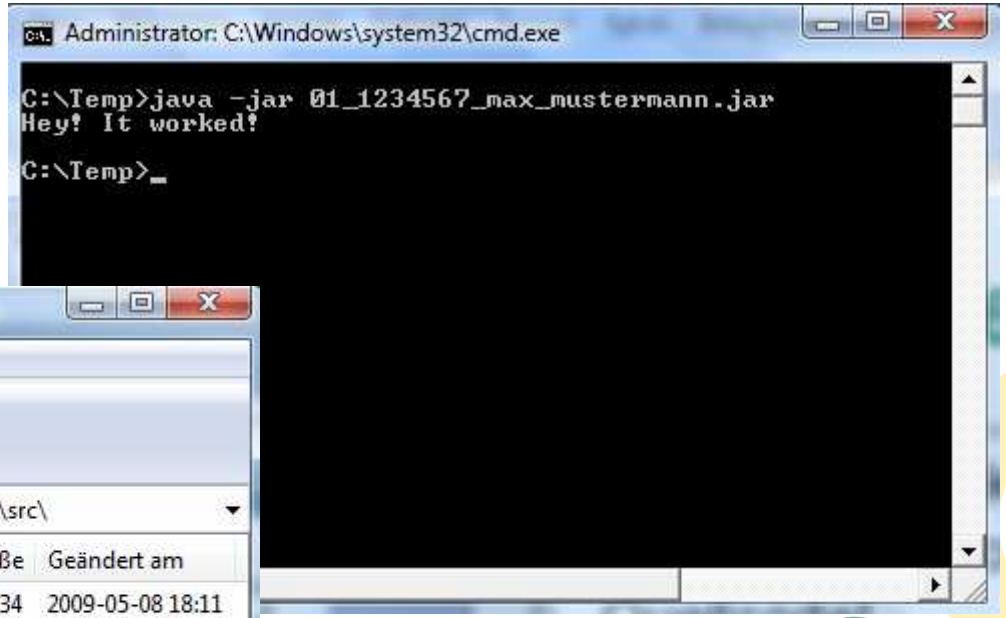


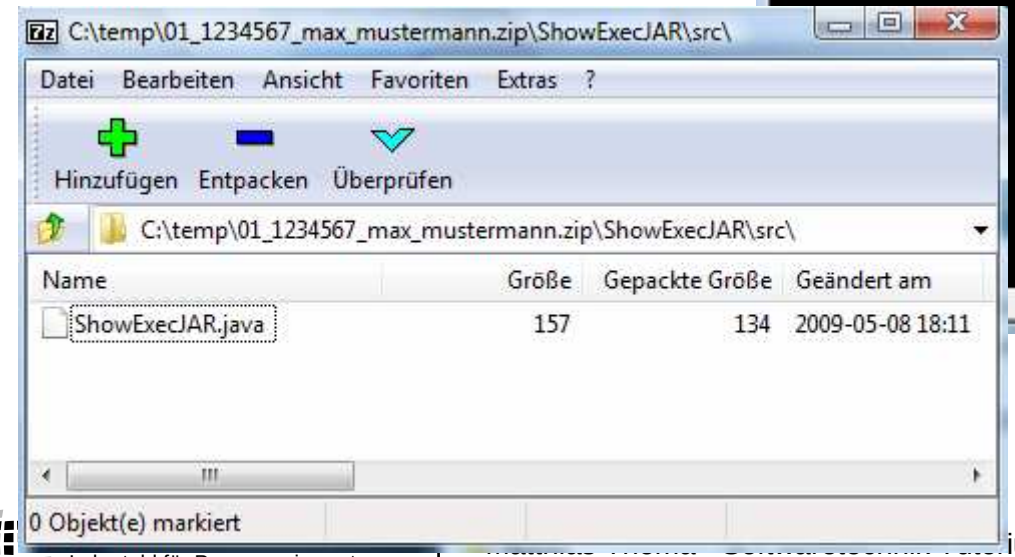
Schritt 6

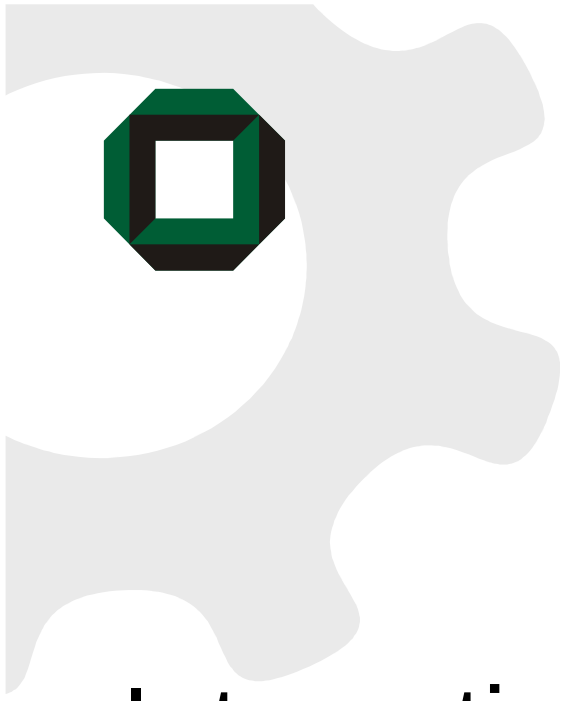

Export
Archive file
Export resources to an archive file on the local file system.
RemoteSystemsTempFiles
ShowExecJAR
.classpath
.project
Filter Types...
Select All
Deselect All
To archive file:
c:\temp\01_1234567_max_mustermann.zip
Browse...
Options
Save in zip format
Save in tar format
Compress the contents of the file
Create directory structure for files
Create only selected directories
< Back
Next >
Finish
Cancel

Schritt 7 – Testen!

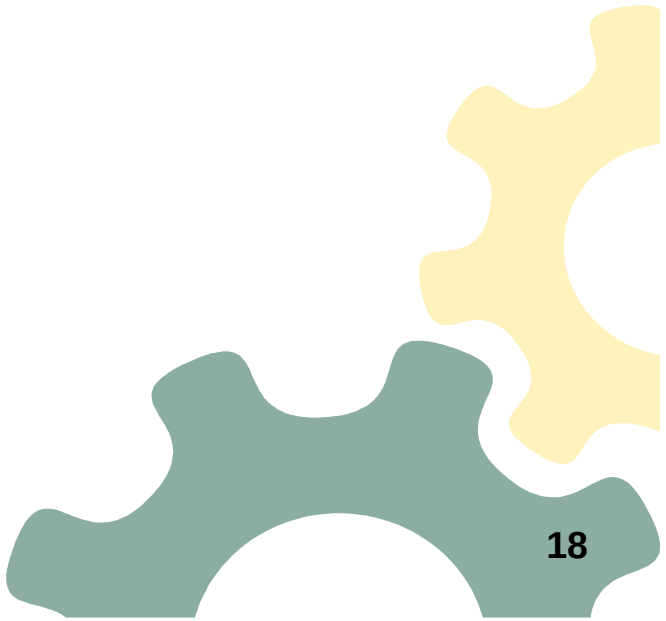
- Bitte keine Punkte verschenken durch nicht ausführbare Programme oder fehlenden Quellcode!
- Ins Quellcode ZIP schauen!

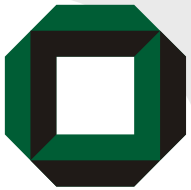






Internationalisierung und Lokalisierung

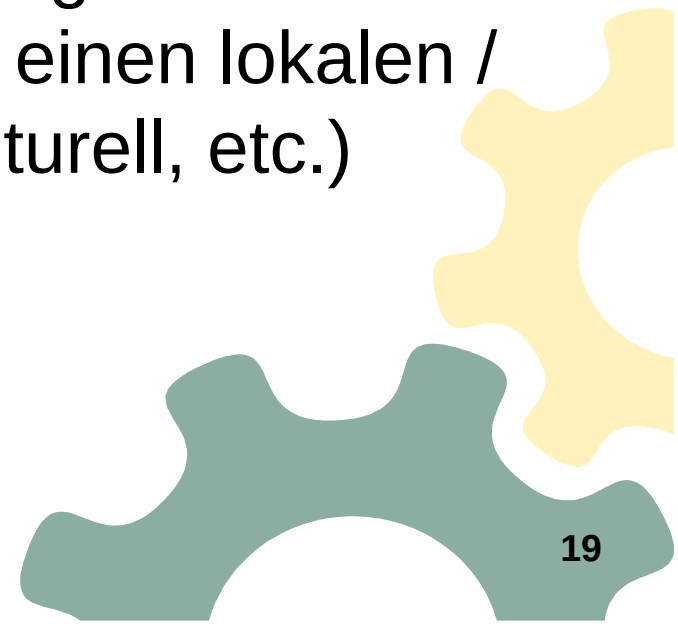


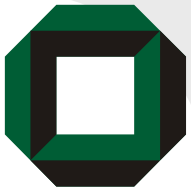


Internationalisierung und Lokalisierung

Internationalisierung: Gestaltung einer Software in der Art, dass es leicht an lokale/regionale Märkte angepasst werden kann.

Lokalisierung: Konkrete Anpassung einer internationalisierten Software an einen lokalen / regionalen Markt (sprachlich, kulturell, etc.)

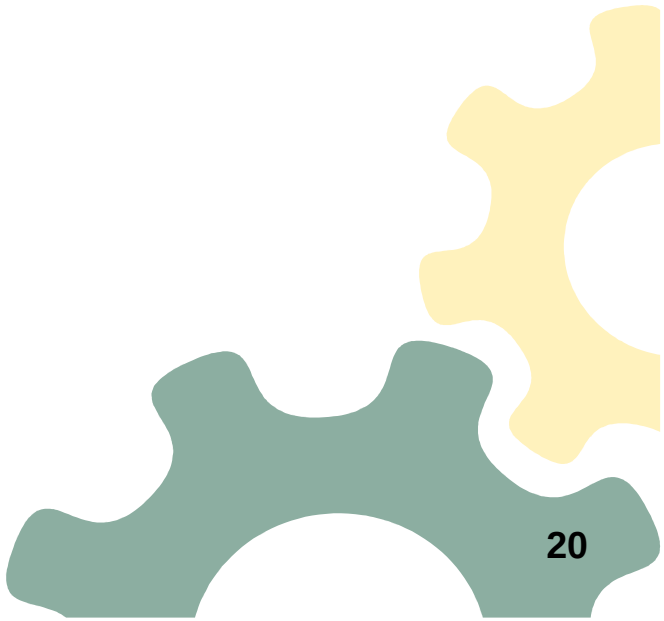


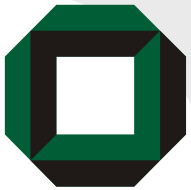


Internationalisierung und Lokalisierung

Lokalisierung:

- Sprache (Deutsch/Englisch/Französisch etc.)
- Datum-/Zeitformat
- Farbe
- Grafiken
- (u.v.m.)





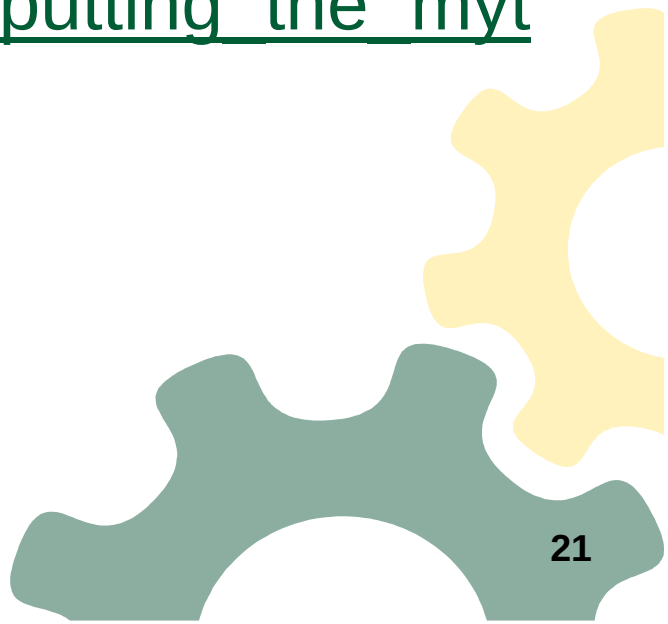
Internationalisierung und Lokalisierung

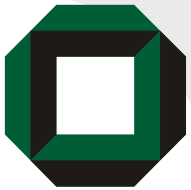
Lesehinweis:

http://java.sun.com/docs/books/tutorial/i18n

Lesetipp:

http://blogs.sun.com/i18ngal/entry/putting_the_myths_together_what

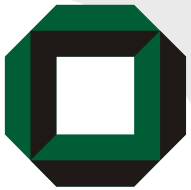




Internationalisierung und Lokalisierung

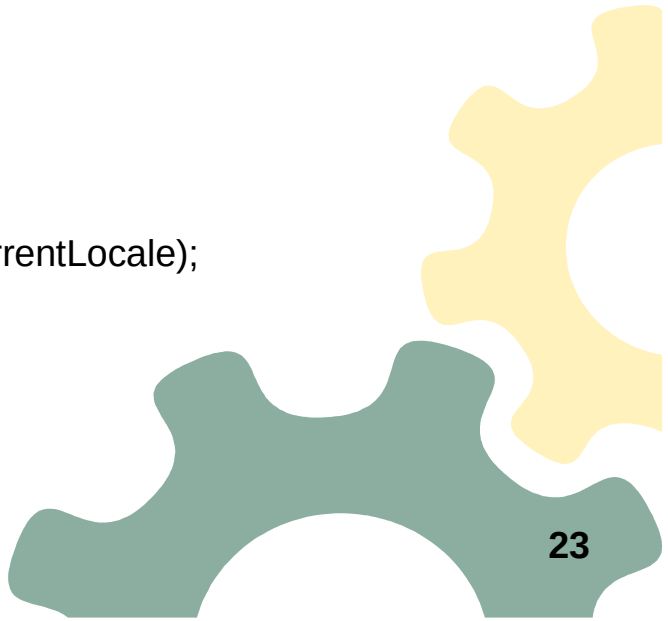
```
public class NotI18N {
    static public void main(String[] args) {
        System.out.println("Hello.");
        System.out.println("How are you?");
        System.out.println("Goodbye.");
    }
}
```

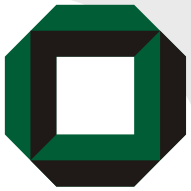
Die Beispiele zur Internationalisierung und Lokalisierung orientieren sich an:
http://java.sun.com/docs/books/tutorial/i18n/



Internationalisierung und Lokalisierung

```
public class I18NSample {

    static public void main(String[] args) {
        String language;
        String country;
        if (args.length != 2) {
            language = new String("en"); country = new String("US");
        } else {
            language = new String(args[0]); country = new String(args[1]);
        }

        Locale currentLocale;
        ResourceBundle messages;

        currentLocale = new Locale(language, country);

        messages = ResourceBundle.getBundle("MessagesBundle", currentLocale);
        System.out.println(messages.getString("greetings"));
        System.out.println(messages.getString("inquiry"));
        System.out.println(messages.getString("farewell"));
    }
}
```





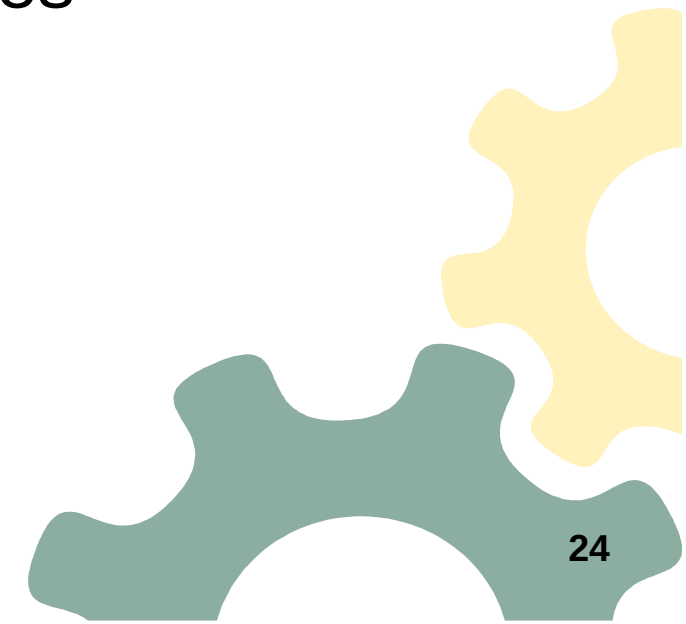
Internationalisierung und Lokalisierung

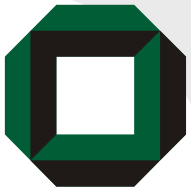
MessageBundle.properties

```
greetings = Hello.
farewell = Goodbye.
inquiry = How are you?
```

MessagesBundle_de_DE.properties

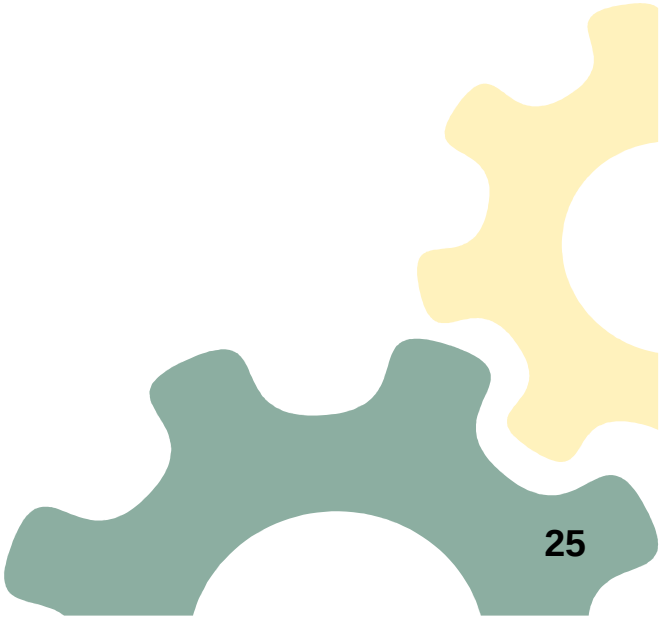
```
greetings = Hallo.
farewell = Tschüß.
inquiry = Wie geht es?
```

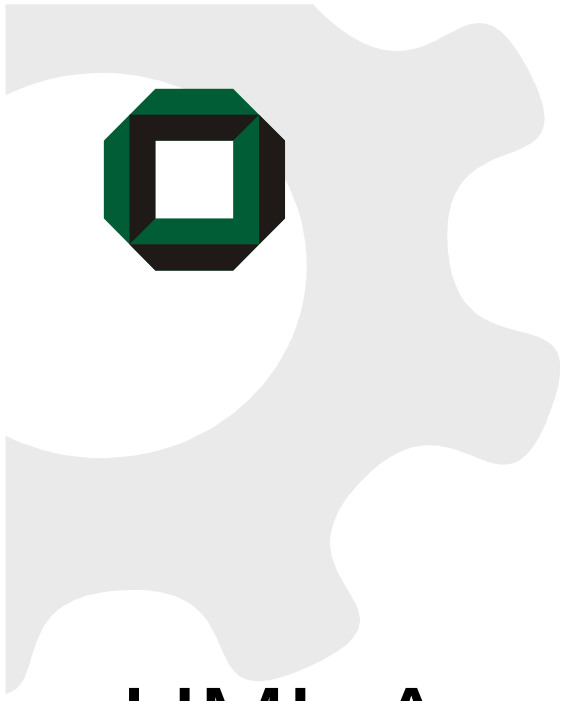




Internationalisierung und Lokalisierung

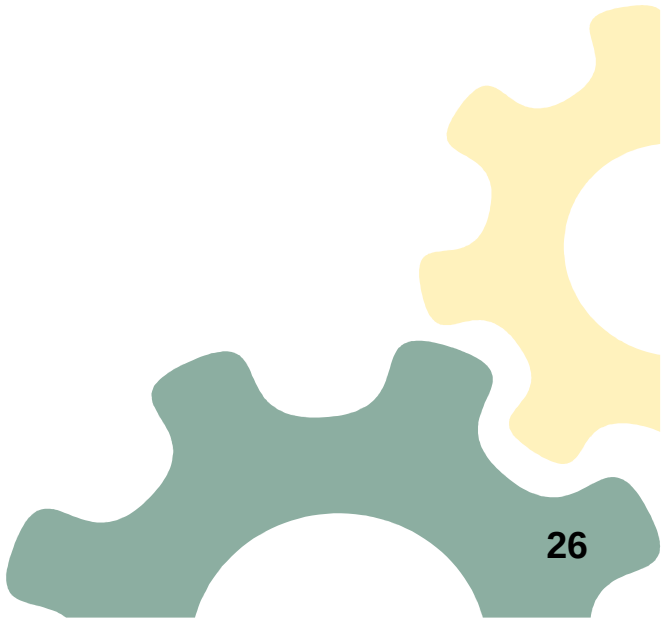
```
% java I18NSample fr FR
Hallo.
Wie geht es?
Tschüß.

% java I18NSample en US
Hello.
How are you?
Goodbye.
```

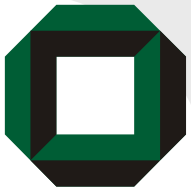




UML Anwendungsfälle

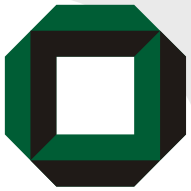
Use Cases





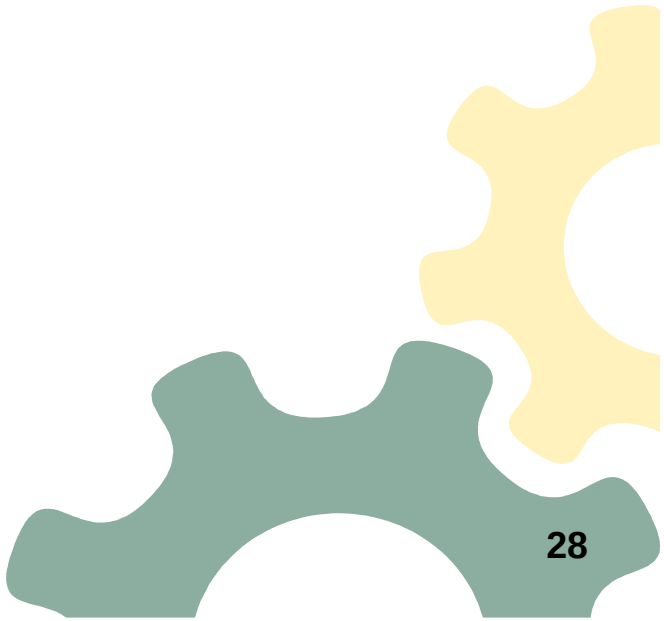
Anwendungsfalldiagramme

- Ein Anwendungsfall ist eine typische, gewollte Interaktion eines oder mehrerer Akteure mit einem (geschäftlichen oder technischen) System.
- Ein Anwendungsfall wird stets durch einen Akteur initiiert und führt i.d.R. zu einem durch einen Akteur wahrnehmbaren Ergebnis.
- Ein Anwendungsfall beschreibt was ein System leisten muss, nicht *wie* es das leisten muss – er kann insbesondere mehrere verschiedene Ablaufvarianten umfassen



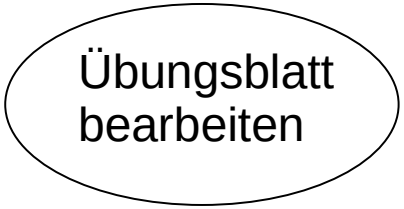
Elemente

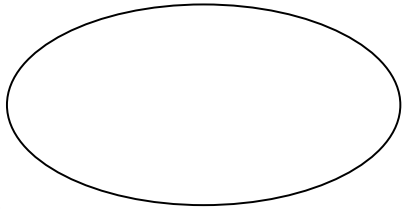
- Anwendungsfälle
- System
- Akteure
- Assoziationen
- Include Beziehung
- Extend Beziehung



Anwendungsfall

- Ein Anwedungsfall zeigt das Verhalten (das was, nicht das wie) eines Systems.

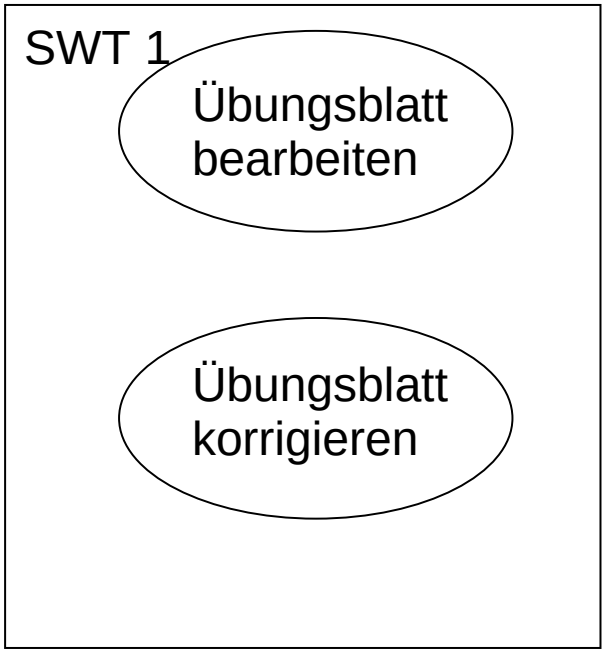




Übungsblatt bearbeiten

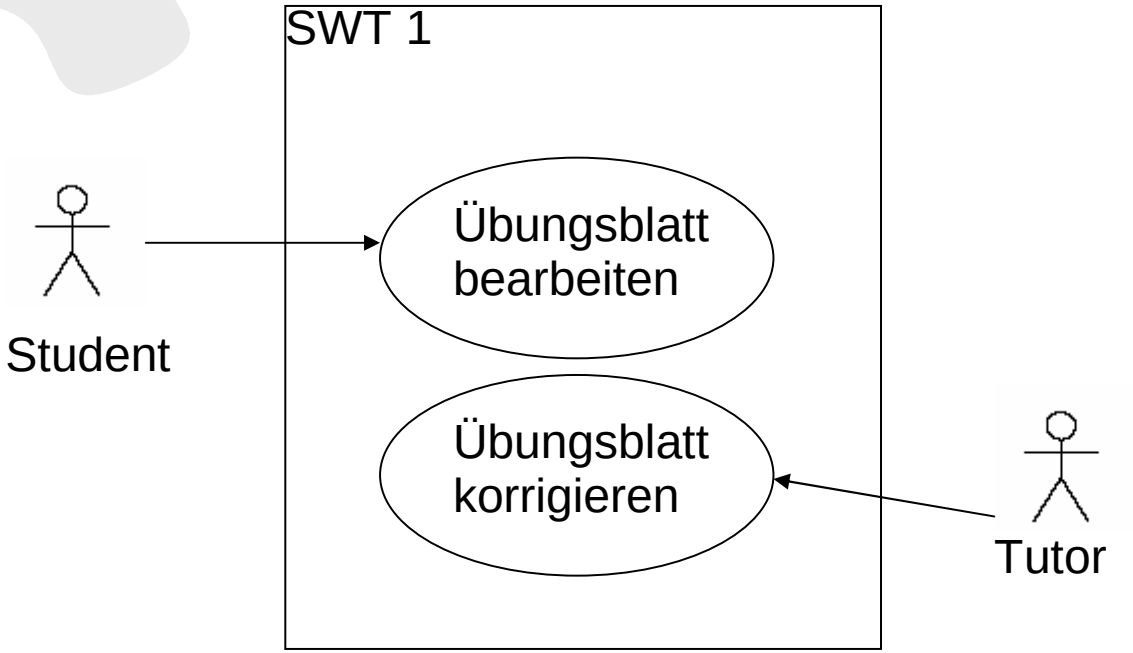
System

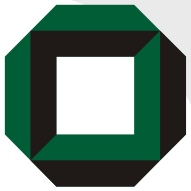
- Kann verwendet werden um die Anwendungsfälle innerhalb eines Systems zu „gruppieren“.



Akteur

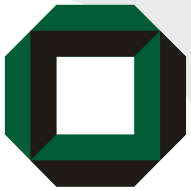
- Ein Akteur ist etwas, das mit dem modellierten System interagieren muß.





<<include>>

- Wird verwendet, wenn verschiedene Anwendungsfälle die gleiche Teilfunktionalität beinhalten.
- Der Anwendungsfall A verwendet das Verhalten eines Anwendungsfall B.



<<extend>>

- Die Funktionalität des Anwendungsfalles *B* ist eine (optionale) Erweiterung des Anwendungsfalls *A*.
- Die extend-Beziehung sollte verwendet werden, wenn das Verhalten eines Anwendungsfalles erweitert werden kann (aber nicht muß)

Extend und Include im Vergleich

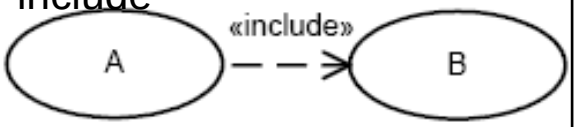
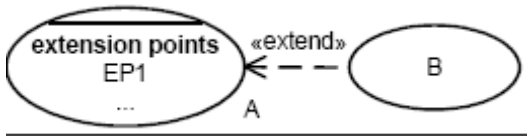
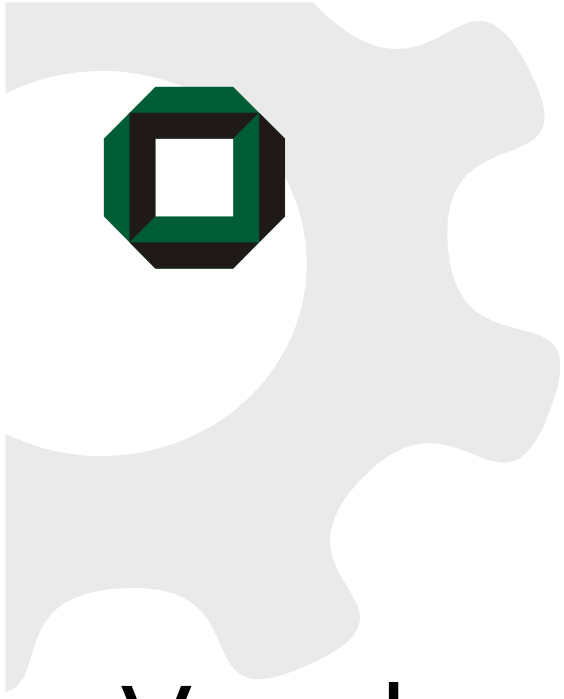
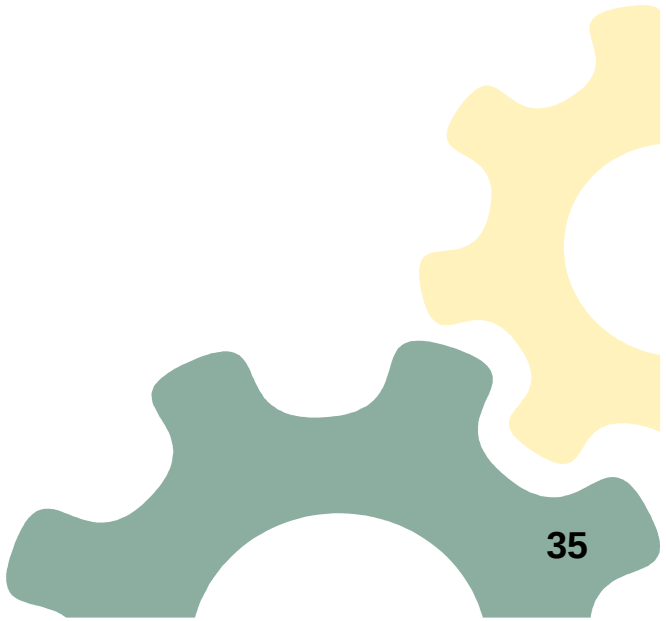
	include 	extend 
Bedeutung	Ablauf von A schließt immer Ablauf von B ein.	Ablauf von A kann, muß aber nicht durch Ablauf von B erweitert werden.
Wann?	Ablauf von B kann in verschiedenen Anwendungsfällen genutzt werden.	A besitzt neben dem Normalverhalten auslagerbare Sonderfälle.
Bedeutung für die Modellierung	A ist meist unvollständig und wird erst durch Inklusion von B zu einem vollständigen Ablauf	A ist meist vollständig und kann durch B optional erweitert werden.

Abbildung (gekürzt) nach: UML2 Glasklar, Jeckle u. a., Carl Hanser Verlag, 2003, ISBN 3-446-22575-7, S. 196



Vererbung und das Substitutionsprinzip

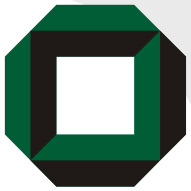


Vererbung

- Vererbung sol… …ist-ein“ Beziehu…
- Vererbun…
  - Signa… …t von ih… Method…
  - Implem… …er Vaterkla… …tation.

In Java:

	Signatur-vererbung	Implementierung-vererbung	Mehrfach-vererbung
Schnittstellen	✓		✓
Klassen		✓	

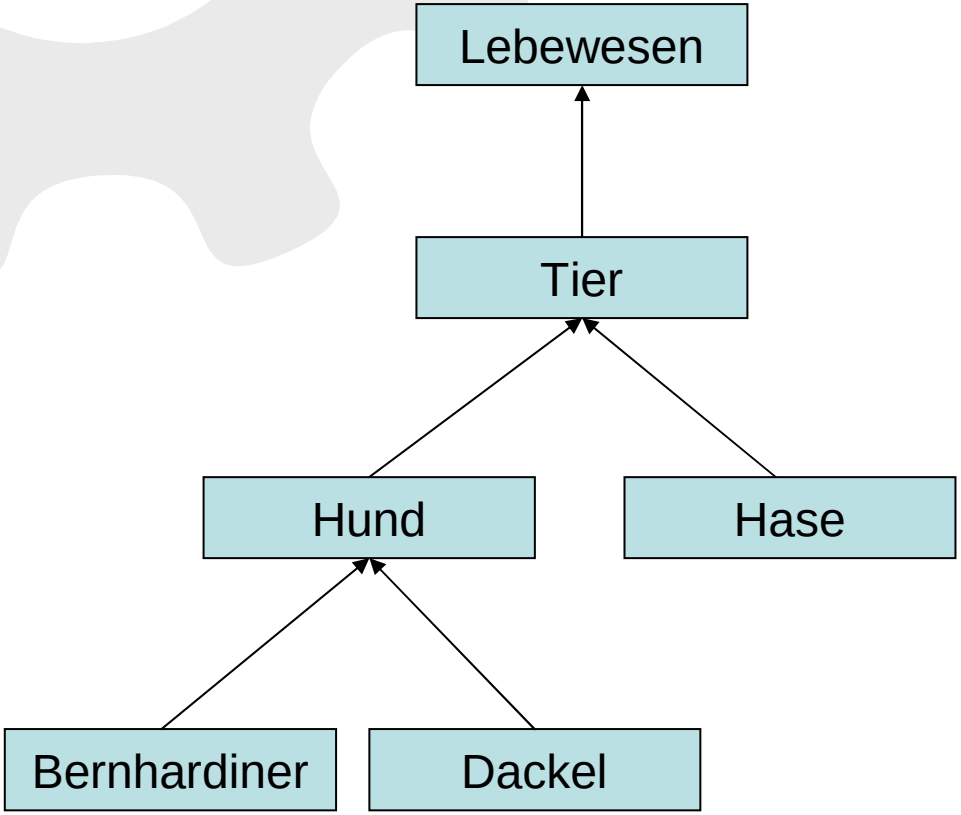


Substitutionsprinzip

- Substitutionsprinzip („ist ein“ Beziehung): Jedes Exemplar einer Unterklasse hat die gleichen Eigenschaften, die ein Exemplar der Oberklasse hätte; es lässt sich genau so verwenden.

⇨ Das Substitutionsprinzip fordert nicht, dass alle Signaturen identisch sind!

Varianzen

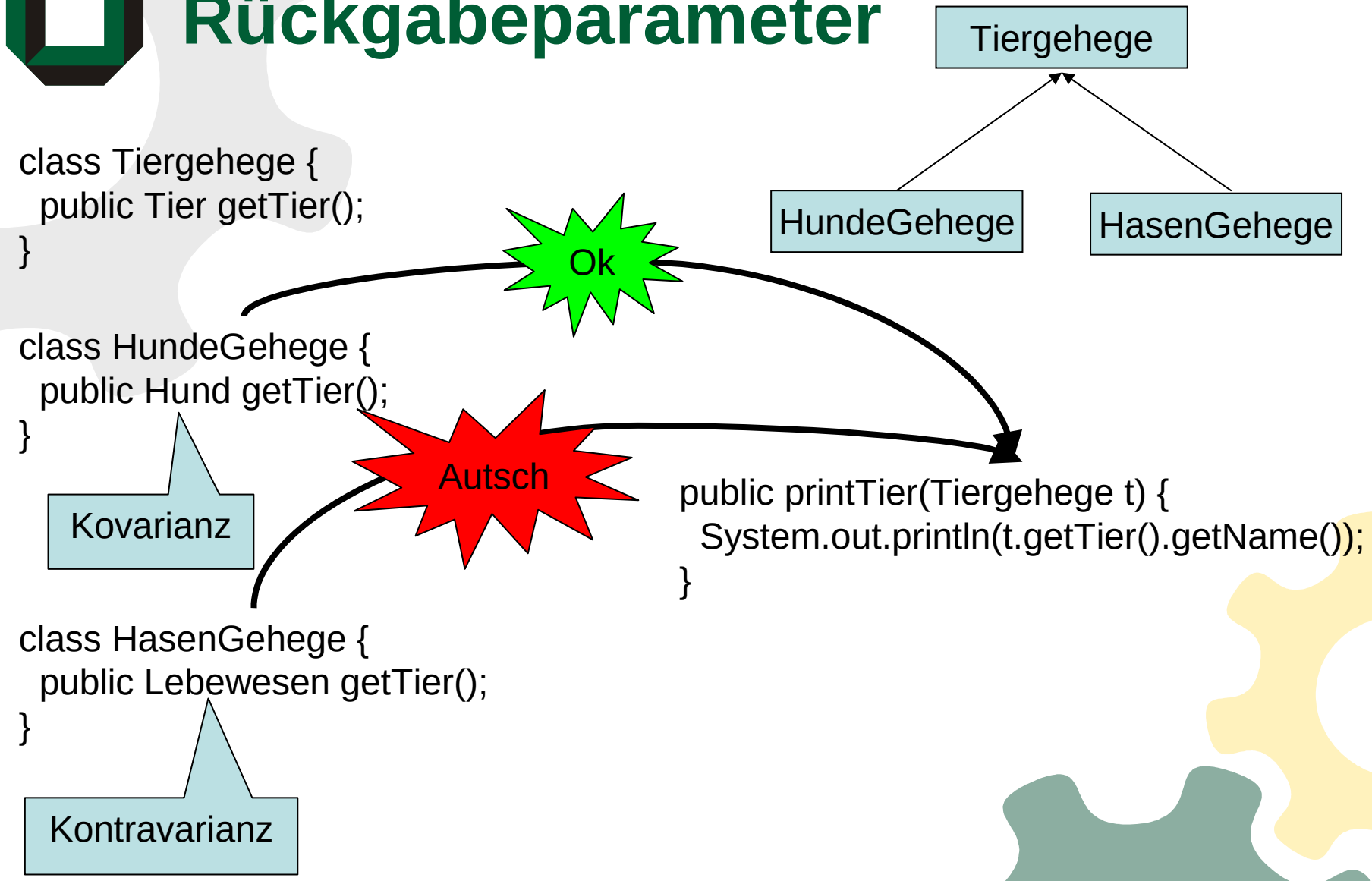


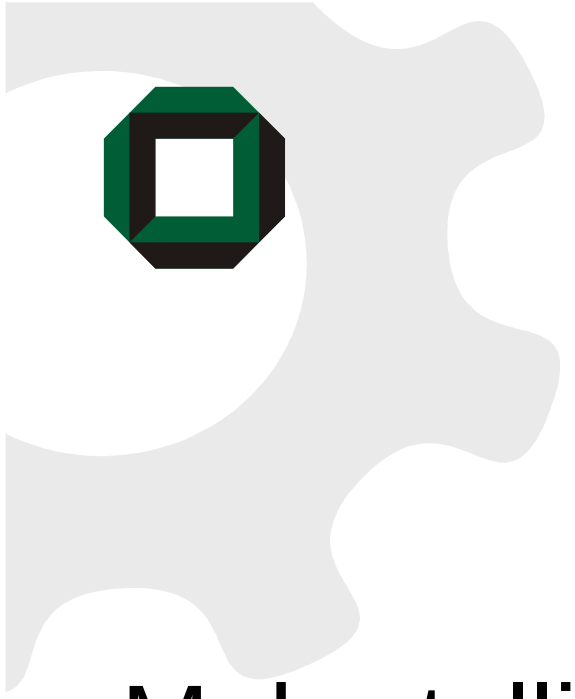
Kovarianz: Spezialisierung
„Entgegen der Pfeilrichtung“

Kontravarianz: Generalisierung
„In Pfeilrichtung“

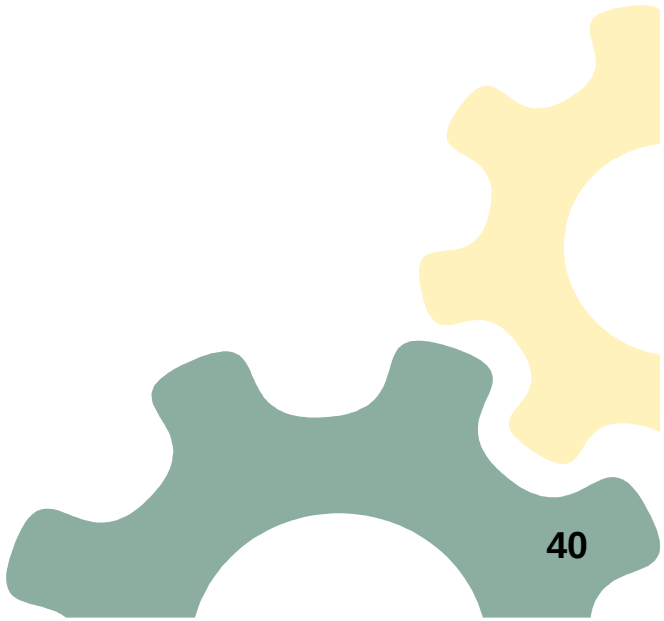
```
class Tier {
  public String getName();
}
```

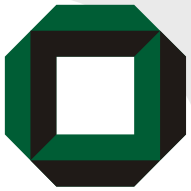
Rückgabeparameter





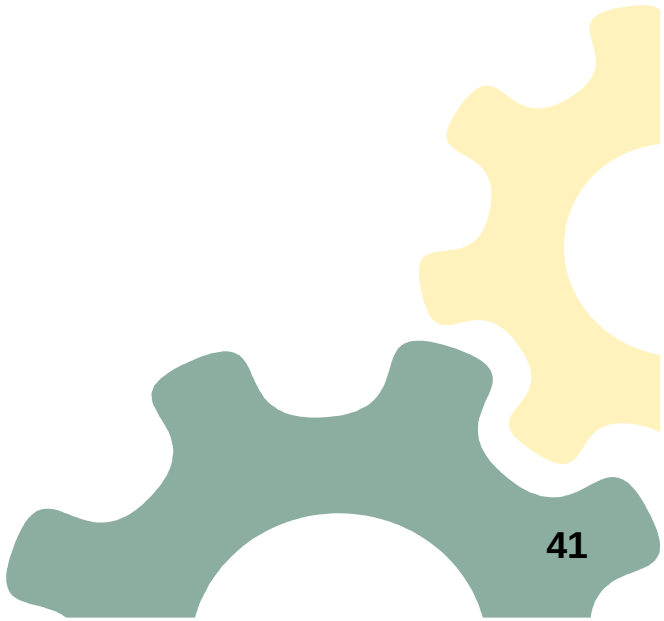
Mehrstellige Assoziationen



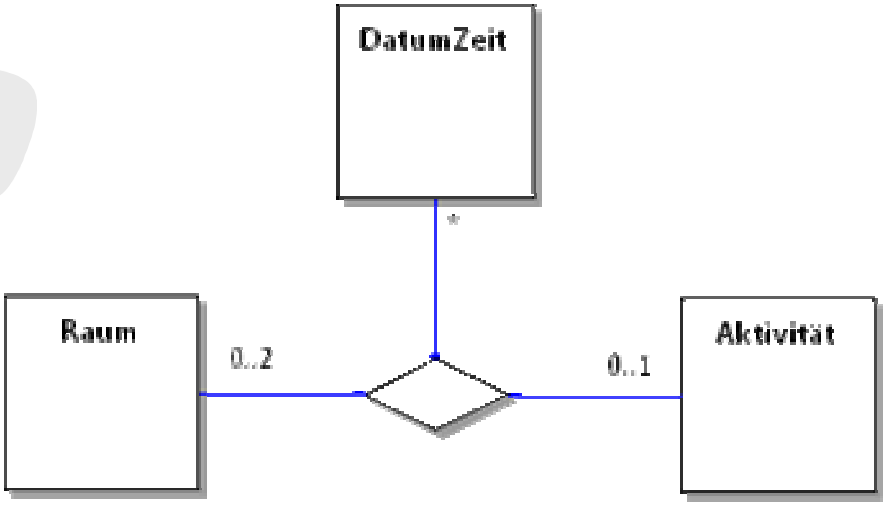


Mehrstellige Assoziationen

- Definition: Betrachte bei n-ären Assoziationen ‚mit Stelligkeit *n,* Kombination von *n*-1 Objekten und bestimme, mit wie vielen Objekten der verbleibenden Klasse sie verbunden sein kann.



Mehrstellige Assoziationen

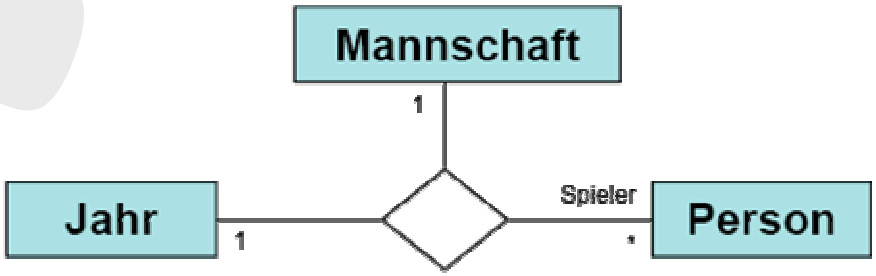


•Jeder Kombination aus (Raum, Zeit) ist eine, oder keine Aktivität zugeordnet. Ein beliebiger Raum ist an einem beliebigen Datum also entweder besetzt oder frei.

•Jeder Kombination aus (Raum, Aktivität) sind keine oder beliebig viele Zeiten zugeordnet. In einem bestimmten Raum kann eine bestimmt Aktivität mehrmals stattfinden.

•Jeder Kombination aus (Datum, Aktivität) ist keiner, einer oder zwei Räume zugeordnet. An einem bestimmten Datum kann eine bestimmte Aktivität gar nicht stattfinden, oder in bis zu zwei Räumen.

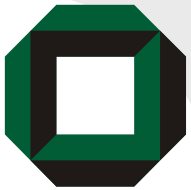
Mehrstellige Assoziationen



•Jeder Kombination aus (Jahr, Mannschaft) sind beliebig viele Personen (Rolle: Spieler) zugeordnet. In jedem Jahr spielen in jeder Mannschaft entsprechend beliebig viele Spieler.

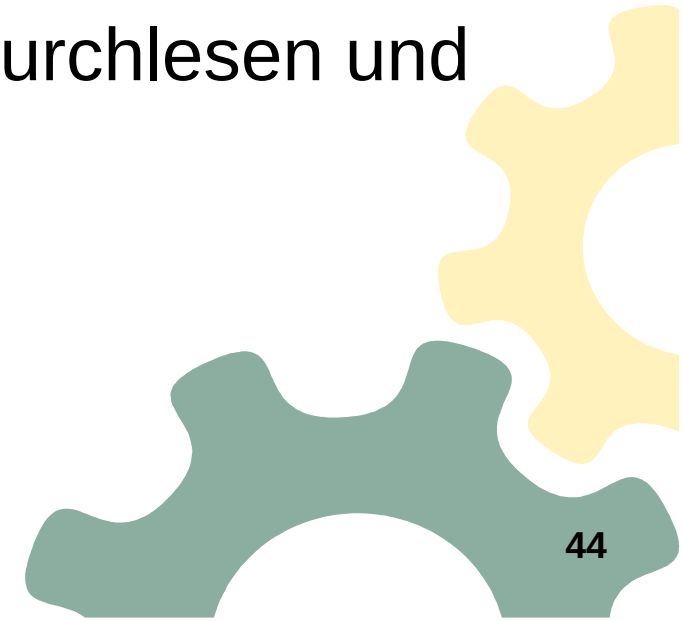
•Jeder Kombination aus (Jahr, Person) ist eine Mannschaft zugeordnet. Jede Person hat in jedem Jahr also in einer Mannschaft gespielt. Arbeitslosigkeit unmöglich.

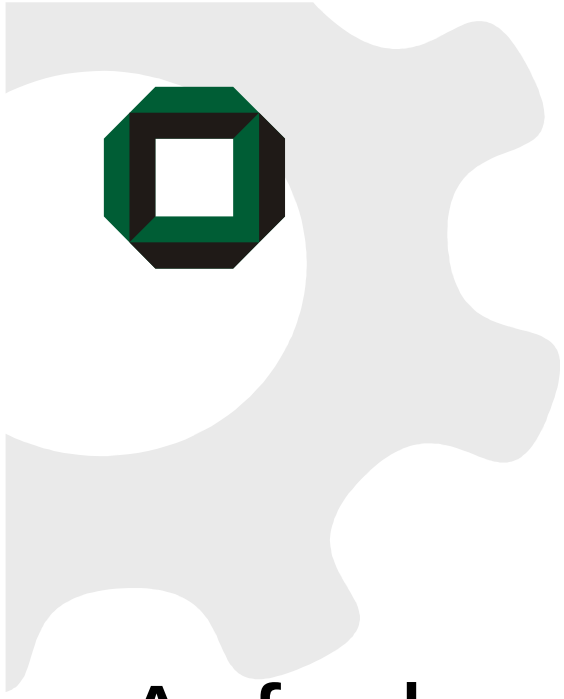
•Jeder Kombination aus (Mannschaft, Person) ist ein Jahr zugeordnet. Ein Spieler kann also nur einmal für eine Mannschaft spielen.



Tipps für das nächste Übungsblatt

- Es besteht die Möglichkeit, dass das Internationalisierung/Lokalisierungsthema von heute wichtig werden könnte…
- Schaut auch „Actions“ in Swing an. Die könnten ggf. wichtig werden.
- Die Abgabemodalitäten genau durchlesen und einhalten!





Aufgaben