

Ziele der Vorlesung „Software-Engineering I“

„Engineering-Philosophie“

- **Vermittlung des Engineering-Denkens:**

„nicht einfach loslegen, sondern strukturiert vorgehen und alle Teilaspekte berücksichtigen“

- Basis für eine erfolgreiche Softwareentwicklung
- ➔ Übungen und Aufgaben nicht ausschließlich softwarebezogen
- Förderung und Forderung des „Mitdenkens“, der Kreativität, Eigeninitiative, Verantwortlichkeit und Arbeit im Team

„Engineering-Philosophie“

- **Vermittlung der Engineering-Grundlagen**

„wo fange ich an und wo höre ich auf?“

- Software-Arten (prozedural vs. objektorientiert)
- Vorgehensmodelle, Grundprinzipien, ...
- Lastenheft, Pflichtenheft, Analyse, Entwurf, Realisierung, ...
- Modularisierung, Modulintegration, ...
- Tests (Testarten, Testplanung, Durchführung, ...)
- Management (Projekt-M., Konfigurations-M, Qualitäts-M)
- . . .

„Engineering-Philosophie“

- **Vermittlung von Software-Engineering-Methodiken**

„welche Werkzeuge/Methoden kann ich wann und wo einsetzen?“

- **Datendefinition / Datenmodellierung**
Syntax-Diagramm, *Data Dictionary*, UML (Instanzen- und Klassendiagramm)
- **Prozessdefinition / Prozessmodellierung**
Strukturierte Analyse, SADT, Programmablaufplan, Struktogramme, Pseudocode, UML (Aktivitäts-, Zustands- und Sequenzdiagramm)
- **Strukturiertes Testen**
Testplan, automatisiertes Testen (z.B. JUnit)
- **Abbildung (Mapping) der Datenmodelle auf lauffähigen Quellcode,**
- **Integration von Datenbanksystemen (ORM - objektrelationales Mapping)**
UML -> JDBC
- **Objektorientierung**
Analyse- und Design-Patterns, Komponenten

Prüfungsleistungen, Umfang, Bewertung

- **Prüfungsleistungen**

ALT:

- 2 Testate (WS, SS)
- Programmentwurf

NEU:

- Programmentwurf

- **Umfang, Bewertung**

ALT:

- Workload:
144 Std. Präsenz,
306 Std. Selbststudium
- 15 Credit Points

NEU:

- Workload:
96 Std. Präsenz,
174 Std. Selbststudium
- 9 Credit Points

Vereinbarungen

- Sie bekommen sämtliche gezeigten Präsentation als PDF-Dateien
- Laptop, Tablet & Handy werden ohne Aufforderung bzw. Genehmigung während der Vorlesung **nicht** benutzt
- Fragen können jederzeit gestellt werden
 ➔ **Die Vorlesung lebt von Ihren Fragen und Beiträgen!**
- Nichtanwesenheit (Krankheit, wichtige Termine, ...) per E-Mail vor der Vorlesung mitteilen (Richard.Lutz@kit.edu)

Einstiegsaufgabe:

**Erstellen Sie ein Programm, das die zwei
berühmtesten Worte der Programmiererwelt
ausgibt:**

„Hallo Welt“

1. Lösungsmöglichkeit(en):

```
>> echo Hallo Welt  
Hallo Welt  
>>
```

DOS und UNIX-Shell

Weiterhin: Skripte in Perl, csh, sh, awk, ...

Etwas aufwändiger: Mini-Programm in C-/C++, Pascal, Java, ...
(als Programmaufruf ausgeführt)

```
>> java HalloWelt  
Hallo Welt  
>>
```

DOS und UNIX-Shell

2. Lösungsmöglichkeit:



2. Lösungsmöglichkeit:



Folgende Fragen hätten gestellt werden können:

- Eigenes Programm (Aufruf) oder Befehl auf Kommandozeile?
- „Aussehen“ des Programms (User Interface)?
- Leistungsfähigkeit des Programms (nur Ausgabe oder „mehr“)?
- Standalone-Element oder als Komponente zu integrieren?
- Benutzerumgebung (DOS, WINDOWS, UNIX, MAC, Shell, ...)?
- Start durch Mausklick oder durch Benutzereingabe?
- Ausgabe des Textes mit oder ohne Anführungszeichen?
- Farbe des Textes?
- Und einige mehr ...