

VP Multi-paradigm Programming SS 2013

Andreas Naderlinger
andreas.naderlinger@cs.uni-salzburg.at

Inhalt (1)

- Überblick über (Haupt-)Programmierparadigmen
- Funktionale Programmierung (LV Sw-Techniken?)
 - Scheme (Lisp) / Scala
- Aspect-oriented programming → Mehrdimensionale Modularisierung
- Synchronous Block Diagrams / Dynamic systems (~Data-flow paradigm) (MATLAB/Simulink)
 - Mixing Declarative / Imperative / Dynamic
 - Real-time programming paradigms atop Synchronous Block Diagrams

Inhalt (2)

- “Principles/Concepts” of Programming Languages
 - Referate

Termine

- 05.3. Vorbesprechung
- 12.3. VP
- 19.3. VP
- 26.3. -- Ostern
- 02.4. -- Ostern
- 09.4. -- SpringSim Conference
- 16.4. VP
- 23.4. VP
- 30.4. VP, Referat:
- 07.5. VP, Referat:
- 14.5. VP, Referat:
- 21.5. -- Pfingsten
- 28.5. -- Mathworks Summit
- 04.6. -- Mathworks Summit
- 11.6. VP, Referat:
- 18.6. VP, Referat:
- 25.6. final lecture: exam

+ 2 Ersatztermine

April/Mai/Juni

**→ Im Anschluss ans
reguläre VP.
Termine: tba**

Allgemeines

- Dienstag 14:00 - 15:30
– T06
- Anwesenheitspflicht
- Homepage zum VP:

www.softwareresearch.net → Teaching

- Termine

Referate

- Ablauf:
 - 45 Minuten Vortrag
 - Fragen von Publikum/LV-Leiter an Vortragenden
 - Fragen vom Vortragenden/LV-Leiter ans Publikum
 - ☒ → (Verständnis-Check)
 - uU. Praktische Übungen
- Einzelbesprechungen nach Ostern.
- Deliverables
 - Slides
 - Handout (1-2 pages)
 - Literature survey
 - ☒ List of papers/books with very short summary, description which parts were relevant for you.

Benotung

- Referat samt Unterlagen
- Klausur
- Hausübungen
- Mitarbeit

Referatsthemen

1. How Scala code is executed on the JVM / Functional Java byte-code / Compilation
2. Funktionale/Logische Sprachen in der Praxis (z.B. Erlang,...). Lessons learned, Pros, Cons ...
3. Typesystems: Variance, Covariance, generics, Arrays (java, c#, c++), Strong?, Static?, type equivalence, type inference
4. Classical Design Patterns in duck-typing languages (python)
5. Runtime organization, exception handling (stack frames). long jumps. java, C, ...
6. Garbage Collection algorithms
7. Actor-oriented programming / Multi-paradigm modeling (model-driven engineering, simulation). (Ptolemy)
8. ... Vorschläge sind willkommen
 - Prog. Paradigms for the web
 - Prog. Paradigms for mobile devices
 - Theoretical Background of func.prog (lamda)
 - Concurrency (on different levels, in different languages,...)