

# Fortgeschrittene Programmierung

## WS 21/22

Michael Hanus

30. November 2021

### Detaillierter Vorlesungsverlauf

- 25.10.** Organisatorisches;  
Nebenläufige Programmierung in Java: grundlegende Begriffe, Synchronisationsproblem, Semaphore, Dining Philosophers
- 26.10.** Klasse `Thread`, Interface `Runnable`, Eigenschaften von Thread-Objekten, Monitor-Konzept, Synchronisation von Threads in Java, `synchronized`-Methoden, `synchronized`-Anweisung, synchronisierte Methoden vs. synchronisierte Anweisungen, synchronisierte Collections, Kommunikation zwischen Threads, `wait`, `notify`
- 1.11.** genaue Bedeutung und sinnvolle Benutzung von `wait`, `notify`, `notifyAll`, einelementiger Puffer, Benutzung von Synchronisationsobjekten, Beenden und Unterbrechen von Threads, `InterruptedException`
- 2.11.** Serialisierung von Daten, Idee von RMI, Parameterübertragung, Serverseite, Clientseite, RMI-Registrierung, Probleme bei Client-side Synchronisation mit RMI
- 8.11. Einführung in die funktionale Programmierung:** Variablenbegriff, Programm, Funktionsdefinitionen, Ausdrücke, Beispiel `square`, Beispiele `min/fac`, Auswertungsmöglichkeiten, Fibonacci-Zahlen (rekursiv und iterativ)
- 9.11.:** Lokale Definitionen, Layout-Regel, Vorteile lokaler Definitionen; Basisdatentypen, Typnotationen, algebraische Datentypen (Aufzählungstypen, Verbundtypen, gemischte Typen, Listen), Ausgabe von Daten (`deriving Show`), Operatoren
- 15.11.** polymorphe Funktionen (`length`, `++`), `last`), Definition polymorpher algebraischer Datentypen, `Maybe` und `maybeLast`, Binärbäume, `String`, Vereinigungstypen (`Either`), Tupel (`fst`, `snd`)
- 16.11.** Funktionen `zip/unzip`, Pattern Matching (Patternaufbau, case-Ausdrücke), Guards, Funktionen höherer Ordnung, anonyme Funktionen, partielle Applikation, Currying, Sections, Funktion `flip`
- 22.11.** generische Programmierung (`map`, `foldr`, `filter`, `foldl`), Kontrollstrukturen als Funktionen höherer Ordnung (`while`), Funktionen als Datenstrukturen (Implementierung von Feldern)

- 23.11.** wichtige Funktionen höherer Ordnung (Komposition, `curry/uncurry`, `const`), Funktionen höherer Ordnung in imperativen Sprachen (Ruby, Java 8, JavaScript), Motivation und Struktur von Typklassen, Instanzen, vordefinierte Funktionen in Typklassen
- 29.11.** Typklassen für polymorphe Datentypen, Standardklassen, `deriving`, Klasse `Read` und Funktionen `read` und `reads`; Unterschiede bei Auswertungsstrategien, Programmsignatur, Terme
- 30.11.** Programm, Termersetzungssystem, Substitution, Position, Teilterm, Reduktionsschritt, Normalform, Wertaufruf, Namensaufruf, Reduktionsstrategien (LI, RI, LO, RO, PI, PO), Berechnungsstärke von `outermost` und `parallel outermost`, Rechnen mit unendlichen Datenstrukturen (`from`, `primes`)