

# Funktionale Programmierung

LVA 185.A03, VU 2.0, ETCS 3.0

WS 2011/2012

Jens Knoop



Technische Universität Wien  
Institut für Computersprachen  
knoop@complang.tuwien.ac.at  
[www.complang.tuwien.ac.at/knoop](http://www.complang.tuwien.ac.at/knoop)



# Inhalt der Lehrveranstaltung

- ▶ Funktionaler Programmierstil (Programme als Systeme rekursiver Rechenvorschriften)
- ▶ Lambda-Kalkül (Grundlage der semantischen Fundierung funktionaler Programmiersprachen)
- ▶ Auswertungsstrategien für Ausdrücke und Programme (eager evaluation, lazy evaluation)
- ▶ Rekursion, Rekursionstypen
- ▶ Funktionen höherer Ordnung (Programmieren mit Funktionen als Argument und Resultat)
- ▶ Polymorphie (parametrisch, ad-hoc)
- ▶ Typsysteme, Typinferenz (monomorph, polymorph)
- ▶ Ströme und Stromverarbeitung
- ▶ ...

...sowie die Umsetzung&Anwendung dieser Konzepte in Haskell

# Ziele der Lehrveranstaltung

*...auf der Meta-Ebene.*

- ▶ Einsicht in die Konzepte funktionaler Sprachen
- ▶ ihre Umsetzung in einer *state-of-the-art* Sprache, hier Haskell
- ▶ sinnvoller Einsatz dieser Konzepte (Tipps, Tricks und mehr!)

Protagonisten meinen

**Functional Programming is Fun!**

*...ist das wirklich wahr?*

# Wenn ja, warum?

Eine Antwort: *...weil funktionale Programmierung etwas von der Eleganz der Mathematik in die Programmierung bringt!*

(P. Pepper, [6])

Ein “Klassiker” in diesem Zusammenhang:

- ▶ Hughes, J. *Why Functional Programming Matters*.  
Computer Journal 32(2), 98-107, 1989

# Ein vieldiskutiertes Thema

## Praxisrelevanz

- ▶ Wadler, P. *An angry half-dozen*. ACM SIGPLAN Notices 33(2), 25-30, 1998
- ▶ Wadler, P. *Why no one uses functional languages*. ACM SIGPLAN Notices 33(8), 23-27, 1998

## Positive Beispiele aus Wirtschaft und Hochschule:

- ▶ Aus der Wirtschaft im großen Maßstab:
  - ▶ Telekommunikationsanwendungen bei Ericsson unter Verwendung der funktionalen Sprache *Erlang*
- ▶ Aus Hochschule (Forschung&Lehre) und Wirtschaft:
  - ▶ LVA 185.187 Optimierende Übersetzer VU 3.0 ECTS 4.5 WS 2010/11: *Program Analysis Generator (PAG)* (Univ. Saarbrücken & AbsInt GmbH)

# Funktionale Programmierung

...rundet die einführenden Lehrveranstaltungen zu wichtigen Programmierstilen ab.

- ▶ **Objektorientierte Programmierung**

*LVA 185.A01 Objektorientierte Programmieretechniken VU  
2.0 ECTS 3.0 WS 2011/12*

- ▶ **Logikorientierte Programmierung**

*LVA 185.A12 Logikprogrammierung und Constraints VU  
4.0 ECTS 6.0 WS 2011/12*

- ▶ **Funktionale Programmierung**

*LVA 185.A03 Funktionale Programmierung VU 2.0 ECTS  
3.0 WS 2011/12*

...die in entsprechenden fortgeschrittenen Lehrveranstaltungen fortgeführt und vertieft werden.

# Was Sie mitbringen sollten...

## Voraussetzungen

- ▶ grundlegende Programmierkenntnisse  
(Einführung in das Programmieren, Grundlagen der Programmierung, Programmkonstruktion)

*(Darüberhinaus mögen die in Abschnitt 1.6 des ab 01.10.2006 gültigen neuen Studienplans niedergelegten Voraussetzungen zur Anwendung kommen (ZbGr).)*

## Anrechenbarkeit

- ▶ Siehe TISS

# Aufbau der Lehrveranstaltung

- ▶ Vorlesung
- ▶ Übung in Gruppen (2er-Gruppen)
- ▶ Plenumsübung *Haskell Live*
- ▶ Schriftliche Prüfung (sog. Klausur) (90 Minuten) über Vorlesungs- und Übungsstoff und einen wissenschaftlichen Artikel, den Sie sich selbstständig im Lauf der Vorlesungszeit erschließen, und zwar:

**John Hughes.** *Why Functional Programming Matters.* Research Topics in Functional Programming. D. Turner (Hrsg.), Addison Wesley, 1990.

Anmeldung zur Klausur **zwingend erforderlich!** Und zwar über das elektronische Anmeldesystem. Informationen dazu werden rechtzeitig in der Vorlesung und über TISS bekanntgegeben.

## Unterlagen (On-line)

- ▶ Informationen zu Haskell: <http://www.haskell.org/>
- ▶ Haskell-Tutorial:  
<http://www.haskell.org/tutorial/>
- ▶ Hugs-Interpreter: <http://www.haskell.org/hugs>
- ▶ ...

## Unterlagen (Off-line)

1. Folien und Begleitmaterial zur Vorlesung *Funktionale Programmierung* aus den vergangenen Wintersemestern. Institut für Computersprachen, TU Wien.
2. Simon Thompson. *Haskell: The Craft of Functional Programming*, Addison-Wesley (Pearson), 2nd edition, 1999.
3. Graham Hutton. *Programming in Haskell*. Cambridge University Press, 2007.
4. Richard Bird. *Introduction to Functional Programming using Haskell*, Prentice-Hall, 2nd edition, 1998.
5. Manuel Chakravarty, Gabriele Keller. *Einführung in die Programmierung mit Haskell*, Pearson Studium, 2004.
6. Peter Pepper. *Funktionale Programmierung in OPAL, ML, Haskell und Gofer*, Springer-Verlag, 2. Auflage, 2003.
7. ...

# Forschungsliteratur

Quellen aktueller Forschungsarbeiten und -resultate im Umfeld funktionaler Programmierung sind neben anderen insbesondere folgende Zeitschrift und Konferenzserie:

- ▶ *Zeitschrift*

- ▶ The Journal of Functional Programming, Paul Hudak, Greg Morrisett (Hrsg.), Cambridge, UK.  
<http://www.cambridge.org/uk/journals/JFP/>

- ▶ *Konferenzserie*

- ▶ ACM SIGPLAN International Conference Series on Functional Programming (ICFP)  
<http://www.acm.org/sigs/sigplan/icfp.htm>

Siehe auch

- ▶ <http://www.cs.luc.edu/icfp>

für weitere Hinweise auf einschlägige und wichtige Zeitschriften und Konferenzen im Umfeld funkt. und log. Programmierung.

# Anmeldung

## Zweistufig

- ▶ Stufe 1): *TISS-Anmeldung*  
...jeder für sich über TISS
- ▶ Stufe 2a): *Benutzerkonto-Anmeldung*  
...jeder für sich über das elektronische  
Lehrstuhl-Anmeldesystem
- ▶ Stufe 2b): *Benutzerkonten-Gruppenbildung*  
...ein Gruppenrepräsentant für die Gruppe ebenfalls über  
das Lehrstuhl-Anmeldesystem

# Wichtig

Die Teilnahme an der Lehrveranstaltung ist nur möglich, wenn

- ▶ sowohl die TISS-Anmeldung als auch die Benutzerkonten-Anmeldung und -Gruppenbildung **erfolgreich** abgeschlossen sind.

Stichtag dafür:

- ▶ der Abgabetermin für die Abgabe des ersten Aufgabenblatts.

# Tutoren, Studienassistent, Mitveranstalter

In diesem Semester

- ▶ Als Tutoren

- [1 ] Florian Hassanen

- [2 ] Bong Min Kim

- [3 ] Christoph Spörk

- [4 ] Bernhard Urban

- ▶ Als Studienassistent

- [5 ] David Flamme

- ▶ Als Mitveranstalter

- [6 ] Ass.Prof. Dipl.-Ing. Dr. techn. Ulrich Neumerkel

- [7 ] Dipl.-Ing. Jakob Zwirchmayr

# Vorlesung und Übungsaufgaben

- ▶ Dienstags von 08:15 Uhr - ca. 09:45 Uhr im Informatik-HS (Ziel: Vorlesungsteil bis Ende Dezember abschließen!)
- ▶ Beginnend mit dem 11.10.2011 im Regelfall jeden Dienstag ein neues Aufgabenblatt (abrufbar auf der Webseite der LVA]
- ▶ Abgabe: Lösungen nach einer Woche (Dienstag) um 15 Uhr automatisch aus Home-Verzeichnis (top-level! Nicht in Unterverzeichnissen!) abgesammelt
- ▶ Nachträgliche Abgabe: Änderungen nach einer weiteren Woche automatisch abgesammelt
- ▶ Insgesamt ca. 10 Abgaben
- ▶ Gesamtpunktzahl pro Abgabe gemäß der Formel:  
(Punkte Erstabgabe + Punkte Zweitabgabe) / 2

# Benützung von Rechnern

- ▶ Server für Laborübung: `g0.complang.tuwien.ac.at`
- ▶ Terminals im Labor (Argentinerstraße 8, Erdgeschoss im Innenhof)
- ▶ Arbeiten auf anderen Rechnern möglich (z.B. zu Hause)
- ▶ Nötige Software: Hugs (frei verfügbar)
- ▶ Abgaben ausschließlich auf `g0.complang.tuwien.ac.at`

# Beurteilung

- ▶ Je zur Hälfte Übungsbeurteilung und Ergebnis der schriftlichen Prüfung (Klausurergebnis)  
**Hauptklausurtermin:** *Do, 19.01.2012* (Anmeldung erforderlich!); danach 3 *Nachtragsklausurtermine* zu Beginn (02.03.2012), in der Mitte (27.04.2012) und zu Ende der Vorlesungszeit im SS 2012 (22.06.2012) (Anmeldung jeweils erforderlich!). Nach Ablauf der Vorlesungszeit im SS 2012 *keine* weiteren Nachtragstermine. Ausstellung dann noch offener Zeugnisse im Juli/August 2012).  
*Merken Sie sich diese Termine bitte vor und planen Sie entsprechend!*
- ▶ Positive Note nur, wenn beide Teile positiv
- ▶ Schriftliche Prüfung (Klausur) (keine Hilfsmittel, Anmeldung über TISS erforderlich!)
- ▶ Punkte für Lösungen: max. 100/Abgabe, ca. 10 Abgaben
- ▶ Mindestens 50% der Punkte für positive Übungsbeurteilung
- ▶ Halbe Punkteanzahl für nachträgliche Abgaben
- ▶ Nachträgliche Abgaben können die Punkteanzahl positiv und negativ (bei Verschlechterung der Lösung) beeinflussen
- ▶ **Achtung:** Auch wenn Sie schon beim ersten Mal 100 Punkte hatten, müssen Sie für die Nachabgabe eine Lösung zum Absammeln vorhalten (z.B. die Lösung für die Erstabgabe!)

# Bei Fragen und Problemen

Insbesondere:

- ▶ Webseiten:
  - ▶ LV-Stammseite:  
`www.complang.tuwien.ac.at/knoop/fp185161.html`
- ▶ Plenumsübung *Haskell Live*
- ▶ Tutoren im Labor (zu bestimmten Zeiten; Informationen dazu in Kürze auf der Webseite zur Vorlesung)

# Nächste Vorlesungs- & *Haskell Live*-Termine

- ▶ Di, 11.10.2011, Vorlesung von 08:15 Uhr bis 09:45 Uhr im Informatik-HS
- ▶ Fr, 14.10.2011, *Haskell Live* von 14:15 Uhr bis 15:00 Uhr im Informatik-HS

Weitere Termine: siehe Webseite der Lehrveranstaltung

Erstes Übungsblatt: Di, 11.10.2011!

Erste Abgabe: Di, 18.10.2011.

# Die FP-Teammitglieder wünschen Ihnen

...viel Erfolg bei dieser Lehrveranstaltung und dass Sie auch langfristig von ihr profitieren!

*Zu guter letzt:*

Die Vorlesung lebt mit Ihnen! Ihre Rückmeldungen, Anregungen, Verbesserungsvorschläge sind willkommen!  
Natürlich auch, wenn Ihnen etwas gut gefallen hat!