

## Organisatorisches und Überblick

Prof. Dr. David Sabel

LFE Theoretische Informatik



Beteiligte:

- **Prof. Dr. David Sabel**
- **Sarah Vaupel**

Email-Verteiler an alle Organisatoren:

`ffp2020@tcs.ifi.lmu.de`

## Vorlesung

- Mittwoch 10-12, ~~B 106 (Hauptgebäude)~~
- Freitag, 10-12, ~~B 106 (Hauptgebäude)~~ ca. alle 2 Wochen

Vorläufiger Plan für die Freitage:

24.04.    08.05.    29.05.    05.06.    19.06.    10.07.    24.07.

## Übung: 3 Termine

- Dienstags 16-18, ~~B 011 (Hauptgebäude)~~ ab 28.04.2020
- Donnerstags 14-16, ~~B 011 (Hauptgebäude)~~ ab 30.04.2020
- Donnerstags 16-18, ~~003 (Schellingstr. 5)~~ ab 30.04.2020

Bis auf Weiteres finden die Veranstaltungen **digital** und **nicht in Präsenz** statt:

**Genauerer dazu auf den nächsten Folien!**

Anstelle der üblichen Frontal-Vorlesung stellen wir folgendes bereit (nach und nach, Termine wie angekündigt):

- Skript zur Vorlesung
- Vorlesungsfolien
- ScreenCasts (Aufgezeichnete Besprechung der Vorlesungsfolien, „Live“-Coding etc.)
- Forum auf Moodle, um Fragen und Probleme gemeinsam zu besprechen.

- Aufgaben zum Selbstlösen und Programmieren
- Freiwillige Einreichung der eigenen Lösungen **und Fragen zum Stoff** über Uni2work
- Gemeinsames Besprechen und Vorführen als Online-Meetings via Zoom (oder anderem Werkzeug, wenn nicht praktikabel) und/oder Moodle-Live Chat
- Forum auf Moodle, um Fragen und Probleme gemeinsam zu besprechen.

# Das Wichtigste: Feedback

---

Bitte planen Sie ein, dass **Vieles schief gehen** wird!

Wenn etwas nicht funktioniert:

- Gelassen bleiben, aber Bescheid geben!!!
- Bitte kommunizieren Sie mit uns, per Email, per Chat, per Video-Meeting, ...
- **Wir finden eine Lösung!**

Traditionelle Prüfung in FFP:

- **Eigenes Softwareprojekt** in Haskell, welches das Verständnis von mindestens 3 Themen der Vorlesung demonstriert
- Teambildung möglich und erwünscht: (maximale Teamgröße: 3)
- Abnahme: Projektdemonstration und Befragung zum Code.  
Je nach Lage:  
Präsenzabnahme oder (Online-Meeting & schriftlicher Bericht)
- Anmeldung der Projekte: Details folgen in wenigen Wochen.
- Termine für die Abnahmen: Im September

- Durchklicken auf **[www.tcs.ifi.lmu.de](http://www.tcs.ifi.lmu.de)**
- Anmelden im uni2work und für den Moodle-Kurs
- Unser Vorgehen:

**Was über uni2work geht, machen wir über uni2work.**

Moodle daher eher nur für Kommunikation (Chat, Forum)

- Aktuelle und organisatorische Informationen beachten



- Ungefähr: Jede Woche ein Aufgabenblatt  
(Feiertagsregelungen werden bekannt gegeben)
- Abgabemöglichkeiten und -fristen auf dem Blatt beachten.
- In der Regel: Aus- und Abgabe Freitags.
- Freiwillige Abgabe und Korrektur über uni2work
- Erstes Blatt wird am 24.04. ausgegeben und kann bis 01.05. abgegeben werden
- Übungen starten nächste Woche mit:  
Hinweisen zur Installation und Verwendung der Werkzeuge und  
**der Digitalisierung (Zoom etc.)**

Ankündigungen im Uni2work und im Moodle-Forum für Ankündigungen beachten.  
(Z.B. Zoom-Raum usw.)

- **Motivation und Eigenschaften funktionaler Programmiersprachen**
- **Überblick über Haskell:** Funktionen, Datentypen, Listen, Module (zum Großteil aus ProMo bekannt)
- **Semantik funktionaler Programmiersprachen:** Funktionale Kernsprachen und deren Operationale Semantik
- **Typklassen:** Definition, Wichtige Klassen, Auflösung der Überladung in Haskell
- **Applikative Funktoren und Monaden:** Monadisches Programmieren, IO-Monade, Dateizugriff
- **Typisierungsverfahren bzw. -algorithmen:**  
Iterative polymorphe Typisierung, Milners Typisierungsverfahren

„Spezialthemen“:

- **Metaprogrammierung mit Template Haskell**
- **Funktionale Referenzen (Linsen, Prismen)**
- **Parallel und Concurrent Haskell**
- **Spracherweiterungen** Multiparameterklassen, Typfamilien, GADTs, View Patterns, Overloaded Strings, ...
- **Webframework für Haskell: Yesod**
- **GUI-Programmierung** in Haskell

**Lipovaca, M.**

*Learn You a Haskell for Great Good! A Beginner's Guide*

No Starch Press, 2011.

**Bird, R.**

*Thinking Functionally with Haskell,*

Cambridge University Press, 2014

**Thompson, S.**

*Haskell: The Craft of Functional Programming.*

Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

**Hutton, G.**

*Programming in Haskell.*

Cambridge University Press, 2007.

**O'Sullivan, B., Goerzen, J., & Stewart, D.**

*Real World Haskell.*

O'Reilly Media, Inc, 2008.

**Marlow, S.**

*Parallel and Concurrent Programming in Haskell*

O'Reilly, ISBN 9781449335939, 2013.

**Snoyman, M.**

*Developing Web Applications with Haskell and Yesod,*

O'Reilly, 2012. ISBN 1449316972.

**Barendregt, H. P.**

*The Lambda Calculus. Its Syntax and Semantics.*

North-Holland, Amsterdam, New York, 1984.

**Hankin, C.**

*An introduction to lambda calculi for computer scientists.*

Number 2 in Texts in Computing. King's College Publications, London, UK, 2004.

**Haskell Wiki:** <http://www.haskell.org/haskellwiki>.

**Hoogle:** <http://haskell.org/hoogle>  
durchsucht die Paket-Dokumentationen

**Haskell-Mailing-Listen:**

- [haskell-cafe@haskell.org](mailto:haskell-cafe@haskell.org)
- [beginners@haskell.org](mailto:beginners@haskell.org)
- [haskell@haskell.org](mailto:haskell@haskell.org)

**Stack Overflow:** <http://stackoverflow.com/questions/tagged/haskell>