

# Funktionale Programmierung

LVA 185.A03, VU 2.0, ECTS 3.0  
WS 2020/2021

– Vorbesprechung –

(Stand: 06.10.2020)

Jens Knoop



Technische Universität Wien  
Information Systems Engineering  
Compilers and Languages



Abhalten

Überblick

A Motivation

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressourcen

# Abhaltemodus: Hybrid

Aufgrund geltender **COVID-19-Beschränkungen** stehen Hörsäle, Übungsgruppenräume und Rechnerlabore im WS 2020/21 nur mit stark eingeschränkter Sitzplatzkapazität und Belegbarkeitshäufigkeit zur Verfügung.

Die verbleibenden Raumressourcen sind vorzugsweise Lehrveranstaltungen des 1. Studienjahrs zur Verfügung gestellt.

Für die Lehrveranstaltung **Funktionale Programmierung** ist deshalb ein **hybrider Abhaltemodus** nötig!

**Ziel:** Die aus der **Unmittelbarkeit** von **Präsenzveranstaltungen** resultierenden **Vorteile** möglichst umfassend zu erhalten, in **Online-Veranstaltungen** durch ein Echtzeitvideokonferenzformat. Deshalb:

- ▶ **Präsenzveranstaltungen**, wo möglich (**Kleingruppen- und Plenumsübungen**).
- ▶ **Online-Veranstaltungen** als Echtzeitvideokonferenzen, wo Präsenzveranstaltungen nicht möglich (**Vortrags- und umgekehrte Klassenzimmereinheiten**).

# Funktionale Programmierung im WS 2020/21

A Motivation

B L<sup>3</sup>: Lernergebnisse, Lernaktivitäten, Leistungsnachweise

C Organisation, Ablauf

D Ressourcen

Abhalten

Überblick

A Motivation

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressourcen

# A

## Motivation

Abhalten

Überblick

**A Motivation**

Einordnung

Generalthema

Leitfrage

Gliederung

Einstieg

Relevanz,  
Anwendungen

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressourcen

# Funktionale Programmierung

...komplementiert und rundet die grundlegenden Lehrveranstaltungen zu wichtigen klassischen **Programmierparadigmen** und **-stilen** ab:

- ▶ **Funktionale Programmierung**

*LVA 185.A03 Funktionale Programmierung*  
*VU 2.0 ECTS 3.0*

- ▶ **Objektorientierte Programmierung**

*LVA 185.A01 Objektorientierte Programmieretechniken*  
*VU 2.0 ECTS 3.0*

- ▶ **Logikorientierte Programmierung**

*LVA 185.A12 Logikprogrammierung und Constraints*  
*VU 4.0 ECTS 6.0*

...die alle auch in zugehörigen **fortgeschrittenen Lehrveranstaltungen** fortgeführt und vertieft werden.

# Generalthema der Lehrveranstaltung

...der **funktionale** Programmierstil, gekennzeichnet durch:

- Programmieren mit Funktionen als Argument und Resultat von Funktionen (sog. **Funktionen höherer Ordnung**).
- **Polymorphie** auf Datentypen und Funktionen.
- Programme als Systeme (wechselweise) rekursiver Rechenvorschriften.
- Lambda-Kalkül als rigorose semantische Fundierung funktionaler Programmiersprachen und Programmierung.
- Auswertungsordnungen für Ausdrücke und Programme, insbesondere frühe und späte Auswertung (engl. *eager*, *lazy evaluation*).
- ...
- Umsetzung, Anwendung der Konzepte in Haskell.

Can programming be liberated  
from the von Neumann style?

John W. Backus (1924-2007)  
*Turing Award* Preisträger 1977



John W. Backus. *Can Programming be Liberated from the von Neumann Style? A Functional Style and its Algebra of Programs*. Communications of the ACM 21(8):613-641, 1978.  
(Turing Award Speech)

# Lehrveranstaltungsgliederung (1)

- ▶ **Teil I: Einführung**
  - Warum fkt. Programmierung? Warum mit Haskell?  
Beispiele und Werkzeuge.
- ▶ **Teil II: Grundlagen**
  - Elementare Typen, Funktionen, Datentypdeklarationen.
- ▶ **Teil III: Applikative Programmierung**
  - Funktionen über elementaren Werten und Resultaten.
- ▶ **Teil IV: Funktionale Programmierung**
  - Funktionen über Funktionen, mit Funktionen als Argument und Resultat; Polymorphie auf Funktionen und Datentypen.



# Lehrveranstaltungsgliederung (2)

- ▶ **Teil V: Fundierung funktionaler Programmierung**
  - $\lambda$ -Kalkül, Auswertungsordnungen, Typprüfung, Typinferenz.
- ▶ **Teil VI: Weiterführende Konzepte**
  - Ein-/Ausgabe, Fehlerbehandlung, Modulkonzept, Reflexives Programmieren, Programmieren mit Strömen und Funktionen höherer Ordnung.
- ▶ **Teil VII: Abschluss**
  - Rückblick, Ausblick.
- ▶ **Literaturverzeichnis**
- ▶ **Anhänge**
  - Imperative vs. fkt. Programmierung: Schlaglichter
  - Formale Rechenmodelle
  - Andere funktionale Sprachen

# Zum Einstieg

...drei kurze und süffig zu lesende Artikel aus den Jahren:

**1998:** Philip Wadler. [Why no one uses Functional Languages](#). ACM SIGPLAN Notices 33(8):23-27, 1998.

...there is a tension between building useful systems and extending the frontiers of research *⟨note: overall, the article is a plea for functional programming⟩*.

**2011:** Yaron Minsky. [OCaml for the Masses](#). Communications of the ACM 54(11):53-58, 2011.

...why the next language you learn should be functional!

**2018:** Neil Savage. [Using Functions for Easier Programming](#). Communications of the ACM 61(5):29-30, 2018.

...when the limestone of imperative programming has worn away, the granite of functional programming will be revealed underneath *⟨quote of Simon Peyton Jones⟩*.

# Warum die nächste Sprache

...funktional sein sollte:

- ▶ Konrad Hinsen. [The Promises of Functional Programming](#). Computing in Science and Engineering 11(4): 86-90, 2009.

...adopting a **functional programming** style could make your programs more robust, more compact, and more easily parallelizable.

- ▶ Konstantin Läufer, George K. Thiruvathukal. [The Promises of Typed, Pure, and Lazy Functional Programming: Part II](#). Computing in Science and Engineering 11(5): 68-75, 2009.

...this second installment picks up where Konrad Hinsen's article "The Promises of Functional Programming" [...] left off, covering **static type inference** and **lazy evaluation** in **functional programming languages**.

# Functional Programming is Fun!

...weil funktionale Programmierung  
etwas von der Eleganz der Mathematik  
in die Programmierung bringt!

Peter Pepper. Funktionale Programmierung  
in OPAL, ML, Haskell und Gofer.  
Springer-V., 2. Auflage, 2003.

Abhalten

Überblick

A Moti-  
vation

Einordnung

Generalthema

Leitfrage

Gliederung

**Einstieg**

Relevanz,  
Anwendun-  
gen

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressource

# Wo fkt. Progr. heute angewendet wird (1)

...von **Wissenschaft**:

- ▶ Jerzy Karczmarczuk. **Scientific Computation and Functional Programming**. Computing in Science and Engineering 1(3):64-72, 1999.

...modern functional programming languages and lazy functional techniques are useful for describing and implementing abstract mathematical objects in quantum mechanics.

- ▶ Noah M. Daniels, Andrew Gallant, Norman Ramsey. **Experience Report: Haskell in Computational Biology**. In Proc. 17th ACM SIGPLAN International Conference on Functional Programming (ICFP 2012), 227-234, 2012.

...Haskell gives computational biologists the flexibility and rapid prototyping of a scripting language, plus the performance of native code.

# Wo fkt. Progr. heute angewendet wird (2)

...über **Wirtschaft**:

- ▶ Curt J. Simpson. **Experience Report: Haskell in the “Real World”**: Writing a Commercial Application in a Lazy Functional Language. In Proceedings of the 14th ACM SIGPLAN International Conference on Functional Programming (ICFP 2009), 185-190, 2009.

...describe[s] the initial attempt of experienced business software developers with minimal functional programming background to writ a non-trivial, business-critical application entirely in Haskell. ...discuss[es] the advantages and difficulties of Haskell in these circumstances, with a particular focus on issues that commercial developers find important but that may receive less attention from the academic community.

Abhalten

Überblick

A Moti-  
vation

Einordnung

Generalthema

Leitfrage

Gliederung

Einstieg

Relevanz,  
Anwendun-  
gen

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressource

# Wo fkt. Progr. heute angewendet wird (3)

...bis **System-** und **Web-Programmierung**:

- ▶ Iavor S. Dachki, Thomas Hallgren, Mark P. Jones, Rebekah Leslie, Andrew Tolmach. [Writing System Software in a Functional Language: An Experience Report](#). In Proceedings of the 4th International Workshop on Programming Languages and Operating Systems (PLOS 2007), Article No. 1, 5 Seiten, 2007.

...we describe our experience developing a prototype operating system, House, in which the kernel, device drivers, and even a simple GUI, are all written in Haskell.

- ▶ Michael Snoyman. [Developing Web Applications with Haskell and Yesod](#). O'Reilly, 2012.

...supports high-performing applications that are modular, type-safe, and concise.

# Wo fkt. Progr. heute angewendet wird (4)

...ausgewählte sechs weitere Vorzeigebispiele:

- ▶ Philip Wadler. *An angry half-dozen*. ACM SIGPLAN Notices 33(2):25-30, 1998.

...You've scrutinized functional languages. You've admired the elegance of lambda calculus, checked the benchmarks from the compilers, noted the security provided by strong typing. Now you want to know if they have been used to serious purpose. Mathematical elegance is well and good, but will it run that mission-critical system?

Here are a half-dozen exemplars of [serious] functional programs...

...insgesamt:

- ▶ Mihai Maruseac. *Haskell: A Language for Modern Times*. Crossroads, the ACM Magazine for Students 24(1):64-66, 2017.



Also lautet der Beschluss,  
dass der Mensch was lernen muss.

*Max und Moritz*

Wilhelm Busch (1832-1908)

dt. Schriftsteller, satirischer Zeichner und Maler

## B

L<sup>3</sup>: Lernergebnisse, Lernaktivitäten,  
Leistungsnachweise

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

Lernergebnis

Lernaktivität

Leistungsnach

Gesamtbeurte

Beurteilungss

Tests

ECTS-  
Schlüssel

Kompetenzen

C  
Ablauf

D  
Ressourcen

# Lernergebnisse

Nach **positiver Absolvierung** der LVA können Sie (u.a.)

1. die **grundlegenden Prinzipien und Konzepte** des fkt. Programmierstils und seiner theoretischen Fundierung
  - 1.1 erläutern und von anderen Programmierstilen wie prozeduraler, objektorientierter, logischer Prog. abgrenzen,
  - 1.2 am Bsp. von Haskell aufzeigen und illustrieren.
2. **programmiertechnische Aufgabenstellungen** im Kleinen
  - 2.1 in aufgabenangemessene Haskell-Programme umsetzen,
  - 2.2 aussagekräftige Testfälle für ihre Validierung erstellen,
  - 2.3 die Programme m. einem Sprachinterpretierer ausführen.
3. die **Bedeutung u. Auswertungsw.** v. Haskell-Programmen erklären und schrittweise mit Papier und Bleistift für verschiedene Auswertungsordnungen ausführen.
4. das **gewählte Vorgehen** sachlich und fachlich begründen.
5. **Haskell-Prog.** auf syn./sem. Korrektheit, Allgemeinheit, Wiederverw., Eff./Perf.-eigensch., angemessene Verw. fkt. progr.-sprachl. Konzepte untersuchen und beurteilen.

Abhalten

Überblick

A Motivation

B Lernergebnisse

Lernergebnisse

Lernaktivitäten

Leistungsnahe

Gesamtbeurteilung

Beurteilungsskizzen

Tests

ECTS-Schlüssel

Kompetenzen

C Ablauf

D Ressourcen

# Lernaktivitäten, Lernprozess, Methoden

...zum Erreichen der Lernergebnisse:

1. **Angeleitetes eigenständiges Erlernen und Einüben:** Durch Vorträge (Orientierung, Einordnung) u. umgek. Klassenzimmersitz. (Übung, Vertiefung) angeleitetes eigenständiges Erlernen u. Einüben der als Lernergebnisse beschriebenen Fähigkeiten mithilfe bereit gestellter Lehr- u. Lernunterlagen, programmier- und nichtprogrammiertechnischer Übungsaufgaben u. nach Bedarf selbstgewählter weiterer Materialien aus ergänzend u. vertiefend vorgeschlagenen Lehrbüchern, Tutorien und wissenschaftl. Originalarbeiten.
2. **Vorbild- und rückmeldungsgelitetes Lernen:** Präsentieren, erläutern, begründen, vergleichen, wertend gegenüberstellen eigener und fremder Aufgabenlösungen aus sachl. und fachl. Sicht in tutorgeleiteten Übungsgruppen.
3. **Selbsteinschätzungstests:** Tests zur regelmäßigen Selbsteinschätzung und Selbstreflexion des eigenen bisherigen Lernfortschritts und Lernerfolgs.

Abhalten

Überblick

A Motivation

B Lernergebnisse

Lernergebnisse

Lernaktivitäten

Leistungsnachweise

Gesamtbeurteilung

Beurteilungsskizzen

Tests

ECTS-Schlüssel

Kompetenzen

C

Ablauf

D

Ressourcen

# Leistungsnachweise

Die Freude flieht auf allen Wegen,  
der Ärger kommt uns gern entgegen.

*Balduin Bählamm, der verhinderte Dichter*

Wilhelm Busch (1832-1908)

dt. Schriftsteller, satirischer Zeichner und Maler

## Leistungsnachweise (prüfungsimmanent)

1. **Online:** 7 beurt. Angaben programmiertechn. Aufgaben

(Punkte Angaben:  $\sum_{j=1}^7 \text{Punkte\_aus\_Angabe}_j$ )

Punkte pro Angabe:  $\frac{1}{2} * \sum_{i=1}^2 \text{Punkte\_aus\_Abgabe}_i$ )

2. **Präsenz:** 3 beurteilte 90-minütige schriftliche Tests über Vorlesungs- und Übungsstoff (mit 1 bis 3 Testteilnahmen nach eigener Wahl)

(Punkte Tests:  $\frac{1}{\# \text{Test\_Teilnahmen}} * \sum_{j=1}^3 \text{Punkte\_aus\_Test}_j$ )

Weitere beurteilte Leistungsnachweise gibt es nicht.

# Gesamtbeurteilung, Zeugnisausstellung

## Gesamtbeurteilung, Gesamtnote

Gewichtet im Verhältnis 1 zu 1 aus Beurteilungen der

- programmiersprachlichen Aufgabenlösungen
- schriftlichen Tests

falls beide Teilbeurteilungen positiv sind, sonst nicht genügend; bei halben Noten gibt grundsätzlich die Note der schriftlichen Tests den Ausschlag.

## Zeugnisausstellung

- Nach dem letzten schriftlichen Test.

# Beurteilungsschema

...ident für

- Programmiertechnische Aufgabenlösungen
- Schriftliche Tests

Erreichte Punktzahl in % der Maximalpunktzahl	Note
$\geq 87.5$	1
$\geq 75.0$	2
$\geq 62.5$	3
$\geq 50.0$	4
$< 50.0$	5

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

Lernergebnis

Lernaktivität

Leistungsnach

Gesamtbeurte

**Beurteilungss**

Tests

ECTS-  
Schlüssel

Kompetenzen

C

Ablauf

D

Ressourcen

# Zu Inhalt und Ablauf

...jedes der drei schriftlichen Tests:

- **Dauer:** 90 Minuten.
- **Teststoff:** Vorlesungs- und Übungsstoff incl. zweier zusammengehörender wissenschaftlicher Artikel, die Sie sich im Lauf der Vorlesungszeit eigenständig erschließen (zugänglich aus TUV-Netz in IEEE Digital Library):
  1. Konrad Hinsen. **The Promises of Functional Programming**. Computing in Science and Engineering 11(4): 86-90, 2009.
  2. Konstantin Läufer, George K. Thiruvathukal. **The Promises of Typed, Pure, and Lazy Functional Programming: Part II**. Computing in Science and Engineering 11(5): 68-75, 2009.
- **Hilfsmittel:** Keine.
- **Anmeldung:** Erforderlich (via **TISS**), Fristen beachten!

Abhalten

Überblick

A Motivation

B Lern-  
er-  
gebnisse

Lernergebnisse

Lernaktivitäten

Leistungsnahe

Gesamtbeurteilung

Beurteilungsskizzen

Tests

ECTS-  
Schlüssel

Kompetenzen

C Ablauf

D Ressourc

# ECTS-Punkteaufschlüsselung

<b>Angeleitete Lernaktivitäten (Hybrider Modus)</b>	
– Vortrag (7 Online-Termine * 0.75h)	5.25h
– Umgek. Klassenzimmer (7 Online-Termine * 0.5h)	3.5h
– Kleinübungen (3 Präsenz-Termine * 1.5h 4 Online-Termine * 0.75h)	4.5h 3.0h
– Plenumsübungen (7 Präsenz-Termine * 0.75h)	5.25h
<b>Eigenständige Lernaktivitäten (Home Universitying)</b>	
– Selbstgeleitetes Erarbeiten von Lernergebnissen (Richtwert: I/3.0h, II/5.0h, III/3.0h, IV/6.0h, V/6.0h, VI/4.0h, VII/1.0h)	28.0h
– Speziell: Lösen der Übungsaufgaben (Richtwert: 4 Angab. * 2.5h + 3 Angab. * 4.0h)	22.0h
– Vorbereitung auf schriftliche Tests	2.0h
<b>Schriftlicher Test (Präsenz)</b>	1.5h
<b>Summe</b>	<b>75.0h</b>

**Anm:** I bis VII beziehen sich auf Teil I bis Teil VII der LVA-Unterlagen.



# Kompetenzbereiche (adressiert; nicht ausdrückl. unterrichtet)

## Fachliche und methodische Kompetenzen

- **Fachliche K.:** Wissen um Grundlagen, Fundierung, Umsetzung und Anwendung des funktionalen Programmierstils.
- **Methodische K.:** Sach- und fachgerechte Anwendung programmiersprachlicher Konzepte des fkt. Programmierstils.

## Kognitive und praktische Kompetenzen

- **Kognitive K.:** Abstraktes und mathematisch-logisches Denken, Analyse-, Verständnis- und Problemlösefähigkeit.
- **Praktische K.:** Angemessene Anwendung von Haskell auf konkrete programmiertechnische Aufgabenstellungen.

## Soziale Kompetenzen und Selbstkompetenzen

- **Soziale K.:** Kommunikations-, Präsentations-, Argumentationsfähigkeit, Team- und Gruppenfähigkeit.
- **Selbstk.:** Selbstverantwortung, Selbstorganisation, Selbstreflexion, Wissensdrang, Hartnäckigkeit, Ausdauer,...

Abhalten

Überblick

A Motivation

B Lernergebnisse

Lernergebnisse

Lernaktivitäten

Leistungsnachweise

Gesamtbeurteilung

Beurteilungsskizzen

Tests

ECTS-Schlüssel

Kompetenzen

C

Ablauf

D

Ressourcen

# C

## Organisation, Ablauf

Abhalten

Überblick

A Motivation

B Lern-  
er-  
geb-  
nisse

**C  
Ablauf**

Anmeldung

LVA-Aufbau

Ü-Gruppen

Feedback

Termine

Caveat

D  
Ressourcen

# Anmeldung zur Lehrveranstaltung

## Anmeldung

- in TISS bis spätestens **Freitag, 09.10.2020 (12:00 Uhr)**.

## Voraussetzung einer validen Anmeldung:

- Erfolgreich abgeschlossene **STEOP** (Ausnahme: Erasmus<sup>+</sup>-Studierende).

## Studierende mit **valider Anmeldung** erhalten

- ein Benutzerkonto auf der Maschine `g0.complang.tuwien.ac.at`
- Benutzerkennung und erstes Lösungswort per elektronischer Nachricht an ihre Standardadresse `e<Matr.Nr>@student.tuwien.ac.at`

zur **Bearbeitung** und **Abgabe** von **Übungsaufgaben**.

# Aufbau der LVA nach Lernaktivitäten

Fünf Hauptlernaktivitäten (als Basis weiterer eigenst. Lernakt.):

## 1. Angeleitete Lernaktivitäten

1.1 Vortrag: Orientierung/Einordnung (7 Oline-Termine; ein-/zweiwöchentlich, dienstags, 08:15-09:00 Uhr).

1.2 Umg. Klassenzimmer: Üb./Vertief. nach Eigenstud. (7 Online-Term.; ein-/zweiw., dienstags, 09:15-09:45 Uhr).

1.3 Kleingruppenübungen (3 Präsenz-Termine zu 90min., 4 Online-Termine zu 45min.; wöchentl., versch. Zeiten).

1.4 Plenumsübungen (7 Präsenz-Termine zu 45min; wöch.)

2. Programmiertechnische Aufgaben mit Einzelabgaben (7 Angaben, wöchentlich; **beurteilt, prüfungsimmanent**).

3. Nichtprogrammiertechnische Aufgaben ohne Abgabe (4 Angaben, wöchentlich; **ohne Beurteilung**).

4. Selbsteinschätzungstests zur Vorlesung und zusätzlich als Teil der Übungsgruppenarbeit; **ohne Beurteilung**).

5. Schriftliche Tests (90min., 1, 2 oder 3 Teilnahmen nach freier Wahl; **beurteilt, prüfungsimmanent**).

# Übungsgruppenarbeit

1. **Ausgabe von Aufgaben:** Wöchentlich freitags, abrufbar auf der Webseite der LVA; insgesamt 7 Angaben, davon die ersten 4 auch mit nichtprogrammiertechnischen Aufgaben, beginnend Freitag, den 16.10.2020.
2. **Abgabe von prog.-techn. Lösungen (Erst-/Zweitabgabe):**
  - 2.1 **Erstabgabe:** Eine Woche nach Ausgabe bis 12:00 Uhr; automatische Absammlung aus Home-Verzeichnissen (oberste Verzeichnisebene, nicht in Unterverzeichnissen).
  - 2.2 **Zweitabgabe:** Nach Verbesserung der Erstabgabe (zu verlängerter Zweitabgabefrist siehe Dokument 'Allgemeine Hinweise zum Übungsablauf' auf LVA-Webseite).
  - 2.3 **Erreichte Punkte:** Hälfte der Summe beider Abgaben.
3. **Nichtprogrammiertechnische Aufgaben:** Keine Abgabe, keine Beurteilung; Vorstellung, Besprechung und Diskussion von Lösungsvorschlägen durch Teilnehmer in den Klein-, ggf. auch Plenumsübungsgruppen.

# Übungsgruppenarbeit (fgs.)

4. **Selbsteinschätzungstests:** Keine Abgabe, keine Beurteilung; je Übungsgruppentermin ein etwa 10-minütiger Test; Besprechung und Diskussion von Lösungsvorschlägen durch Teilnehmer in den Übungsgruppen.
5. **Übungsgruppenarbeit:** Präsentieren, erläutern, begründen, vergleichen, wertend gegenüberstellen eigener und fremder programmier- und nichtprogrammiertechnischer Aufgabenlösungen aus sachlicher und fachlicher Sicht.

Abhalten

Überblick

A Motivation

B Lern-  
ergebnisse

C  
Ablauf

Anmeldung

LVA-Aufbau

Ü-Gruppen

Feedback

Termine

Caveat

D  
Ressourcen

# Rückmeldungen zu Lernfortschritt, Lernerfolg

Drei Hauptformen:

## 1. Programmier- und nichtprogrammiertechnische Aufgaben

- Vorstellung, Besprechung und Diskussion ausgewählter Lösungsvorschläge in den Übungsgruppen.
- Direkte und indirekte Rückmeldung zu aktiver Beteiligung und Präsentationen in den Übungsgruppen.
- Halbautomatische Testsystemrückmeldung zu Angaben.
- Nach kapazitärer Möglichkeit: Tutorkommentierung einzelner Abgaben und Aufgaben (keine Notenrelevanz).

## 2. Selbsteinschätzungstests

- Vorstellung, Besprechung und Diskussion der Lösungen in den Übungsgruppen.
- Selbstreflexion.

## 3. Schriftliche Tests

- Einsichtnahme nach abgeschlossener Verbesserung.

Anm.: Übungsgruppe steht hier für Klein- und Plenungsübungsgruppen.

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

C  
Ablauf

Anmeldung  
LVA-Aufba  
Ü-Gruppen

**Feedback**  
Termine  
Caveat

D  
Ressource

# Geplante Termine für Vorträge und umgekehrte Klassenzimmereinheiten

...in Form von **Echtzeitvideokonferenzen**:

Vortrag, umgek. Klassenz.	Thema Vortrag	Thema umgek. Klassenz.
Di, 06.10.2020, 08:15-09:45	Teil I	n.a. (Vorbereitung)
Di, 13.10.2020, 08:15-09:45	Teil II	Teil I
Di, 27.10.2020, 08:15-09:45	Teil III	Teil II
Mi, 04.11.2020, 08:15-09:45	Teil IV	Teil III
Mi, 18.11.2020, 08:15-09:45	Teil V	Teil IV
Mi, 02.12.2020, 08:15-09:45	Teil VI	Teil V
Mi, 16.12.2020, 08:15-09:45	Teil VII	Teil VI

Die Angaben Teil I bis Teil VII beziehen sich auf die entsprechenden Teile der Lehrveranstaltungsunterlagen.



# Geplante Termine für Angaben, KÜGs u. PÜGs

...KÜGs teils in Präsenz (FAV HS 1), teils als Echtzeitvideokonf.:

Angabe	Ausgabe	Erstabgabe	Punkte	zugeh. KÜG-Termine
1	Fr, 16.10.20	Fr, 23.10.20	50	KW 45: 03.-06.11.20
2	Fr, 23.10.20	Fr, 30.10.20	50	KW 46: 09.-13.11.20
3	Fr, 30.10.20	Fr, 06.11.20	50	KW 47: 16.-20.11.20
4	Fr, 06.11.20	Fr, 13.11.20	50	KW 48: 23.-27.11.20
5	Fr, 13.11.20	Fr, 20.11.20	100	KW 49: 30.11.-04.12.20
6	Fr, 20.11.20	Fr, 27.11.20	100	KW 50: 07.-11.12.20
7	Fr, 27.11.20	Fr, 04.12.20	100	KW 51: 14.-18.12.20

...PÜGs in Präsenz (Informatik-Hörsaal):

PÜG	KW/Ort	Gruppe I	Gruppe II
1	KW 45/Inf.-HS	Di, 03.11.20, 9-10 Uhr	Mi, 04.11.20, 16-17 Uhr
2	KW 46/Inf.-HS	Di, 10.11.20, 9-10 Uhr	Mi, 11.11.20, 16-17 Uhr
3	KW 47/Inf.-HS	Di, 17.11.20, 9-10 Uhr	Mi, 18.11.20, 16-17 Uhr
4	KW 48/Inf.-HS	Di, 24.11.20, 9-10 Uhr	Mi, 25.11.20, 16-17 Uhr
5	KW 49/Inf.-HS	Di, 01.12.20, 9-10 Uhr	Mi, 02.12.20, 16-17 Uhr
6	KW 50/Inf.-HS	Di, 08.12.20, 9-10 Uhr	Mi, 09.12.20, 16-17 Uhr
7	KW 51/Inf.-HS	Di, 15.12.20, 9-10 Uhr	Mi, 16.12.20, 16-17 Uhr

Abhalten

Überblick

A Motiv-  
ation

B Lern-  
ergeb-  
nisse

C  
Ablauf

Anmeldung  
LVA-Aufba  
Ü-Gruppen

Feedback

**Termine**

Caveat

D  
Ressource

# Geplante Termine u. Orte für die schriftl. Tests

Schriftl. Tests	Datum	Zeit	Ort
Test 1	Do, 14.01.2021	16:00 - 18:00 Uhr	EI7/EI3
Test 2	Fr, 05.03.2021	15:00 - 17:00 Uhr	EI7
Test 3	Fr, 28.05.2021	15:00 - 17:00 Uhr	HS11

- Eine Anmeldung in **TISS** ist für Teilnahme an allen Terminen erforderlich (siehe **TISS** für An- und Abmeldefristen); **merken** Sie sich die Termine bitte vor und **planen Sie entsprechend!**
- Über die obigen Termine hinaus, **keine weiteren Termine für schriftliche Tests.**
- Teilnahme an allen Tests möglich, an mindestens einem verpflichtend.
- Endgültige Termine, Zeiten, Orte, Fristen, Änderungen werden in **TISS** bekanntgegeben; im Zweifel gelten die Angaben in **TISS**.

# Caveat

Alle Angaben zum geplanten Lehrveranstaltungsablauf, insbesondere zu Präsenzveranstaltungen, deren Terminen und Teilnehmerkapazitäten, gelten vorbehaltlich jederzeit möglicher erneuter Verschärfungen von COVID-19-Beschränkungen durch amtliche Stellen, Rektorat, etc.

Auszug aus einer Information von VR Matyas vom 31.07.2020:

*Wir möchten [...] darauf hinweisen, dass eine Verschärfung der Sicherheitsmaßnahmen während der Sommermonate, bzw. während des Semesters notwendig werden kann.*

...Änderungen im Lehrveranstaltungsablaufs- und -abhaltungsplan (auch erheblicher Natur) sind daher jederzeit und auch kurzfristig möglich!

# D

## Ressourcen

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

C  
Ablauf

**D**  
Ressourcen

Lehrbücher

Zeitschriften,  
Konferen-  
zen

Rechner

Bei Fragen

Tutoren

Erasmus

Viel Erfolg!

# Lehrbücher, wiss. Arbeiten, Haskell im Netz

...gezielte Hinweise auf

## ▶ Lehrbücher, wissenschaftliche Arbeiten

- Siehe Vorlesungsunterlagen, detaillierte Leseempfehlungen am Ende jedes Kapitels.

## ▶ Haskell im Netz

- Haskell-Homepage: [www.haskell.org/](http://www.haskell.org/)
- Haskell-Wiki: [wiki.haskell.org/Haskell/](http://wiki.haskell.org/Haskell/)
- Haskell-Tutorial: [www.haskell.org/tutorial/](http://www.haskell.org/tutorial/)
- Glasgow Haskell Compiler GHC und Interpretierer GHCi: [hackage.haskell.org/platform/](http://hackage.haskell.org/platform/)
- Hugs-Interpretierer: [www.haskell.org/hugs](http://www.haskell.org/hugs)

## ▶ Haskell auf youtube

- Interview mit John Hughes über 'Funktionale Programmierung und Haskell'.  
<https://www.youtube.com/watch?v=LnX3B9oaKzw>

Abhalten

Überblick

A Motivation

B Lernergebnisse

C Ablauf

D Ressourcen

Lehrbücher  
Zeitschriften,  
Konferenzen  
Rechner  
Bei Fragen  
Tutoren  
Erasmus  
Viel Erfolg!

# Wichtige wiss. Zeitschriften und Konferenzen

...zur Publikation von Forschungsergebnissen im Umfeld funktionaler Programmierung und von Haskell sind besonders:

## ► Zeitschriftenreihe:

- The *Journal of Functional Programming*. Matthias Fellisen, Jeremy Gibbons (Hrsg.), Cambridge, UK, seit 1991.

<https://www.cambridge.org/jfp>

## ► Konferenz- und Symposiumsreihen:

- ACM SIGPLAN International Conference Series on Functional Programming (ICFP), jährlich seit 1996.
- ACM SIGPLAN International Haskell Symposium Series, jährlich seit 2008 (2002-2007 als ACM SIGPLAN Haskell Workshop Series).

<https://www.haskell.org/haskell-symposium>

Abhalten

Überblick

A Motivation

B Lern-  
er-  
gebnisse

C Ablauf

D Ressource

Lehrbücher  
Zeitschriften,  
Konferenzen

Rechner

Bei Fragen  
Tutoren

Erasmus  
Viel Erfolg!

# Arbeit mit TUW- und eigenen Rechnern

- Server für die praktischen Programmierübungen: [g0.complang.tuwien.ac.at](http://g0.complang.tuwien.ac.at)
- Arbeit mit TUW-Rechnern: Grundsätzl. verfügbar im Labor Argentinierstr. 8, Erdgeschoss im Innenhof; **aufgrund von COVID-19-Beschränkungen vorauss. nicht nutzbar!**
- Arbeit mit anderen/eigenen Rechnern: **Ist möglich (z.B. zu Hause) und aufgrund von COVID-19-Beschränkungen unvermeidbar.**
- Abgaben von Übungsaufgaben: Ausschließlich auf dem Server [g0.complang.tuwien.ac.at](http://g0.complang.tuwien.ac.at).
- Nötige Software: GHC/GHCi, Hugs (frei verfügbar).
- **Wichtig:** Abgaben werden auf der [g0](http://g0) ausschließlich unter GHCi getestet. Überzeugen Sie sich deshalb stets von der gewünschten Funktionalität Ihrer Programmierlösungen auf der [g0](http://g0) unter GHCi!

Abhalten

Überblick

A Motivation

B Lern-  
ergebnisse

C Ablauf

D Ressourc

Lehrbücher  
Zeitschriften,  
Konferenzen

Rechner

Bei Fragen  
Tutoren

Erasmus  
Viel Erfolg!

# Anlaufstellen

...bei Fragen und Problemen:

- Webseite der LVA:  
[www.complang.tuwien.ac.at/knoop/fp185A03.html](http://www.complang.tuwien.ac.at/knoop/fp185A03.html)
- Umgekehrtes Klassenzimmer im Anschluss an den Vortragsteil
- Klein-/Plenumsübungsgruppe

## Vorlesungsmaterialien, Angaben, Termine

Denn was man schwarz auf weiß besitzt,  
kann man getrost nach Hause tragen.

Faust. Eine Tragödie.

Johann Wolfgang von Goethe (1749-1832)  
dt. Dichter und Naturforscher

- Webseite der Lehrveranstaltung:

[www.complang.tuwien.ac.at/knoop/fp185A03\\_ws2021.html](http://www.complang.tuwien.ac.at/knoop/fp185A03_ws2021.html)



# Tutoren im WS 2020/2021

1. Jana Chadt, BSc
2. Lukas Grassauer, BSc
3. Niki Herl
4. Samuel Pilz, BSc
5. Hannes Siebenhandl, BSc

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressourcen

Lehrbücher  
Zeitschriften,  
Konferen-  
zen

Rechner

Bei Fragen

**Tutoren**

Erasmus

Viel Erfolg!

# Interesse an gefördertem Auslandsstudium?

Die **Erasmus/LLP-Programmlinie** der EU bietet eine Vielzahl lohnender Möglichkeiten, z.B.

- Linköping University, Schweden
- Aalto University, Finnland
- The University of Copenhagen, Dänemark
- Universität Halle-Wittenberg, Deutschland
- Universität Paderborn, Deutschland
- Universidad Politècnica de València, Spanien
- ...

Mehr dazu: [www.complang.tuwien.ac.at/knoop/erasmus](http://www.complang.tuwien.ac.at/knoop/erasmus)

# Zum Vorbesprechungsabschluss

Dabei sein ist  
80 Prozent des Erfolges.

Woody Allen (\* 1935)  
amerik. Schauspieler und Regisseur

...wir, die FP-Teammitglieder, wünschen Ihnen viel (Lern-) Erfolg für diese Lehrveranstaltung und dass Sie möglichst großen Nutzen daraus ziehen, auch langfristig!

**Nicht zuletzt:**

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

Abhalten

Überblick

A Moti-  
vation

B Lern-  
ergeb-  
nisse

C  
Ablauf

D  
Ressourcen

Lehrbücher  
Zeitschriften,  
Konferenzen

Rechner

Bei Fragen

Tutoren

Erasmus

Viel Erfolg!