

Pro-Pla

Tutorium – Gruppe 2 & 8

Henning Dieterichs

Organisatorisches

- Henning Dieterichs, henning-propa@outlook.de
- Schreibt mich für Anregungen, Fragen und Feedback jederzeit an
- Folien: <https://github.com/hediet/propa2017>
- Vorlesungsseite: <https://pp.ipd.kit.edu/lehre/WS201718/paradigmen/>
- Übungsseite: <https://pp.ipd.kit.edu/lehre/WS201718/paradigmen/uebung/>
- Es gibt ein Forum:
https://ilias.studium.kit.edu/goto.php?target=crs_732265&client_id=produktiv
- Abgabe: https://praktomat.cs.kit.edu/pp_2017_WS

Übungsblätter

- Abgabe Donnerstags 11:30 Uhr
- Besprechung/Korrektur zum nächsten Tutorium
- Analoge Abgaben
 - Einwurf in die Briefkästen im UG von Geb. 50.24
- Digitale Abgaben
 - Praktomat
- Abgabe NUR für euch!
- Beam Teamarbeit: Bitte nur einer abgeben! (Egal wer)

Das Grauen: Propa Klausur

- 120 Minuten
- 120 Punkte
- Alles aus **Papier** darf mitgenommen werden
- Lernt auf Verständnis und nicht auswendig!
 - Dann ist die Klausur leicht



Wer von euch ...

- hat vor,
 - die Klausur zu schreiben?
 - die Vorlesung zu besuchen?
 - Übungsblätter abzugeben?
- wünscht sich
 - ein interaktives Tutorium? / ein nicht-interaktives Tutorium?
 - die Besprechung der Aufgaben?
 - Wiederholung? / Vertiefung?
 - Praxis? / Theorie?
- kennt noch
 - Reguläre Ausdrücke?
 - Kontextfreie Grammatiken?

Einrichtung

- Hat jemand Probleme?
- Online: <https://repl.it/languages/haskell>
 - Nur für wenig Quell-Code geeignet!
- GHCi
 - `:quit` Beenden
 - `:load file.hs / :l file.hs` Lädt Datei
 - `:reload / :r` Lädt alle geladenen Dateien neu

Allgemeines zu Haskell: Literale und Variablen

- Literale („literal“)
 - Integer/Float: $[0-9]^+ / [0-9]^+.[0-9]^+$
 - Strings: `“ . * “`
- Variablen
 - varid: $[_a-zA-Z'][_a-zA-Z0-9']^*$
 - var: varid oder (varsym)
 - varsym: $[!#\$\%&*+./<=>?\@\\^|-~:]^+$
 - varop: varsym oder `varid`

Allgemeines zu Haskell - Programmaufbau

- Formale Grammatik:
 - <https://www.haskell.org/onlinereport/haskell2010/haskellch10.html>

Erste Schritte mit Haskell

Definieren Sie eine Funktion `max3 x y z` welche das Maximum dreier Zahlen zurück gibt.
Geben Sie dabei Varianten an, die ausschließlich

- **`if .. then .. else`** verwendet
- *guard*-Notation verwendet (siehe Vorlesungs-Beispiel `binom n k`)
- die vordefinierte Funktion **`max`** `x y` verwendet

Haskell



<http://xkcd.com/1312/>