

Hinweise zu Organisation und Ablauf des Übungsteil der LVA 185.A03 Funktionale Programmierung im WS 2019/20

Programmiermöglichkeit im Labor “Argentinierstr.”

Für die Bearbeitung, Lösung und Abgabe der Programmieraufgaben stehen Ihnen (im Rahmen der Kapazitäten) die Rechner im Labor im Erdgeschoss des Institutsgebäudes Argentinierstraße 8 zur Verfügung. Sie erreichen dieses Labor über den kleinen Innenhof im Erdgeschoss. Einen Lageplan finden Sie auf

<http://www.complang.tuwien.ac.at/ulrich/p-1851-E.html>.

Um die Aufgaben zu lösen, rufen Sie bitte den Hugs 98-Interpreter durch Eingabe von `hugs` in der Kommandozeile einer Shell auf. Falls Sie die Übungsaufgaben auf Ihrem eigenen Rechner bearbeiten möchten, müssen Sie zunächst Hugs 98 installieren. Hugs 98 ist beispielsweise unter www.haskell.org/hugs/ für verschiedene Plattformen verfügbar. Der Aufruf der jeweiligen Interpretierervariante ist dann vom Betriebssystem abhängig.

Ausgabe von Angaben, Erst- und Zweitabgabe von Lösungen

Angaben werden stets montags auf der Webseite

<http://www.complang.tuwien.ac.at/knoop/lehre/ws1920/fp185A03>

ausgegeben und können dort heruntergeladen werden.

Die Erstabgabe von Lösungen erfolgt am Montag (12:00 Uhr) eine Woche nach Ausgabe einer Angabe.. Der Zweitabgabezeitpunkt richtet sich nach dem Beurteilungszeitpunkt der Erstabgabe (siehe Abschnitt “Abgabezeitpunkt von Zweitabgaben”).

Die Abgabedatei für jede Angabe muss in Ihrem Home-Verzeichnis (keinesfalls in einem Unterverzeichnis!) auf der Maschine

`g0.complang.tuwien.ac.at`,

abgelegt werden (ggf. mittels `ssh/scp` oder vergleichbarer Dienstprogramme, wenn Sie nicht unmittelbar auf der `g0` arbeiten). Der Name für die Abgabedatei ist auf dem Angabenblatt stets vorgegeben. Dieser Name ist für Erst- und Zweitabgabe zu einer Angabe ident.

Benachrichtigung über erreichte Punkte

Für die ersten vier Angaben sind je **50** Punkte erreichbar, für die letzten drei je **100** Punkte. Die Ergebnisse der Bewertung Ihrer abgegebenen Lösungen wird in einer Datei in Ihrem Home-Verzeichnis auf der `g0` abgelegt.

Abgabezeitpunkt für Zweitabgaben

Für jede Angabe können Sie nach Erhalt der Bewertung der Erstabgabe Verbesserungen vornehmen und als zweite Lösung abgeben. Der Zeitpunkt für diese zweite Abgabe richtet sich dabei nach dem Zeitpunkt, an dem Sie die Bewertung der Erstabgabe erhalten.

Es ist der Montag (12:00 Uhr), der auf den Freitag folgt, an dem Sie die Ergebnisse der Erstabgabe spätestens um 09:00 Uhr erhalten haben. Im Normalfall ist dies der Montag zwei Wochen nach der Ausgabe einer Angabe.

Wichtig

Vorhalten von Aufgabenlösungen zu beiden Abgabeterminen für eine Angabe

Zu beiden Abgabeterminen für eine Angabe und bis zum Erhalt der jeweiligen Bewertung (gleich ob Erst- oder Zweitabgabe für eine Angabe) müssen Sie ihre Lösungen der Programmieraufgaben in einer entsprechend der Aufgabenvorgabe benannten Datei im Homeverzeichnis Ihres Accounts auf dem Übungsrechner `g0` zum automatischen Absammeln abgelegt bereit halten. Ihre Abgabedatei wird zu den genannten Zeitpunkten automatisch aus ihrem Home-Verzeichnis kopiert. Findet sich dort keine Abgabedatei, erhalten Sie für diesen Abgabetermin 0 Punkte.

Ihre Abgabedatei darf deshalb **keinesfalls** in einem Unterverzeichnis Ihres Home-Verzeichnisses stehen oder anders benannt sein als auf der Angabe vorgegeben.

Arbeit mit Hugs und GHCi; Bewertung ausschließlich unter Hugs

Sie können Ihre Aufgabenlösungen sowohl mit Hugs als auch mit GHCi vorbereiten. Die **Bewertung** Ihrer Aufgabenlösungen erfolgt jedoch **ausschließlich unter Hugs in der auf dem Übungsrechner `g0` installierten Version.**

Wenn Sie Ihre Aufgabenlösungen nicht direkt auf der `g0` (im Labor “Argentinierstr.” oder über eine `ssh`-Verbindung) unter Hugs entwickeln, müssen Sie sich deshalb nach erfolgter Übertragung Ihrer Abgabedatei auf die `g0` (z.B. über eine `scp`-Verbindung) **unbedingt vergewissern**, dass Ihre möglicherweise mit einem anderen Werkzeug entwickelte Lösung **auch unter Hugs auf dem Übungsrechner `g0` wie von Ihnen erwartet arbeitet.**

Machen Sie sich rechtzeitig mit der technischen Abwicklung von Aufgabenabgabe und Überprüfung des erwarteten Programmverhaltens auf der `g0` **vertraut, in jedem Fall rechtzeitig vor dem ersten Abgabetermin.**

Online-Tutorien zu Haskell und Hugs

Unter www.haskell.org/hugs/pages/hugsman/ finden Sie ein *Online*-Manual für Hugs 98. Lesen Sie in jedem Fall die ersten Abschnitte des Manuals, darunter Abschnitt 3 zum Thema „Hugs for beginners“ und probieren Sie die darin beschriebenen Beispiele aus. Machen Sie sich mit dem Haskell-Interpreter zumindest so weit vertraut, dass Sie problemlos einfache Ausdrücke auswerten lassen können.

Ein weiteres *Online*-Tutorial zu Haskell finden Sie auf haskell.org/tutorial/. Hinweise zur Arbeit mit Hugs finden Sie auch in Kapitel 4 aus:

- H. Conrad Cunningham. *Notes on Functional Programming with Haskell*. Course Notes, University of Mississippi, 2007 (Chapter 4, Using the Hugs Interpreter).
citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.2822&rep=rep1&type=pdf

Fragen von allgemeinem Interesse zu Vortrag und Übung können Sie auch über das TISS-Forum zur Lehrveranstaltung zur Diskussion stellen.

Hinweise zu Haskell und möglichen (Anfangs-) Klippen und Hürden

Die Syntax von Haskell birgt im großen und ganzen keine besonderen Fallstricke und ist zumeist intuitiv, wenn auch im Vergleich zu anderen Sprachen anfangs möglicherweise ungewohnt und deshalb “gewöhnungsbedürftig”.

Eine Hürde für neu mit Haskell beginnende Programmierer sind Einrückungen. Einrückungen tragen in Haskell Bedeutung für die Bindungsbereiche und damit für die Programmbedeutung. Sie müssen deshalb unbedingt eingehalten werden (vgl. z.B. Kap. 3.7 der Vorlesung zu „Funktions- und Programmlayout, Abseitsregel“). Alles, was zum selben Bindungsbereich gehört, muss in derselben Spalte beginnen. Diese in ähnlicher Form auch in anderen Sprachen wie `Occam` vorkommende Konvention erlaubt es, Strichpunkte und Klammern einzusparen.

Ein Anwendungsbeispiel in Haskell: Wenn eine Funktion mehrere Zeilen umfasst, muss alles, was nach dem „`=`“ steht, in derselben Spalte beginnen oder noch weiter eingerückt sein als in der ersten Zeile. Anderenfalls liefert Hugs dem Haskell-Programmierbeginner (scheinbar) unverständliche Fehlermeldungen wegen fehlender Strichpunkte.

Weiters sollen alle Funktionsdefinitionen und Typdeklarationen in Haskell-Programmen in derselben Spalte ganz links beginnen. Verwenden Sie keine Tabulatoren oder stellen Sie die Tabulatorgröße auf acht Zeichen ein.

Achten Sie auf richtige Klammerung, auch wenn Haskell vielfach keine Klammern verlangt, da sie gemäß geltender Prioritätsregeln automatisch ergänzt werden können. Im Zweifelsfall ist es gute Praxis, ggf. überflüssig zu klammern, um „Überraschungen“ zu vermeiden.

Operatoren haben immer eine niedrigere Priorität als das Hintereinanderschreiben von Ausdrücken (d.h. Funktionsanwendungen). Der Unterstrich „`_`“ zählt zu den Kleinbuchstaben. Wenn Sie nicht sicher sind, verwenden Sie lieber mehr Klammern als aufgrund von Klammereinsparungsregeln möglicherweise nötig sind.

Beachten Sie, dass das Minussymbol kontextabhängig als Infix- (Subtraktion) oder Prefix-Operator (Vorzeichenwechsel) interpretiert wird.

Achtung: Der Ausdruck „`potenz 2 -1`“ entspricht dem Ausdruck „`(potenz 2) - (1)`“ (oder kürzer: „`potenz 2 - 1`“) und nicht dem Aufruf „`potenz 2 (-1)`“, wie man möglicherweise erwarten könnte.

Funktionsdefinitionen und Typdeklarationen können Sie nicht direkt im Haskell-Interpreter schreiben, sondern nur aus Dateien laden. Genauere Hinweise zur Haskell-Syntax finden Sie z.B. unter haskell.org/tutorial/.