

185.A03 Funktionale Programmierung WS 22

Freitag, 07.10.2022

TdT-Angabe

‘Teste das Testsystem’

1. **Testsystemlauf** für Abgaben von `Angabe_TdT.hs` bis Mi, 12.10.2022, 12:00 Uhr
2. **Testsystemlauf** für Abgaben von `Angabe_TdT.lhs` bis Fr, 14.10.2022, 12:00 Uhr
(Keine Beurteilung der Abgaben der TdT-Angabe!)

Ziel und Zweck der TdT-Angabe:

Bearbeitung und Abgaben der TdT-Angabe in Form eines *gewöhnlichen* (für Testsystemlauf 1) und *literaten* (für Testsystemlauf 2) Haskell-Skripts (siehe dazu Kapitel 1.3.2 der Vorlesungsunterlagen)

1. sind freiwillig, nicht verpflichtend, nicht beurteilt.
2. bieten Ihnen Gelegenheit,
 - (a) das Kopieren von Dateien in Ihr Heimverzeichnis auf die Maschine (keine Webseite!)

`g0.complang.tuwien.ac.at`

mittels `scp`-Verbindung (*secure copy*) und das entfernte Arbeiten auf der `g0` mittels `ssh`-Verbindung (*secure shell*) praktisch zu erproben.
 - (b) die GHCi- oder/und Hugs-Programmierungsumgebung auf Ihrem eigenen Rechner auf Funktionalität und das korrekte Zusammenspiel mit dem Testsystem zu überprüfen und dass Ihre Abgabedateien den vom Testsystem geforderten Formateigenschaften genügen.
 - (c) verschiedene positive wie negative Rückmeldungsarten des Testsystems im Erfolgs- bzw. Misserfolgsfall kennenzulernen.
 - (d) den Unterschied zwischen gewöhnlichen und literaten Haskell-Skripten praktisch zu erproben.

Zu diesem Zweck wird es zwei Läufe des Testsystems geben: den ersten Testsystemlauf für ein

- gewöhnliches Haskell-Skript in der (Abgabe-) Datei `Angabe_TdT.hs`

für Abgaben bis Mittwoch, 12.10.2022, 12:00 Uhr; den zweiten Testsystemlauf für das entsprechende

- literate Haskell-Skript in der (Abgabe-) Datei `Angabe_TdT.lhs`

für Abgaben bis Freitag, 14.10.2022, 12:00 Uhr. Beide Abgabedateien müssen Sie bis zum Ende der jeweiligen Abgabefrist auf oberstem Niveau in Ihrem Heimverzeichnis abgelegt haben, um vom jeweiligen Testsystemlauf profitieren zu können.

Aufgabe 1 für den 1. Testsystemlauf mit gewöhnlichem Haskell-Skript:

Schreiben (oder kopieren) Sie die nachstehenden Funktionen in Ihre Abgabedatei `Angabe_TdT.hs`. Erweitern Sie dazu die im TUWEL-Kurs zur Lehrveranstaltungs zur Verfügung gestellte gleichnamige Rahmendatei.

Teil A: Zum Kennenlernen positiver Rückmeldungen des Testsystems im Erfolgsfall.

```
type Nat0 = Integer

fakultaet :: Nat0 -> Nat0
fakultaet n
  | n >= 1 = n * fakultaet (n-1)
  | n == 0 = 1

fibonacci :: Nat0 -> Nat0
fibonacci n
  | n >= 2 = fibonacci (n-1) + fibonacci (n-2)
  | n == 1 = 1
  | n == 0 = 0

binom :: (Nat0,Nat0) -> Nat0
binom (n,k) = div (fakultaet n) (fakultaet k * fakultaet (n-k))
```

Teil B: Zum Kennenlernen negativer Rückmeld. des Testsystems im Misserfolgsfall.

```
-- Inhaltlich fehlerhafte Implementierung
fak_fehlerhaft :: Nat0 -> Nat0
fak_fehlerhaft n
  | n < 0    = fak_fehlerhaft (-n)
  | n == 0   = 1
  | n <= 10  = n * fak_fehlerhaft (n-1)
  | n <= 15  = n + fak_fehlerhaft (n-1)
  | True     = fak_fehlerhaft 42

-- Inhaltl. fehlerhaft, zusätzl. fehlt die vom Testsystem geforderte Typsignatur
fib_fehlerhaft 0 = 0
fib_fehlerhaft 1 = 1
fib_fehlerhaft n = fib_fehlerhaft (n-2) * fib_fehlerhaft (n-1)

binom_fehlerhaft :: (Nat0,Nat0) -> Nat0
binom_fehlerhaft (n,k) = div (fakultaet n) (fakultaet k * fak_fehlerhaft (n-k))

-- Inhaltlich korrekte, aber Zeitüberschreitungen erzwingende Implementierung
binom_langsam :: (Nat0,Nat0) -> Nat0
binom_langsam (n,k)
  | k == 0 || n == k = 1
  | True              = binom_langsam (n-1,k-1) + binom_langsam (n-1,k)
```

Aufgabe 2 für den 2. Testsystemlauf mit literatem Haskell-Skript:

Schreiben Sie für den zweiten Testsystemlauf Ihr

- gewöhnliches Haskell-Skript in Datei `Angabe_TdT.hs`

in ein

- literates Haskell-Skript in Datei `Angabe_TdT.lhs`

um (siehe Kapitel 1.2.6 der Vorlesungsunterlagen zum Unterschied von gewöhnlichen und literaten Haskell-Skripten). Nutzen Sie auch hier die im TUWEL-Kurs zur Lehrveranstaltung zur Verfügung gestellte Rahmendatei `Angabe_TdT.lhs`.

Wichtig:

1. Befolgen Sie die Hinweise in den Begleitdateien zur TdT-Angabe sorgfältig, um ein reibungsloses Zusammenspiel Ihrer Abgaben mit dem Testsystem sicherzustellen.
2. Speichern Sie alle Funktionen, die für die TdT-Angabe zu implementieren sind, für den
 - 1. Testsystemlauf in einer Datei namens `Angabe_TdT.hs`
 - 2. Testsystemlauf in einer Datei namens `Angabe_TdT.lhs`

und legen Sie sie auf der `g0` auf oberstem Niveau in Ihrem Heimverzeichnis ab.

3. Wichtig für spätere Abgaben: *Verwenden Sie keine selbstdefinierten Module!* Wenn Sie Funktionen wiederverwenden möchten, die Sie für frühere Angaben geschrieben haben, kopieren Sie diese Funktionen in die jeweils aktuelle Abgabedatei. Import-Anweisungen für selbstdefinierte Module schlagen im Zusammenspiel mit dem Testsystem fehl: das Testsystem kopiert stets nur die jeweilige Abgabedatei, keine anderen Dateien oder Module!

Iucundi acti labores.

Getane Arbeiten sind angenehm.

Cicero (106 - 43 v.Chr.)

röm. Staatsmann und Schriftsteller