

Einführung in die funktionale Programmierung

Wintersemester 2021/2022

Aufgabenblatt Nr. 7

Abgabe: Montag 14. Februar 2022, Besprechung 18. Februar 2022

Aufgabe 1 (50 Punkte) Monadisches Programmieren und Typen.

Erinnerung, Hinweise: die Typklasse `Monad` und die `do`-Notation ist im Skript definiert und in den Folien zum monadischem Programmieren.

Die Typklasse `Monad` ist eine Konstruktorklasse und definiert als

```
class Monad m where
  (>>=) :: m a -> (a -> m b) -> m b
  (>>)  :: m a -> m b -> m b
  return :: a -> m a
  fail   :: String -> m a

m >> k      = m >>= \_ -> k
fail s      = error s
```

`IO` ist Instanz der Typklasse `Monad`. Als Beispiele die Typen der Ein- und Ausgabefunktion `getChar :: IO Char` und `putChar :: Char -> IO ()`.

Die Aufgaben:

- (30 Punkte) Betrachten Sie die Funktion `sequence`, definiert wie folgt.

```
sequence :: (Monad m) => [m a] -> m [a]
sequence []           = return []
sequence (action:as) = do
  r <- action
  rs <- sequence as
  return (r:rs)
```

Welche Typen haben die einzelnen Unterausdrücke der Definition?

- (20 Punkte) Wie benutzt man `sequence` um einen String so auszudrucken, dass jedes Zeichen einzeln ausgegeben wird? Geben Sie die Programmierung an.