Hochschule Flensburg
Fachbereich 3
Angewandte Informatik

Jan Christiansen

Hochschule Flensburg
University of Applied Sciences

2. Übung zur Vorlesung „Fortgeschrittene funktionale Programmierung“

Algebraische Datenstrukturen
Labor am Montag, 7. Oktober 2016, 12:15 Uhr

Aufgabe 1 - Graphiken

Definieren Sie ein Modul *Graphics*, indem Sie eine neue Datei mit dem Namen `Graphics.hs` erzeugen und die Datei mit

module *Graphics* **where**

starten.

Implementieren Sie einen Datentyp *Point*, der zur Darstellung von Punkten auf eine 2D-Ebene genutzt werden kann. Definieren Sie basierend darauf einen Datentyp *Object*, der ein Rechteck und einen Kreis darstellen kann. Ein Rechteck wird dabei durch zwei Punkte spezifiziert und ein Kreis durch einen Mittelpunkt und einen Radius.

Definieren Sie einen Datentyp *Color*, der die Farben Schwarz, Rot, Grün und Blau zur Verfügung stellt. Definieren Sie einen Datentyp *Style*, der nur einen Wert vom Typ *Color* enthält. Erweitern Sie die Konstruktoren des Datentyp *Object* jeweils um einen Wert vom Typ *Style*. Schreiben Sie eine Funktion $styleToAttr :: Style \to String$, die einen Stil in seine CSS-Darstellung überführt und die Farbe für *stroke* und *fill* nutzt, also zum Beispiel `"stroke: black; fill: black"` für die Farbe Schwarz liefert. Definieren Sie zu guter Letzt eine Konstante $defaultStyle :: Style$, die als Farbe Schwarz verwendet.

Definieren Sie dann einen Datentyp *Graphic*, der eine Menge von Objekten darstellt. Das heißt, ein Wert vom Typ *Graphic* kann entweder eine leere Graphik darstellen oder ein Objekt und eine weitere Graphik.

Definieren Sie für Ihre Datentypen nun die folgenden Funktionen.

- Die Funktion $single :: Object \to Graphic$ wandelt ein einzelnes Objekt in eine Graphik um.
- Der Infixoperator $(<>) :: Graphic \to Graphic \to Graphic$ nimmt zwei Graphiken und vereinigt sie.
- Die Funktion $objToSVG :: Object \to String$ nimmt ein Objekt und liefert die entsprechende SVG-Repräsentation des Objektes. Eine Linie wird zum Beispiel durch einen Tag `<rect>` und ein Kreis durch den Tag `<circle>` dargestellt. Informieren Sie sich über die Attribute dieser Tags.

Schreiben Sie schließlich eine Funktion $toSVG :: Graphic \to String$, die eine Graphik in einen SVG-Tag umwandelt.

Schreiben Sie Funktionen $rectangle :: Double \to Double \to Graphic$ und $circle :: Double \to Graphic$, um entsprechende Graphiken zu erzeugen. Die Graphiken verwenden dabei den *defaultStyle*. Dabei wird bei der Erzeugung eines Rechtecks Breite und Höhe und bei einem Kreises nur der Radius angegeben. Die beiden Objekte werden jeweils so positioniert, dass sie oben links in der Ecke sitzen.

Definieren Sie ein weiteres Modul in einer anderen Datei. Mit Hilfe des Befehls

import *Graphics*

können Sie das zuvor definierte Modul importieren. Definieren Sie eine Konstante *graphic*, die eine Graphik mit einer Linie und einem Kreis erzeugt. Nutzen Sie die folgende *main*-Funktion, um diese Graphik in eine Datei zu schreiben.

```
main :: IO ()
main = writeFile "graphic.svg" (toSVG graphic)
```

Testen Sie das Ergebnis in einem Browser.

Schreiben Sie eine Funktion $colored :: Color \to Graphic \to Graphic$, welche die Farbe in einer gesamten Graphik ändert. Das heißt, die Funktion nimmt eine Graphik und eine Farbe und ersetzt alle Vorkommen einer Farbe in der Graphik durch die neue Farbe.

Hinweis: Bitte geben Sie die Lösungen der Aufgaben per Mail oder auf Papier bis zum 14.10. ab. Die Papierlösungen können im ersten Stock von Haus A in den Schrank in das entsprechende Fach geworfen werden.