

Übungen zu „Konzepte von Programmiersprachen“, WS 2010/11

Prof. Dr. R. Loogen · Fachbereich Mathematik und Informatik · Hans-Meerwein-Straße, D-35032 Marburg

Nr. 6, Abgabe: Dienstag, 30. November 2010 vor der Vorlesung

14. Fraktale Bilderzeugung mit der SOE-Graphik-Bibliothek

6 Punkte

- (a) Schreiben Sie Funktionen

/ 2,5

```
rectPP, circPP, arrowPP :: Point -> Point -> Graphic
```

die ein Rechteck, einen Kreis bzw. eine dreieckige Pfeilspitze durch die beiden angegebenen Punkte definieren. Die Pfeilspitze sollte zum zweiten Punkt zeigen. Achten Sie darauf, dass Ihre Funktionen unabhängig von der relativen Lage der Punkte im Koordinatensystem funktionieren.

Hinweis: Verwenden Sie zur Quadratwurzelberechnung über ganzen Zahlen die Funktion `round.(sqrt.fromIntegral) :: Int -> Int`.

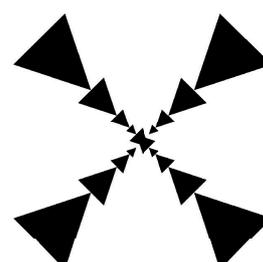
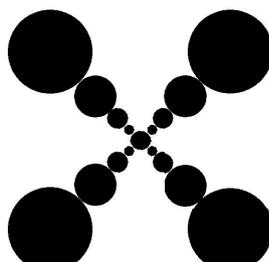
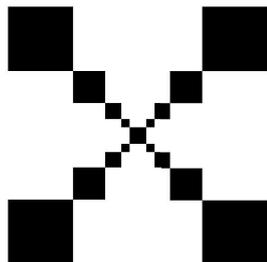
- (b) Schreiben Sie eine rekursive Funktion

/ 3

```
drawRecPic :: Window -> (Point -> Point -> Graphic) ->
```

```
Point -> Point -> Color -> Bool -> IO ()
```

die mit den in (a) definierten Graphiken in einem durch zwei Punkte vorgegebenen Gebiet fraktale Bilder der folgenden Form in der vorgegebenen Farbe erzeugt:



Das Boolesche Argument ermöglicht durch die Eingabe von `True` einen Farbwechsel zwischen den Rekursionsstufen, der bei Eingabe von `False` unterbleibt.

- (c) Schreiben Sie ein Programm zur Fensterausgabe der Graphiken.

/ 0,5

15. map - filter - foldr - zipWith

3 Punkte

Definieren Sie als Instanz vordefinierter Funktionen höherer Ordnung:

- (a) `pairAndOne :: Num a => [a] -> [(a,a)]` paart jedes Element einer Liste von Zahlen mit seinem Nachfolger. Beispiel: `pairAndOne [1,2] =>* [(1,2),(2,3)]`

/ 0,5

- (b) `remove :: Eq a => a -> [a] -> [a]` löscht in einer Liste alle Vorkommen eines Elementes. Beispiel: `remove 'u' "substitute" =>* "sbstitte"`

/ 0,5

- (c) `numTwins :: Eq a => [a] -> Int` zählt in einer Liste, wie oft benachbarte Elemente gleich sind. Beispiel: `numTwins "Hallo, Otto!" =>* 2`

/ 1

- (d) `multNeighbours :: Num a => [a] -> [a]` multipliziert je zwei aufeinanderfolgende Zahlen einer Liste. Beispiel: `multNeighbours [1,2,3,4] =>* [2,6,12]`

/ 1

16. Listenfaltung

3 Punkte

Definieren Sie die folgenden Prelude-Funktionen als Instanz von `foldr` oder `foldl`.

- (a) `unlines :: [String]->String` (b) `takewhile :: (a->Bool)->[a]->[a]`

/ 1+2