

Übungen zu Softwareentwicklung III, Funktionale Programmierung

Blatt 6, Woche 7

Leonie Dreschler-Fischer

WS 2019/2020

Ausgabe: Freitag, 22.11.2019

Bearbeitung im Tutorium: Am Mittwoch, 04.12.2019

Ziel: Rekursion: Die Aufgaben auf diesem Zettel dienen dazu, sich mit dem Entwurf von rekursiven Funktionen vertraut zu machen. Sie entwerfen linear rekursive Funktionen und üben die unterschiedlichen Formen von Rekursion zu unterscheiden.

Vorstellung in den Übungen und Abnahme: Am Mittwoch, 11.12.2019

Bearbeitungsdauer: Die Bearbeitung sollte insgesamt nicht länger als 5 Stunden dauern.

1 Formen der Rekursion

(Bearbeitungszeit 1/2 Std.), 10 Pnkt.

Gegeben seien die folgenden Funktionsdefinitionen:

```
(define (binKo n k)
  (if (or (= k 0) (= n k))
      1
      (+ (binKo (- n 1) (- k 1))
          (binKo (- n 1) k))))

(define (three n)
  (threeX n 0))

(define (threeX n k)
  (if (= n 0)
      k
      (threeY (- n 1) k)))

(define (threeY n k)
  (if (= n 0)
      k
      (threeZ (- n 1) k)))

(define (threeZ n k)
  (if (= n 0)
      k
      (threeX (- n 1) (+ k 1))))

(define (field x y)
  (cond [(= 0 y) 0]
        [(= 0 x) (* 2 y)]
        [(= 1 y) 2]
        [else (let ([rec (field x (- y 1))])
                  (field (- x 1) rec))]))

(define (even n)
  (if (= n 0)
      #t
      (if (and (= (random 2) 0) (= (modulo n 2) 0))
          (or (even n) (even (- n 1)))
```

```
(odd (- n 1))))))
```

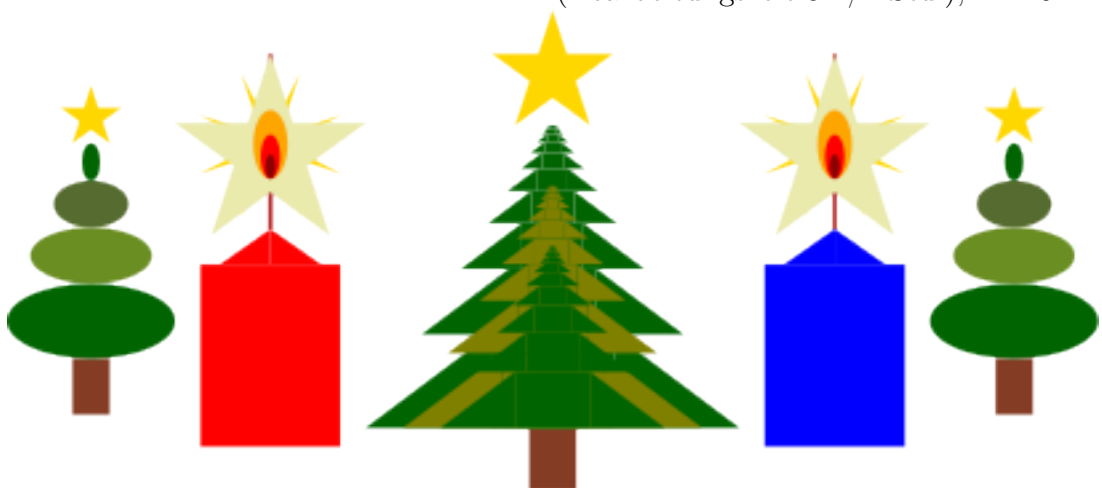
```
(define (odd n)
  (if (= n 0)
      #f
      (even (- n 1)))))
```

Betrachten Sie die angegebenen Funktionen. Geben Sie für jede dieser Funktionen an, welche Arten von Rekursion vorliegen. **Begründen** Sie Ihre Antwort.

	binKo	three	threeX	threeY	threeZ	field	even	odd
end-rekursiv								
linear-rekursiv								
baum-rekursiv								
geschachtelt rekursiv								
direkt rekursiv								
indirekt rekursiv								

2 Ihre Nikolausaufgabe

(Bearbeitungszeit 3 1/2 Std.), 20 Pnkt.



Machen Sie sich mit dem Racket-Modul `"2http/image"` vertraut (siehe DrRacket-Hilfezentrum im Help-Menü).

In dieser Aufgabe ist Ihre Kreativität gefragt! Verwenden Sie grafische Elemente, wie Kreis, Rechteck usw., um ein festliches, weihnachtliches Bild zu komponieren, beispielsweise mit einem geschmückten Tannenbaum, Kerzen, Stapeln von Geschenken usw. Das Programm soll modular aufgebaut sein und gut kommentiert werden.

Ein Beispiel: Die kleinen Bäumchen rechts und links wurden aus olivgrünen Ellipsen, einem gelben Stern und einem braunen Rechteck zusammengesetzt, alles mit `above/align` zentriert übereinandergestapelt:

```
(define baum1 (above/align
  "center"
  ;; der Stern an der Spitze
  (star-polygon 40 5 2 "solid" "gold")
  ;; die Zweige
  (ellipse 20 40 "solid" "darkgreen")
  (ellipse 80 50 "solid" "darkolivegreen")
  (ellipse 130 60 "solid" "olivedrab")
  (ellipse 180 80 "solid" "darkgreen")
  ;; der Stamm
  (rectangle 40 60 "solid" "brown")
))
```

Das Bild soll wiederholte Elemente enthalten, die rekursiv zu erzeugen sind. Wenn ihr Bild auch baumrekursive Strukturen enthält, können Sie sich noch Zusatzpunkte verdienen.

5 Zusatz-
pnkt.

```
(define UFO
  (underlay/align "center"
    "center"
    (circle 10 "solid" "green")
    (rectangle 40 4 "solid" "green")))

(define UFO-BEAM
  (above/align "center"
    (underlay/align "center" "center"
      (circle 10 "solid" "green")
      (rectangle 40 4 "solid" "green"))
    (isosceles-triangle 100 20 "solid" "yellow")))

(define (create-UFO-scene t)
```

```

(let ((height (* (sin (/ t 30)) 50)))
  (underlay/xy (rectangle 500 100 "solid" "white")
               (modulo t 500) height
               (if (< 0 height) UFO UFO-BEAM))))

(animate create-UFO-scene)

```

5 Zusatz-
pnt.

Weitere Zusatzpunkte gibt es, wenn Sie Ihr Weihnachtsbild mit einer Animation “aufpeppen”. Um eine Animation zu erstellen binden Sie das Paket `"2htdp/universe"` ein. Anschließend können Sie eine Szene erstellen, die sich in Abhängigkeit von einem Argument (t) verändert. Das Listing zeigt ein modifiziertes Beispiel aus dem Manual von DrRacket. Die Definition zweier geometrischer Objekte geschieht durch die Funktionen `UFO` und `UFO-BEAM`. Mit Hilfe von `create-UFO-scene` wird eine Animation der Objekte erstellt, die über die Funktion `animate` aufgerufen wird.

Erreichbare Punkte: 30

Erreichbare Zusatzunkte: 10