

Übungen zu Softwareentwicklung III, Funktionale Programmierung

Blatt 3, Woche 4

Leonie Dreschler-Fischer

WS 2017/2018

Ausgabe: Freitag, 10.11.2017,

Abgabe der Lösungen: bis Montag, 20.11.2017, 12:00 Uhr per email bei den Übungsgruppenleitern.

Ziel: Listen und Strings: Die Aufgaben auf diesem Zettel dienen dazu, sich mit den Datenstrukturen von Racket vertraut zu machen. Insbesondere üben Sie die Suche in Assoziationslisten. Außerdem üben Sie die Verwendung von Rekursion, um iterative Schleifen zu programmieren.

Bearbeitungsdauer: Die Bearbeitung sollte insgesamt nicht länger als 4 Stunden dauern.

Homepage:

[http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/
Uebungen_Se_III/Uebungen_Se_III.html](http://kogs-www.informatik.uni-hamburg.de/~dreschle/teaching/Uebungen_Se_III/Uebungen_Se_III.html)

Bitte denken Sie daran, auf den von Ihnen eingereichten Lösungsvorschlägen *Ihren Namen und die Matrikelnummer, den Namen der Übungsgruppenleiterin / des Übungsgruppenleiters und Wochentag und Uhrzeit der Übungsgruppe* anzugeben, damit wir ihre Ausarbeitungen eindeutig zuordnen können.

1 Die internationale Buchstabiertafel

(Bearbeitungszeit 1 1/2 Std.)

Im Seefunk wird die internationale Buchstabiertafel verwendet, wenn Wörter buchstabiert werden müssen. Dabei wird jeder Buchstabe auf einen gut unterscheidbaren Schlüssel abgebildet, nach dem folgenden Schema:

Buchstabe	Schlüssel	Buchstabe	Schlüssel
A	Alpha	T	Tango
B	Bravo	U	Uniform
C	Charlie	V	Viktor
D	Delta	W	Whiskey
E	Echo	X	X-ray
F	Foxtrot	Y	Yankee
G	Golf	Z	Zulu
H	Hotel	0	Nadazero
I	India	1	Unaone
J	Juliet	2	Duotwo
K	Kilo	3	Terrathree
L	Lima	4	Carrefour
M	Mike	5	Pentafive
N	November	6	Soxisix
O	Oscar	7	Setteseven
P	Papa	8	Oktoeight
Q	Quebec	9	Novonine
R	Romeo	,	Decimal
S	Sierra	.	Stop

Peter Panik lernt gerade für die Prüfung zu einem Seesprechfunkzeugnis und hat Schwierigkeiten beim Buchstabieren. Helfen Sie ihm, indem Sie geeignete Racket-Funktionen definieren, die das Üben erleichtern und einen als String gegebenen Text nach der Buchstabiertafel buchstabieren.

1. Eine Datenstruktur für die internationale Buchstabiertafel: Entwerfen und implementieren Sie eine Datenstruktur, in der Sie die Zuordnung der Buchstaben zu ihren Buchstabierschlüsseln speichern können. Erklären und begründen Sie Ihren Entwurf. 4 Pnkt.
2. Eine Codierungsfunktion: Entwerfen und implementieren Sie eine Funktion, die einen Buchstaben (Typ char) mittels der Buchstabiertafel auf seinen Schlüssel abbildet. Nehmen Sie der Einfachheit halber an, dass der Text nur Großbuchstaben enthält. 3 Pnkt.

3. Buchstabieren eines Textes: Definieren Sie eine *rekursive* Funktion, 8 Pnkt.

- die einen Text in Form eines Strings als Eingabe erhält
- und diesen String Zeichen für Zeichen auf eine Liste oder einen String der Buchstabierschlüssel abbildet.

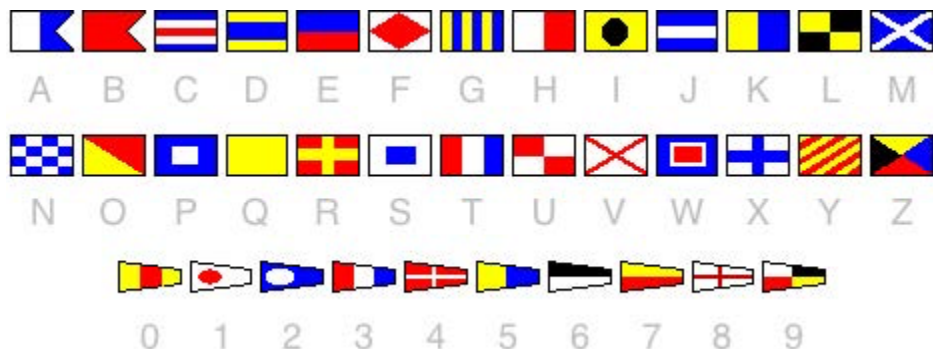
Ein Tipp: Hier ist die Funktion `string->list` nützlich, die eine Zeichenkette in die Liste aller Zeichen wandelt:

`(string->list "SOS")` \rightarrow `'(#\S #\O #\S)`

2 Die internationale Flaggenalphabet

(Bearbeitungszeit 1 1/2 Std.)







Wenn keine Funkverbindung zwischen den Schiffen möglich ist, aber Sichtkontakt besteht, kann auch mit Signalflaggen kommuniziert werden. Dabei wird jedem Buchstaben und jeder Ziffer eine Flagge zugeordnet.



Laden Sie sich das für diese Vorlesung verwendete Modul *SE3-Bib* herunter. Sie finden es unter Stine bei den Übungsblättern. Entpacken Sie die Dateien in das Verzeichnis *Racket-Pfad/collects/se3-bib/*. Stellen Sie bitte zudem sicher, dass in DrRacket bei den "erweiterten Spracheinstellungen" die Checkbox "Compiled-Verzeichnisse bestücken" deaktiviert ist. Nun können Sie das Flaggenmodul mit folgender Anweisung laden:

```
(require se3-bib/flaggen-module)
```

Im Flaggenmodul sind kleine Bilder der Flaggen als Variable vordefiniert: das Symbol „A“ bezeichnet die Flagge „A“ usw. und „Z0“... „Z9“ die Ziffern „0“ bis „9“. Jetzt können Sie Listen von Flaggen bilden:

```
> (list R A C K E T)  $\rightarrow$  (list      )
```

1. **Eine Datenstruktur für das Flaggenalfabet:** Entwerfen und implementieren Sie eine Datenstruktur, in der Sie die Zuordnung der Buchstaben zu ihren Flaggen speichern können. Erklären und begründen Sie Ihren Entwurf. 3 Pnkt.
2. **Eine Codierungsfunktion:** Entwerfen und implementieren Sie eine Funktion, die einen Buchstaben (Typ char) mittels der Flaggentafel auf das Bild der Flagge abbildet. Nehmen Sie der Einfachheit halber an, dass der Text nur Großbuchstaben enthält. 2 Pnkt.
3. **Buchstabieren eines Textes:** Definieren Sie eine *rekursive* Funktion,
 - die einen Text in Form eines Strings als Eingabe erhält
 - und diesen String Zeichen für Zeichen auf eine Liste der Flaggenbilder abbildet. 5 Pnkt.

3 Morse-Codes

(Bearbeitungszeit 1 Std.)

Auch der Morse-Code kann für die Kommunikation von Schiff zu Schiff verwendet werden. Hierbei ist jeder Buchstabe durch eine Tonfolge kodiert, die jeweils nur aus langen Tönen, kurzen Tönen und Pausen besteht. Laden Sie sich für diese Aufgabe bitte die Audiodateien der Morse-Codes (Morse-wav.zip) für die Buchstaben unter <http://bit.ly/morse17> herunter und entpacken Sie sie in einem Verzeichnis ihrer Wahl.

Audio-Dateien können Sie in Racket mit der Funktion `play-sound` abspielen. Diese ist im Paket `Racket/GUI` enthalten, das sie wie folgt einbinden:

```
(require racket/gui/base)
```

Liegt die wav-Datei im gleichen Verzeichnis wie das Racket-Skript, können Sie sie nun so abspielen:

```
> (play-sound "Morse-A.wav" #f)
```

Hierbei verzögert das `“false”` als zweites Argument die weitere Auswertung so lange, bis der Ton vollständig abgespielt ist.

1. **Eine Datenstruktur für die Morse-Codes:** Entwerfen und implementieren Sie eine Datenstruktur, in der Sie die Zuordnung der Buchstaben zu ihren Morse-Codes speichern können. Erklären und begründen Sie Ihren Entwurf. 3 Pnkt.

2. **Eine Codierungsfunktion:** Entwerfen und implementieren Sie eine Funktion, die einen Buchstaben (Typ `char`) mittels der Morse-Codes auf eine Soundausgabe des entsprechenden Codes abbildet. Nehmen Sie der Einfachheit halber an, dass der Text nur Großbuchstaben enthält. 2 Pkt.
3. **Morsen eines Textes:** Definieren Sie eine *rekursive* Funktion, 5 Zusatzpnt.
- die einen Text in Form eines Strings als Eingabe erhält
 - und diesen String Zeichen für Zeichen auf deren Morse-Codes abbildet.

Erreichbare Punkte: 30

Erreichbare Zusatzunkte: 5