10. Aufgabenblatt zur Vorlesung

Konzepte der Programmierung

WiSe 2024/25

Wolfgang Mulzer

Abgabe am Freitag, den 17. Januar 2025, 12 Uhr

Aufgabe 1 Scala I

10 Punkte

(a) Um die gefühlte Temperatur bei Wind zu bestimmen, kann man die folgende Formel zur Bestimmung der Windchill-Temperatur t_{chill} abhängig von der Temperatur t und der Windgeschwindigkeit v verwenden:

$$t_{\text{chill}} = 13.12 + 0.6215 \cdot t + (0.3965 \cdot t - 11.37) \cdot v^{0.16}$$

Implementieren Sie eine Funktion in Scala, welche die Windchill-Temperatur berechnet.

(b) Implementieren Sie eine *endrekursive* Funktion, welche eine natürliche Zahl n erhält und eine Zahl zurückgibt, deren umgekehrte Dezimaldarstellung der Darstellung von n entspricht. Veranschaulichen Sie anschließend anhand der Eingabe $n = 47142$, wie die Funktion schrittweise abläuft.

(c) Recherchieren Sie Vor- und Nachteile von Schleifen, Rekursion und Endrekursion und vergleichen Sie diese miteinander. Geben Sie außerdem passende Python- und/oder Scala-Beispiele, welche die Argumente verdeutlichen (sofern möglich).

(d) Recherchieren Sie den Begriff *referentielle Transparenz* und diskutieren Sie Vor- und Nachteile. Geben Sie, wenn möglich, passende Beispiele in Scala und/oder Python.

Aufgabe 2 Scala II

10 Punkte

In Scala ist es möglich, wie folgt eigene Typen zu definieren:

```
type Uhrzeit = (Int, Int, Int)
val z : Uhrzeit = (14, 15, 0)   // Startzeit der Vorlesung
```

Hier wird der Datentyp `Uhrzeit` definiert, der aus drei ganzen Zahlen besteht: Stunden, Minuten und Sekunden. Implementieren Sie die folgenden Funktionen:

- `istUhrzeit` überprüft, ob ein gegebenes Tripel aus ganzen Zahlen tatsächlich eine gültige Uhrzeit ist.
- `tick` erhält eine Uhrzeit und liefert die Uhrzeit eine Sekunde später. (Beispiel: `tick((22, 59, 59))` liefert `(23, 0, 0)`.)
- `kcit` ist die Umkehrfunktion von `tick`.

- Die Funktionen `addSekunden`, `addMinuten` sowie `addStunden` erhalten eine gültige Uhrzeit und eine ganze Zahl und addieren diese ganze Zahl auf die Sekunden, Minuten bzw. Sekunden.

Aufgabe 3 MapFold *10 Punkte*

(a) Implementieren Sie eine polymorphe Funktion `replace`, welche zwei Parameter a und b sowie eine Liste `xs` erhält und in `xs` alle Vorkommen von a durch b ersetzt. Implementieren Sie die Funktion einmal mit `map` und einmal mit expliziter Rekursion (ohne `map`).

(b) Implementieren Sie eine polymorphe Funktion `isSorted`, welche eine Liste mit Elementen und einen Vergleichsoperator `comp: (A, A) => Boolean` erhält und überprüft, ob die Elemente gemäß des Vergleichsoperators sortiert sind.

Zu Beispiel:

```
isSorted((_:Int) > (_:Int), List(4, 3, 2, 1)) == true
```

(c) Implementieren Sie mit Hilfe der Funktionen höherer Ordnung eine Funktion `takeNumbers`, welche die ersten n Elemente einer Liste von ganzen Zahlen liefert, deren Quersumme jeweils einen gegebenen Wert überschreiten.

(d) Implementieren Sie die `reverse`-Funktion mit einer Faltung.