

Übungen zu Einführung in die Informatik I

Aufgabe 9 **Funktionensdefinitionen in OCaml**

Diese Aufgabe soll die verschiedenen Möglichkeiten wie in OCaml Funktionen definiert werden verdeutlichen.

- a) **Definition von Funktionen:** In OCaml werden Funktionen mit dem Keyword `fun` definiert. Schreiben Sie eine Funktion die den Eingabewert um 1 erhöht. Testen Sie diese Funktion
- b) **Anonyme Funktionen mit mehreren Parametern:** Schreiben Sie eine Funktion die zwei Zahlen addiert. Verwenden Sie hierfür nur einstellige anonyme Funktionen `fun`.
- c) **function vs. fun** Eine weitere Möglichkeit in OCaml Funktionen zu definieren besteht mit dem Schlüsselwort `function`. Erklären Sie kurz den Unterschied zwischen der Verwendung von `function` und `fun`.

Aufgabe 10 **Arbeiten mit Modulen**

Zur besseren Übersicht und um gleichnamige Operationen für verschiedenen Datentypen zu unterscheiden verwendet OCaml sogenannte *Module*. Ein Beispiel ist: `Random.float`, das die Funktion `float` aus dem Modul `Random` aufruft.

In dieser Aufgabe verwenden wir Funktionen aus dem Modul `List` zur Operation auf Listen von Datentypen.

- a) Definieren und belegen Sie in OCaml die beiden Zeichensequenzen `vorname` und `nachname` jeweils mit Ihrem Vor- bzw. Nachnamen. Verknüpfen Sie anschließend diese beiden Zeichensequenzen zu einer einzigen Zeichensequenz `name` unter Verwendung des Konkatenationsoperator `@` und einem Komma sowie einem Leerzeichen zwischen den beiden Namensteilen.
- b) Verwenden Sie die Funktion `length` aus dem Modul `List` um die Länge des ganzen Namens festzustellen.
- c) Revertieren Sie Ihren (oben definierten) Namen mit Hilfe von OCaml durch Zerlegung und unter ausschließlicher Verwendung der Funktionen `hd` und `tl` aus dem Modul `List` und des Operators `::` ('Prepend').
Hinweis: Zur Vereinfachung der Schreibweise bietet es sich an zuerst zwei Funktionen `head` und `tail` zu definieren, welche genau die entsprechenden Funktionen aus dem Modul `List` aufrufen.
- d) Definieren Sie die folgende *Cast*-Funktion `list_of_string` die mit Hilfe von Funktionen aus den Modulen `String` und `Stream` aus einem String eine Liste von Einzelzeichen macht.

```
let list_of_string s =  
  Stream.npeek (String.length s) (Stream.of_string s);;
```

Testen Sie diese Funktion mit ein paar Eingabestrings.

- e) Definieren Sie nun eine rekursive Funktion *revert*, die eine beliebige Liste umkehrt. Testen Sie diese Funktion für Ihren oben definierten Namen sowie (unter Verwendung der Cast-Funktion aus Teilaufgabe c) für den String `reliefpfeiler`.

Aufgabe 11 **Komplexe Zahlen**

Die komplexen Zahlen erweitern den Zahlenbereich der reellen Zahlen derart, dass auch Wurzeln negativer Zahlen berechnet werden können. Dies gelingt durch Einführung einer neuen Zahl i (imaginäre Einheit) als Lösung der Gleichung. Komplexe Zahlen werden meist in der Form $x + y \cdot i$ dargestellt, wobei x und y reelle Zahlen sind und i die imaginäre Einheit ist. Auf die so dargestellten komplexen Zahlen lassen sich die üblichen Rechenregeln für reelle Zahlen anwenden, wobei i^2 stets durch -1 ersetzt werden kann und umgekehrt.

- a) Definieren Sie den Datentyp `complex` als Tupel von zwei reellen Zahlen in OCaml.
- b) Schreiben Sie eine Funktion `add` die zwei komplexe Zahlen addiert.
- c) Schreiben Sie eine Infix-Funktion `++` die zwei komplexe Zahlen addiert.
- d) Erweitern Sie die Definition von `complex` dahingehend, dass eine komplexe Zahl entweder in der normalen Form oder in Polarform darstellen lässt. Die Polarform wird in der Form $r \cdot (\cos \psi + i \cdot \sin \psi)$ angegeben. Schreiben Sie eine Funktion `polar` die als Parameter eine komplexe Zahl c erwartet und diese in der Polarform zurückgibt. Schreiben Sie außerdem eine Funktion die eine komplexe Zahl c in der Form $x + y \cdot i$ zurückgibt. (Hinweis: Benutzen Sie für die Umrechnung zwischen den beiden Darstellungen folgende Formeln:
$$r = \sqrt{x^2 + y^2}; \quad \psi = \arccos \frac{x}{r} \quad \text{für } y \geq 0 \quad \text{und} \quad \psi = \arccos \frac{-x}{r} - \pi \quad \text{für } y \leq 0;$$
$$x = r \cdot \cos \psi; \quad y = r \cdot \sin \psi; \quad \pi = 2 \cdot \arcsin(1)$$
- e) Schreiben sie die Infix-Funktion `++` um, so dass diese zwei Komplexe Zahlen in beliebigen Format addiert und in der Form $x + y \cdot i$ zurückgibt.

Aufgabe 12 (H) **Kalender**

(10 Punkte)

Ziel dieser Aufgabe ist es für einen bestimmten Monat des Jahres 2006 den Monatskalender zu berechnen, d.h. den einzelnen Datumswerten Tage zu zuordnen. Der Kalender soll als Liste von Tupeln ausgegeben werden, wobei jedes Tupel den Tag und das zugehörige Datum enthält. Folgende Liste gibt eine Möglichkeit eines entsprechenden Algorithmus (natürlich können Sie sich auch einen anderen Algorithmus überlegen)

- Definieren Sie einen Datentyp für die Wochentage und die Monate
- Definieren Sie eine Funktion, die den Monaten die Monatslänge zuordnet (Vorlesung!).
- Der 1. Januar des Jahres 2006 ist ein Sonntag.
- Definieren Sie eine Funktion, die zu einem Wochentag (Monat), den nächsten Wochentag (Monat) ermittelt.
- Definieren Sie eine Funktion, die ermittelt, wie viele Tage seit 1.1.2006 bis zu einem bestimmten Datum des Jahres 2006 vergangen sind.
- Definieren Sie nun eine Funktion, die aus der vorhergehenden den Wochentag des Datum bestimmt.
- Erstellen Sie den restlichen Kalender!