

# Übungsblatt 3

## Programmieren 1 – WiSe 22/23

Prof. Dr. Michael Rohs, Jan Feuchter, M.Sc., Tim Dünke, M.Sc

Alle Übungen (bis auf die erste) müssen in Zweiergruppen bearbeitet werden. Beide Gruppenmitglieder müssen die Lösung der Zweiergruppe einzeln abgeben. Die Namen beider Gruppenmitglieder müssen sowohl in der PDF Abgabe, als auch als Kommentar in jeglichen Quelltextabgaben genannt werden. Plagiate führen zum Ausschluss von der Veranstaltung.

Abgabe bis Donnerstag den 03.11. um 23:59 Uhr über <https://assignments.hci.uni-hannover.de/WiSe2022/Prog1>. Die Abgabe muss aus einer einzelnen Zip-Datei bestehen, die den Quellcode, ein PDF für Freitextaufgaben und alle weiteren nötigen Dateien (z.B. Eingabedaten oder Makefiles) enthält. Lösen Sie Umlaute in Dateinamen auf.

### Aufgabe 1: Skript

Das Skript unter <https://postfix.hci.uni-hannover.de/files/prog1script-postfix/script.html> beschreibt verschiedene Vorgehensweisen bei der Lösung von Programmieraufgaben.

- Lesen Sie Kapitel 7 (Recipe for Compound Data) und beantworten Sie folgende Frage in eigenen Worten: Was ist die Aufgabe von constructor-Funktionen und was die von accessor-Funktionen?
- Lesen Sie Kapitel 8 (Recipe for Variant Data) und beantworten Sie folgende Frage in eigenen Worten: Welchen Vorteil bietet `datadef` im Zusammenhang mit Variant Data?
- Lesen Sie Kapitel 9 (Recipe for Self-Referential Data) und beantworten Sie folgende Frage in eigenen Worten: Was versteht man unter Induktionshypothese im Zusammenhang mit rekursiven Funktionen?
- Was war Ihnen beim Lesen der Kapitel 7 bis 9 unklar? Wenn nichts unklar war, welcher Aspekt war für Sie am interessantesten?

## Aufgabe 2: Tempomat

Ein Tempomat hat die Aufgabe, ein gegebenes Tempo möglichst genau zu halten. Entwickeln Sie eine Funktion zur Regelung eines Tempomats, welches abhängig vom aktuellen Tempo und dem Zieltempo entweder bremst, nichts tut, beschleunigt oder eine Notbremsung auslöst.

- Für den Fall, dass das aktuelle Tempo oder das Zieltempo  $< 0$  km/h ist, soll in jedem Fall eine Notbremsung ausgelöst werden.
- Für den Fall, dass das aktuelle Tempo größer als 130 km/h ist, soll der Tempomat bremsen.
- Der Tempomat soll bremsen, falls das aktuelle Tempo mehr als 2% größer ist als Zieltempo.
- Der Tempomat soll beschleunigen, falls das aktuelle Tempo mehr als 2% kleiner ist als das Zieltempo.
- Andernfalls (aktuelles Tempo  $\pm 2\%$  vom Zieltempo) soll der Tempomat nichts tun.

Führen Sie die im Skript unter Recipe for Intervals beschriebenen Schritte durch. Verwenden Sie die Template-Datei `cruise-control.pf`. Bearbeiten Sie die mit `todo` markierten Stellen. Geben Sie mindestens 6 sinnvolle Beispiele (Eingaben und erwartete Resultate) in Form von Testfällen an.

## Aufgabe 3: Schachpositionen

In dieser Aufgabe sollen Funktionen implementiert werden, um verschiedene Darstellungen von Figurpositionen auf einem Schachbrett ineinander zu konvertieren. Verwenden Sie als Template- Datei `chess.pf`.

- Implementieren Sie die Funktion **pos-to-point**: (`pos :Str -> :Arr`), die eine Positionsangabe auf dem Schachbrett der Form "A1" in einen 2D-Punkt der Form `[1, 1]` konvertiert. Auf dem Schachbrett sind die Spalten von "A" bis "H" und die Zeilen von 1 bis 8 benannt. Ergänzen Sie in der Testfunktion zwei weitere Beispiele. Gehen Sie davon aus, dass die Positionsangabe immer zwei Zeichen lang ist. Dies braucht also nicht überprüft zu werden.
- Implementieren Sie die Funktion **point-valid**: (`p :Arr -> :Bool`), die prüft, ob es sich bei der Eingabe um einen gültigen 2D-Punkt auf dem Schachbrett handelt. Ergänzen Sie in der Testfunktion zwei weitere Beispiele.
- Implementieren Sie die Funktion **point-to-pos**: (`a :Arr -> :Str`), die einen 2D-Punkt der Form `[2, 3]` in eine Schachbrettposition der Form "B3" konvertiert. Ergänzen Sie in der Testfunktion zwei weitere Beispiele.
- Erläutern Sie kurz die Funktionsweise der Funktion **knight-next-positions**. Erläutern Sie insbesondere die letzte Zeile.

## Aufgabe 4: Auktionshaus

Ein Prototyp für ein Auktionshaus soll in Postfix implementiert werden (`auction.pf`). Auktionen werden als **Objekt** gespeichert, welches eine von drei Alternativen darstellt.

Eine Auktion kann vom Typ `:Int` sein, wenn Sie noch nicht begonnen hat. Dann entspricht der Wert dem Startzeitpunkt der Auktion (in Sekunden).

Weiter kann sie vom Typ `:Arr` sein, dann enthält das Array das aktuelle Gebot, den aktuellen Bieter und den Endzeitpunkt (in Sekunden).

Wenn die Auktion beendet ist, so ist die Auktion vom Typ `:Str` und enthält den Namen des Höchstbietenden.

Beispiele dazu sind im Code genannt. Vervollständigen Sie die folgenden Funktionen.

- a) Implementieren Sie die Funktion `bid`: (auction `:Obj`, bidder `:Str`, offer `:Num` -> `:Obj`), um das Bieten auf eine Auktion zu ermöglichen.

Ein Gebot soll in folgenden Fällen nicht möglich sein:

- Wenn die Auktion noch nicht begonnen hat oder beendet ist
- Wenn das neue Gebot kleiner oder gleich dem alten Gebot ist
- Wenn der gleiche Bieter versucht sein Gebot zu erhöhen
- Wenn der Bieter der leere String ist `""`

In den oben genannten Fällen soll das alte Gebot erhalten bleiben. Ansonsten wird das Gebot aktualisiert (die Endzeit bleibt immer gleich). Schauen Sie sich dazu auch die Testfälle an.

- b) Implementieren Sie die Funktion `update-auction`: (auction `:Obj`, time `:Int` -> `:Obj`). Diese soll den Zustand einer Auktion anhand einer übergebenen Zeit aktualisieren.

Wenn die übergebene Zeit (Parameter time) gleich oder größer als die Startzeit ist, soll die Auktion initialisiert und gestartet werden. Der Endzeitpunkt entspricht der Startzeit plus der Auktionszeit. Ab dem Endzeitpunkt soll die Auktion beendet werden. **Beachten Sie die Konstanten am Anfang des Codes.**

- c) (OPTIONAL) Wenn alle Testfälle fehlerfrei funktionieren kommentieren Sie die letzte Zeile aus und starten Sie das Programm:

```
#todo: remove comment to animate  
#animate
```

Bieten Sie ein paar Mal in dem Sie auf den Button "Bieten!" klicken und prüfen Sie, ob Ihre Implementierung funktioniert. Die Eingabe des Namens und des Gebots sind leider aktuell nur über die Eingabe der Postfix IDE möglich. Bitte Beenden Sie die Applikation immer über den roten Stopp Button in der IDE.