



Leibniz  
Universität  
Hannover

# ASSIGNMENT 03

Programmieren I  
05.11.2024

NORWIN BERTRAM WIESECKE

10057811

TIMON WELLHAUSEN

10041137

## AUFGABE 1:

- a) 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.

*Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_  
Timon\_Wellhausen\_10041137\_cruise-control.pf“.*

## AUFGABE 2:

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.

- a) 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.

*Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_  
Timon\_Wellhausen\_10041137\_chess.pf“.*

- b) 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.

*Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_  
Timon\_Wellhausen\_10041137\_chess.pf“.*

- c) 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.

*Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_  
Timon\_Wellhausen\_10041137\_chess.pf“.*

- d) Erläutern Sie kurz die Funktionsweise der Funktion `knight-next-positions`. Erläutern Sie insbesondere die letzte Zeile.

Die Funktion erhält als Input einen String, der über die „pos-to-point“ Funktion in einen Array mit zwei Integer-Werten umgewandelt wird. Von diesem Array wird das erste Element als „px“ und das zweite Element als „py“ abgespeichert. Aus diesen beiden Koordinaten werden im nächsten Schritt sämtliche potenzielle Endkoordinaten berechnet. Der Springer bewegt sich immer zwei Felder in eine Richtung und dann ein Feld seitwärts. Dieser Array wird als „a“ abgespeichert. Für jedes Element in „a“ wird als nächstes geprüft, ob der entsprechende Wert eine valide Position auf dem Schachbrett darstellt. Ist dies nicht der Fall, wird er gelöscht ansonsten wird der Zahlenwert über die Funktion „point-to-pos“ in eine String-Form zurückgebracht. Der entstehende Array wird am Ende alphabetisch sortiert.

### AUFGABE 3:

In einer blockbasierten Simulationswelt soll berechnet werden, wie viel Wasser zwischen zwei Wänden eingeschlossen werden kann, ohne dass es über die Wände hinausläuft. Die Simulationswelt wird als Array (jeweils erster Parameter) dargestellt, wobei jeder Eintrag die Höhe der Wand an dieser Position angibt. Das heißt, falls an dieser Position keine Wand ist, so ist der Eintrag im Array 0. Verwenden Sie `water_sim.py` als Template-Datei und implementieren Sie dazu folgende Funktionen:

- a) `set-two-walls`, welche an den als Parameter übergebenen Stellen Wände erzeugt. Die Wandhöhe wird als Parameter übergeben (`wall_one_height` und `wall_two_height`).

Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_Timon\_Wellhausen\_10041137\_water\_sim.py“.

- b) `increase-height`, welche eine der beiden Wände um einen bestimmten Wert erhöht. Der zweite Parameter gibt an, um welche Wand es sich handelt (0 für die erste bzw. 1 für die zweite), der dritte Parameter die zu ergänzende Höhe.

Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_Timon\_Wellhausen\_10041137\_water\_sim.py“.

- c) `calculate-water`, die die Menge an Wasser (in Blöcken) bestimmt, welche zwischen den Wänden ausgekippt werden kann, ohne dass das Wasser über die Wände läuft.

Siehe „Assignment\_03\_Norwin\_Bertram\_Wiesecke\_10057811\_Timon\_Wellhausen\_10041137\_water\_sim.py“.