

### Aufgabe 1:

a) Constructor Functions nehmen Daten als Eingabe und konstruieren ein kompaktes Object welches all diese Daten enthält

Accersor Functions nehmen das kompakte Objekt als Eingabe und sind in der Lage die abgespeicherten Werte des Objekt zurückzugeben

b) Die datadef function erspart mir Accersor Funktionen und Constructor Funktionen zu schreiben und erstellt mögliche Funktionen um zu

gucken ob sie den gleichen übergeordneten Typ haben und ob sie der gleiche Typ sind und so wird viel Schreibarbeit erspart und es

fällt mir einfacher Funktionen für übergeordnete Typen zu schreiben, weil ich durch die Bereitgestellten Funktionen eine einfache Fallunterscheidung

machen kann und somit Code spare und damit mich nicht wiederhole

c) Die Induktionshypothese ist das Aufrufen der Funktion in sich selbst mit der Annahme, dass diese genau so funktioniert wie vorgestellt.

d) Aus den Kapitel war datadef sehr interessant und ich hätte nicht gedacht das Postfix es erlaubt so komplexen Code zu schreiben.

### Aufgabe 2:

Siehe Datei: aufg\_2\_postfix.pf

### Aufgabe 3:

a, b, c: Siehe Datei: chess.pf

d) # remove invalid positions (the ones not on the board)

# iterates over a, dups value of each element, so every element is on the stack twice

# then checks if point is valid, get bool as return, negates value, if point is not valid pop value from stack (stack empty), if point is valid add position to list (stack is empty again), then sorts array