

# Tinpro01-7 Practicumopdracht 2

W. Oele

22 april 2021

## Inleiding

In deze opdracht ga je werken met geparameteriseerde datatypen. Deze datatypen zijn enigszins vergelijkbaar met generics in Java.

### Opdracht 1

Schrijf het datatype Boek. Dit datatype bevat drie velden:

- Prijs: De prijs van het boek.
- Titel: De titel van het boek.
- Auteur: De schrijver van het boek.

Gebruik type synoniemen om de code leesbaar te houden.

### Opdracht 2

Leidt je Boek af van de typeclasses Eq en Ord. We willen boeken onderling kunnen vergelijken (Eq) en op volgorde kunnen zetten (Ord). Voorwaarden:

- Twee boeken zijn aan elkaar gelijk als alle velden gelijk zijn.
- Boeken dienen alfabetisch gesorteerd te kunnen worden op de titel van het boek. Prijs en auteur interesseren ons niet.

### Opdracht 3

Maak een lijst van tenminste vijf boeken. Test of je implementatie van de Eq en Ord typeclasses correct werkt door:

- enkele boeken met elkaar te vergelijken.
- de lijst met boeken te sorteren middels het gebruik van de sort functie. Deze functie zit in de library Data.List.

### Opdracht 4

Schrijf het datatype Box a. Dit datatype is algebraïsch. Een box:

- is Leeg.
- bevat iets, waarvan we van te voren niet weten wat het is.

### Opdracht 5

Schrijf een functie die alle boeken in een gegeven lijst in een `Box` inpakt. Schrijf ook een functie die het omgekeerde doet en, so to speak, alle boeken weer uitpakt.

### Opdracht 6

Schrijf het datatype `Zak`. Ook dit datatype is algebraïsch. Verder gelden dezelfde voorwaarden als bij het `Box` type: Een zak bevat *iets* of is `Leeg`

### Opdracht 7

Leidt zowel het datatype `Box` als het datatype `Zak` af van de `Functor` typeclass en implementeer de functie `fmap` voor beide datatypen.

### Opdracht 8

Schrijf enkele oneliners:

- Elk boek in de boekenlijst wordt ingepakt in een `Zak` en al deze zakken worden, per stuk, ingepakt in een `Box`
- Maak een willekeurige lijst getallen. Stop elk getal in een `Box`. We hebben nu een lijst met boxes, waarbij in elke `Box` een getal zit.

### Opdracht 9

Schrijf het datatype `List`. Dit datatype is geparameteriseerd en definieert recursief een list.

- Leidt de `List` af van de `Functor` typeclass.
- Vul de `List` met dozen met getallen. Gebruik daarvoor de `foldr` of `foldl` functie.
- Gebruik de functionaliteit van `Functor` om een met `Box` gevulde list te vervangen door een met `Zak` gevulde list.

### Opdracht 10 (facultatief)

Schrijf op soortgelijke wijze als in opdracht 7 een binaire boom en demonstreer de werking.