|  |
| --- |
| Functioneel Paradigma\_  Algoritmes, Paradigma’s en Programmeertalen |

Mike Doornenbal - 1628421

[kies een datum]

INHOUDSOPGAVE

[INLEIDING 2](#_Toc18073183)

[1 [hoofdstuktitel] 2](#_Toc18073184)

[1.1 [Paragraaftitel] 2](#_Toc18073185)

INLEIDING

Voor het vak APP (Algoritmes, Paradigma’s en Programmeertalen), hebben wij de opdracht gekregen een challenge te ontwerpen en implementeren in een zuiver functionele programmeertaal. Ik heb ervoor gekozen om RLE-Compressie (Run-Length Encoding) te programmeren in Haskell. Naast deze challenge, wordt ook verwacht dat wij een rapport inleveren waarin onze bevindingen beschreven staan, waaronder het initiële onderzoek naar de programmeertaal, een samenvatting over de implementatie en een reflectie over wat we geleerd hebben van deze opdracht.

Zoals eerder vermeld heb ik ervoor gekozen om deze challenge uit te werken in Haskell. Haskell is een puur functionele programmeertaal die sinds 1990 wordt gebruikt. Ik heb deze taal gekozen omdat het één van de programmeertalen is die als optie wordt gegeven bij de beschrijving van de opdracht. Daarnaast hebben veel klasgenoten ook voor Haskell gekozen, wat maakt dat ik ook hen om hulp kan vragen wanneer ik ergens niet uit kom.

# Onderzoek

Voordat je goed kan werken met functionele programmeertalen, is het zaak om eerst een aantal kernconcepten te begrijpen. Zo is één belangrijk concept het idee dat een functionele programmeertaal zuiver is. Dit houdt in dat een functie die een bepaalde input krijgt altijd dezelfde output genereert.

Naast zuiverheid van functies, bestaan er ook andere concepten. De concepten die relevant zijn gebleken voor de programmeerchallenge staan hieronder uitgelegd.

## Zuiverheid

## Lazy Evaluation

# Challenge

Voor deze opdracht heb ik gekozen om RLE-compressie te implementeren. Van de vooraf gegeven opties, leken degene die met compressie te maken hebben de meest interessante. Uiteindelijk heb ik RLE-compressie gekozen over LZW-compressie, omdat ik (na het bestuderen van de Wikipedia-pagina’s voor beide compressie-algoritmen) bij RLE-compressie al snel een idee kreeg bij hoe het algoritme comprimeert.

# Implementatie

# Reflectie

# Conclusie

Samenvatting van de belangrijkste (leer)punten

# Bronvermelding

Bronvermelding

<https://www.seas.upenn.edu/~cis1940/spring13/lectures.html>

