# Toets PO\_Paradigma

versie 1 2021-22, augustus 2021

**Inleiding**

Dit is een individuele toets, waarbij je een rapport moet schrijven over een klein onderzoek naar een functionele programmeertaal die voor jou nieuw is. Je geeft daarmee aan dat je zelfstandig onderzoek kunt doen, dat je weet wat het functionele paradigma inhoudt en ermee kunt werken, en dat je jezelf een programmeertaal eigen kunt maken, ook in een ander paradigma dan imperatief of OO. Ook oefen je nog een keer verslagleggen. Dit zijn allemaal vaardigheden die je van pas komen bij het afstuderen en als beroepsbeoefenaar in het SD-domein.

Formeel voldoe je aan het eind aan ASD APP-8: *De student onderzoekt zelfstandig een overwegend functionele programmeertaal en programmeert daarin een algoritme uit.*

**Opdracht**

Kies een voor jou onbekende functionele programmeertaal. Leer jezelf die taal, zodanig dat je enige concepten van functioneel programmeren leert. Kies ook een Challenge, een programmeeropdracht die de functionele concepten van je taal doet uitkomen, en programmeer die Challenge uit. Schrijf een rapport over de functionele concepten van je gekozen taal, de uitwerking van de Challenge, en hoe daarin een aantal van die functionele concepten terugkomen.

Lever het rapport op met de source-code van je Challenge als bijlage *in zip-formaat* (geen rar).

**Stappenplan**

Je kunt deze opdracht op twee manieren aanpakken: beginnen bij een (functionele) programmeertaal en daar een Challenge bij zoeken, of beginnen bij een Challenge, en daar een programmeertaal bij zoeken. Begin daar in week 1 mee, dan is het nog rustig. Probeer ook rond week 6 af te ronden, de laatste weken van dit blok heb je het druk met alle drie vakken APP, SWA, en Skills.

Zorg dat je de combinatie Taal en Challenge in de eerste twee weken vertaalt naar een (klein) onderzoeksplan, en check dat met je docent. Een mogelijk tijdspad:

* *Week 1 en 2*:
  + selecteer een Functionele Programmeertaal en een Challenge;
  + schrijf een kort onderzoeksplan (inleiding, vraagstelling met Challenge en het verband met functionele concepten, deelvragen en hoe je die beantwoorden wilt *met referentie naar de methodenkaart*, tijdpad, literatuur waarmee je begint);
  + check je onderzoeksplan en je Challenge met je docent –**deze check is verplicht.**
* *Week 3 en 4*: maak jezelf vertrouwd met de gekozen programmeertaal en de functionele concepten.
* *Week 5 en 6*: werk je Challenge zelfstandig uit en schrijf het onderzoeksverslag:
  + de Challenge en het verslag moeten **eigen werk** zijn, geen vonsten op internet;
  + refereer in het verslag naar de functionele concepten die je in de code van je Challenge gebruikt.
* *Week 7 en 8* (check voor de exacte deadline iSAS): uitloop – je zult het met andere zaken druk hebben, dus vertrouw hier niet teveel op.

**Suggesties voor Functionele Talen**

Hoewel het lastig te controleren is, is het expliciet de bedoeling dat je een programmeertaal kiest die je niet kent. Het leren van een taal die je al kent oefent niets.

Veel functionele talen hebben ook links met andere paradigma’s. Dat is geen probleem, zolang je maar de nadruk legt op de functionele aspecten. Talen die ‘ook OO’ zijn kun je beter vermijden, dat maakt het te gemakkelijk om op ‘oude gewoonten’ terug te vallen. Een eigen voorstel is ook mogelijk, mits je het op tijd aanvraagt en je docent het goedkeurt.

* Taal: **Erlang**
  + Paradigma: functioneel/proces
  + Wat bijzonderheden:
    - focus op robuustheid
    - hotswapping code
    - oorspronkelijk voor embedded systemen
* Taal: **Clojure**
  + Paradigma: functioneel/symbolisch
  + Wat bijzonderheden:
    - JVM platform / Java integratie
    - Lisp familie
    - Meta-programming
* Taal: **Haskell**
  + Paradigma: functioneel (puur)
  + Wat bijzonderheden:
    - geen functies met zijeffecten
    - Lazy Evaluation (o.a. oneindige structuren)
* Taal: **Racket**
  + Paradigma: Multi, o.a. functioneel, logisch
  + Wat bijzonderheden:
    - gebaseerd op de oudste functionele programmeertaal Lisp
    - ook meta-programmeren mogelijk
    - platform gratis en open-source
* Taal: **eigen voorstel**, mits (deels) functioneel
  + *Check je voorstel bij je docent*, niet later dan week 3
  + Als het een leuk idee is kan het in deze lijst worden opgenomen!

**Suggesties voor Challenges**

Juist hier is het leuk om een eigen voorstel te doen. Check dit op tijd, liefst in week 2, maar uiterlijk week 3, bij je docent. Als het een leuk en goed voorstel is, nemen we dat graag op in onderstaande lijst.

Een paar suggesties voor de mensen die nog geen idee hebben:

* LZW compressie in Clojure
* JSON parser in Haskell
* Boom van Pythagoras in Racket

**Toets-eisen**

[PO\_Paradigma 1] Onderzoek voldoet aan SD-criteria voor onderzoeken (methodenkaart).

[PO\_Paradigma 2] Onderzoekt de concepten van een functionele programmeertaal.

[PO\_Paradigma 3] Programmeert een algoritme of programma uit in een overwegend functionele programmeertaal.

[PO\_Paradigma 4] Relateert functionele concepten aan specifieke onderdelen van zijn/haar uitgeprogrammeerde algoritme.

Uiteraard moet je ook voldoen aan de AIM contrôlekaart.

Je voldoet daarmee mede aan de volgende SD-Eindkwalificaties: SD-4, SD-5, SD-7, SD-8.

**Beoordeling**: Elk onderdeel wordt beoordeeld met een O, V, of G. *Nieuw : geen vink maar een eindcijfer*.

Het eindcijfer is 6+aantal(G)-aantal(O), met maximaal een 5 als je a) meer dan één O hebt, of b) als je niet hebt voldaan aan de AIM contrôlekaart. Eén O kun je dus compenseren, meer niet.