

# Rekursie

Steven Bronsveld

February 17, 2019

## 1 Gegevens

Uiterlijke inleverdatum: **Datum 1**

### 1.1 Links

- [Github.com/StevenBrons](https://github.com/StevenBrons)
- <https://natureofcode.com/book>
- <http://hello.processing.org/editor/>

## 2 Leerdoelen

- Methods met `return` van type `int`
- Printen naar het console door middel van `System.out.println()`
- Het recursief aanroepen van methods
- Het recursief tekenen van simpele fractels

## 3 Recursie

### 3.1 Recursieve functies

Recursieve functies zijn functies die een (makkelijkere) versie van zichzelf gebruiken voor het bereken van een antwoord. Kijk bijvoorbeeld naar:

$$f(g, n) = \begin{cases} 1 & n = 0 \\ g * f(n-1) & n > 0 \end{cases}$$

Deze functie geeft  $f(g, n) = g^n$ . Als we dit uitschrijven krijgen we:  $f(3, 4) = 3 * f(3) = 3 * 3 * f(2) = 3 * 3 * 3 * f(1) = 3 * 3 * 3 * 3 * f(0) = 3 * 3 * 3 * 3 * 1 = 81 = 3^4$

$$g(n) = \begin{cases} 1 & n = 0 \\ n * g(n-1) & n > 0 \end{cases}$$

Schrijf de uitwerking van  $g(5)$  helemaal uit. Weet je ook welke functie  $g$  is?

$$h(n) = \begin{cases} 1 & n = 0 \\ 1 & n = 1 \\ h(n-1) + h(n-2) & n > 1 \end{cases}$$

Schrijf de uitwerking van  $h(4)$  helemaal uit. Weet je ook welke functie  $h$  is?

### 3.2 Factorial

Maak een recursieve functie die de *factorial* van een getal berekent. (Zie `f` van de vorige opdracht).

```
1 int factorial(int n) {  
2 }
```

Roep de functie aan in `setup` en print het resultaat.

### 3.3 Codeer Koch's Curve

Maak een functie

```
1 void KochCurve(int n, int x, int y, int l) {  
2 }
```

die Koch's Curve `n` diep, beginnend op `x,y` en eindigend op `x+l,y`.

### 3.4 Codeer een Binary Tree

Maak een method

```
1 void binaryTree(int n, int x, int y) {  
2 }
```

Die een **binary tree** tekent van **n** diep. Op de coördinaten **x**, **y**.

### 3.5 Codeer een Sierpinski Triangle

### 3.6 [Bonus] Koch's Snowflake

### 3.7 [Bonus] Hilbert's Curve

## 4 Inleveren

Als je klaar bent met de hele opdracht kun je deze naar je *repository* pushen.