

## SPD MODULE 2: METHODEN, PARAMETERS, VELDNAMEN EN FOUTEN

## LESSTOF

Screencasts: <https://www.youtube.com/playlist?list=PL113E672F784F955B>

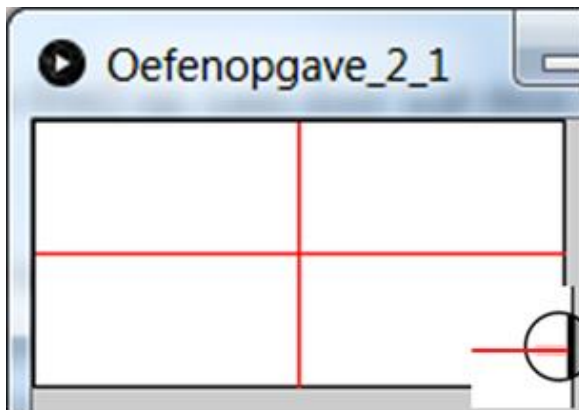
Reader: Hoofdstuk 2

## OEFENOPGAVEN

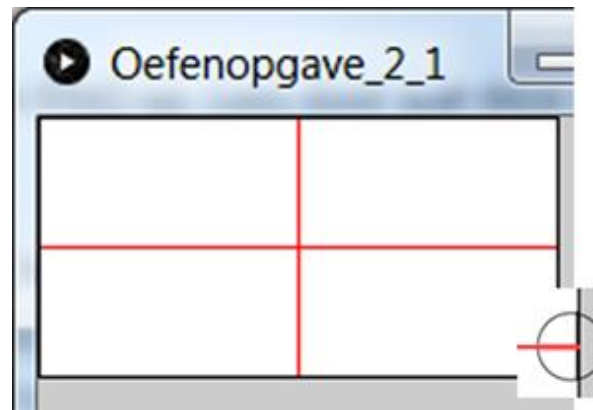
## OPGAVE 2.1

Zoek in de API (of de reference) de methode *line()* op. Lees door wat deze doet en wat de betekenis is van de parameters bij deze methode.

Teken een rechthoek waarvan de breedte 2 keer zoveel pixels heeft als de hoogte (zwarte randen, witten binnenkant). Verdeel met behulp van rode lijnen dit rechthoek in 4 vlakken. Let op, de rode lijnen moeten precies binnen de rechthoek vallen, dus niet over de zwarte rand heen gaan.

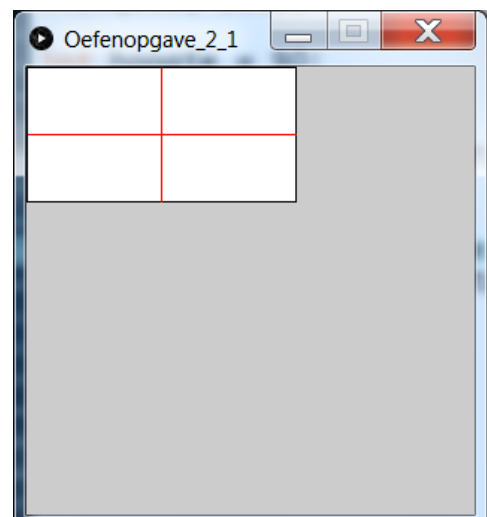


Goed, beide rode lijnen blijven precies binnen de rechthoek



Fout, de horizontale lijn gaat over de rechtrand heen, de verticale lijn gaat over de onderrand heen.

Test je programma met verschillende hoogtes; de breedte van de rechthoek moet zich automatisch aanpassen en de binnenlijnen moeten exact binnen de lijnen blijven.

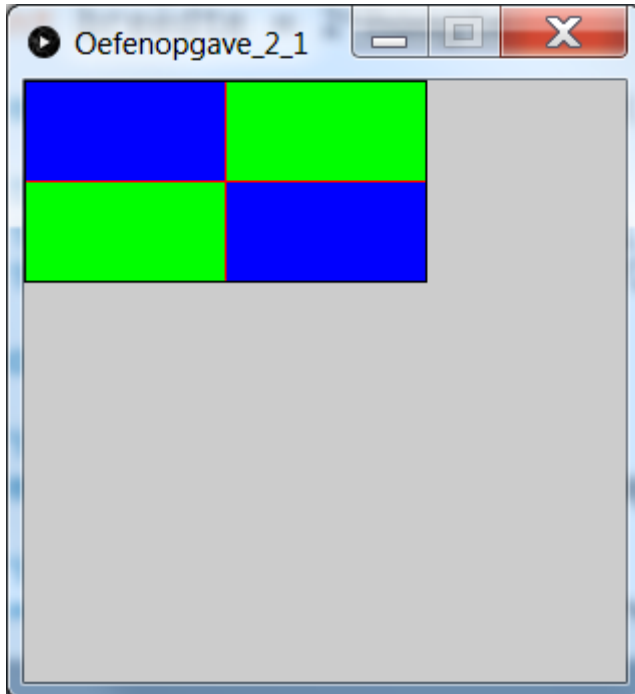


---

OPGAVE 2.2

Teken binnen elk kwadrant een rechthoek met een eigen kleur. Doe dit zo dat de buitenranden en de binnenlijnen niet overschreven worden. Met de functie `noStroke()` kun je rechthoeken tekenen zonder randje.

Het resultaat zou er zo uit kunnen zien:



Tip: Begin met 1 kwadrant en maak dat eerst helemaal goed. Doe daarna de andere kwadranten.

Wat gebeurt er als je de hoogte een even waarde geeft? Bijv. 100. En wat gebeurt er als je de hoogte een oneven waarde geeft?

---

OPGAVE 2.3

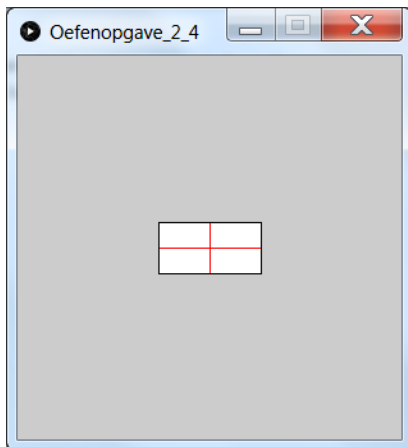
Maak een kopie van je uitwerking van opgave 2.2 en pas deze zodanig aan dat de breedte van de rechthoek niet groter is dan de breedte van het scherm. Gebruik dus de `constrain()-methode`. Let op, de breedte moet nog steeds 2x de hoogte zijn, dus als de rechthoek te breed wordt, heeft dat ook consequenties voor de hoogte. De breedte van het scherm kun je opvragen met de variabele `width` (dat is een standaardvariabele van Processing).

## OPGAVE 2.4

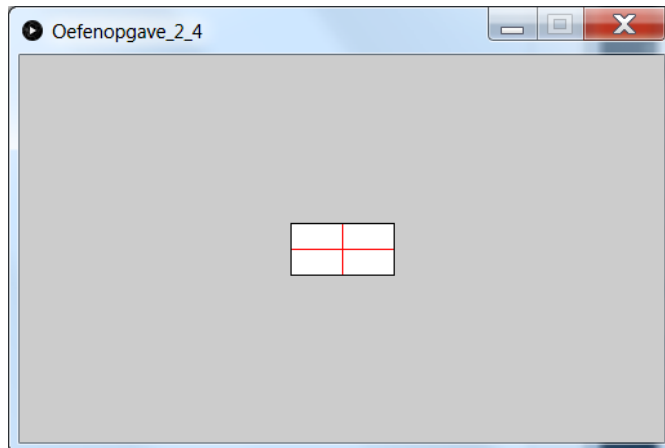
Pas je uitwerkingen van 2.3 zodanig aan dat de rechthoek met lijnen altijd precies midden in het scherm getekend wordt. Neem in je code de volgende twee variabelen en hun initialisatie op en gebruik deze om de rechthoek met rode lijnen te centreren.

```
int middelpuntX = width / 2;  
int middelpuntY = height / 2;
```

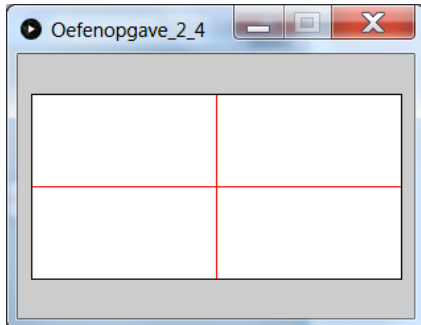
Hieronder een paar voorbeelden van de opgave.



Schermbreedte = 300  
Schermhoopte = 300  
rechthoekhoepte = 40  
(en dus rechthoekbreedte = 80)



Schermbreedte = 500  
Schermhoopte = 300  
rechthoekhoepte = 40  
(en dus rechthoekbreedte = 80)



Schermbreedte = 300  
Schermhoopte = 200  
rechthoekhoepte = 140  
(en dus rechthoekbreedte = 280)

## MODULEOPGAVE 2: BMI

De BMI (body-mass index) geeft de verhouding weer tussen de lengte en het gewicht van een persoon. Deze verhouding wordt gebruikt bij het bepalen of iemand een gezond of ongezond gewicht heeft.

## FORMULE

$$BMI = \frac{\text{gewicht}}{\text{lengte}^2}$$

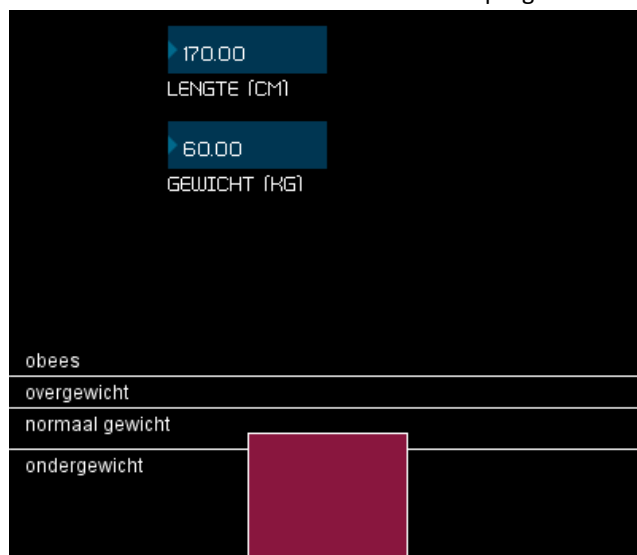
## CATEGORIEËN

Om te bepalen of iemand een gezond of ongezond gewicht heeft, worden de onderstaande categorieën gebruikt<sup>1</sup>:

BMI	Categorie
Minder dan 18,5	Ondergewicht
18,5 tot 25	Normaal gewicht
25 tot 30	Overgewicht
30 of meer	Obees

Maak een programma dat op basis van de lengte en het gewicht de BMI uitrekent en dit laat zien als een staaf. Zorg er ook voor dat het programma laat zien in welke categorie de persoon valt. De lengte en het gewicht moeten kunnen worden ingevoerd door de gebruiker.

Hieronder is een screenshot te zien van het programma:



<sup>1</sup> In werkelijkheid worden er meer categorieën onderscheiden. Zie <http://nl.wikipedia.org/wiki/Queteletindex>

## DEEL 1

Beschrijf welke variabelen je nodig hebt en waarvoor. Probeer hierbij ook rekening te houden dat de opmaak nog steeds goed te zien moeten zijn als je window groter of kleiner wordt. Geef per variabele de naam, datatype en waarvoor ze gebruikt wordt aan.

## DEEL 2

Implementeer je programma in Processing en gebruik daarbij de in deel 1 beschreven variabelen.

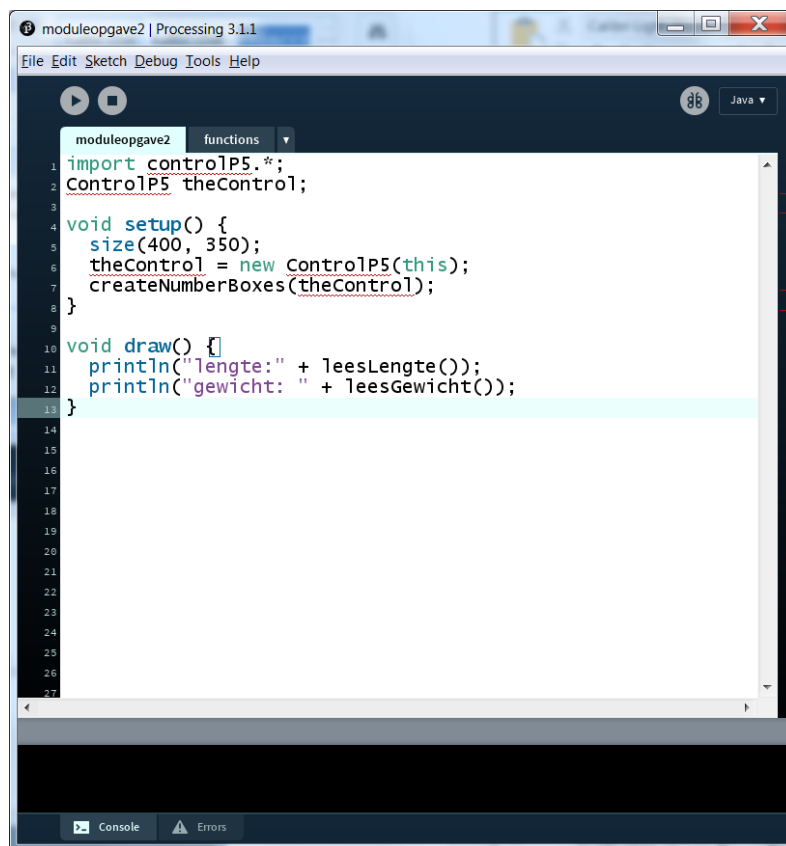
## VOORBEREIDING

In deze opgave gebruiken we twee *numberboxes* om ervoor te zorgen dat de gebruiker de lengte en het gewicht kan invoeren. Deze componenten zijn te vinden in een externe bibliotheek die *ControlP5* wordt genoemd.

Dit soort bibliotheken moeten in processing geïmporteerd worden.

## DE WERKOMGEVING

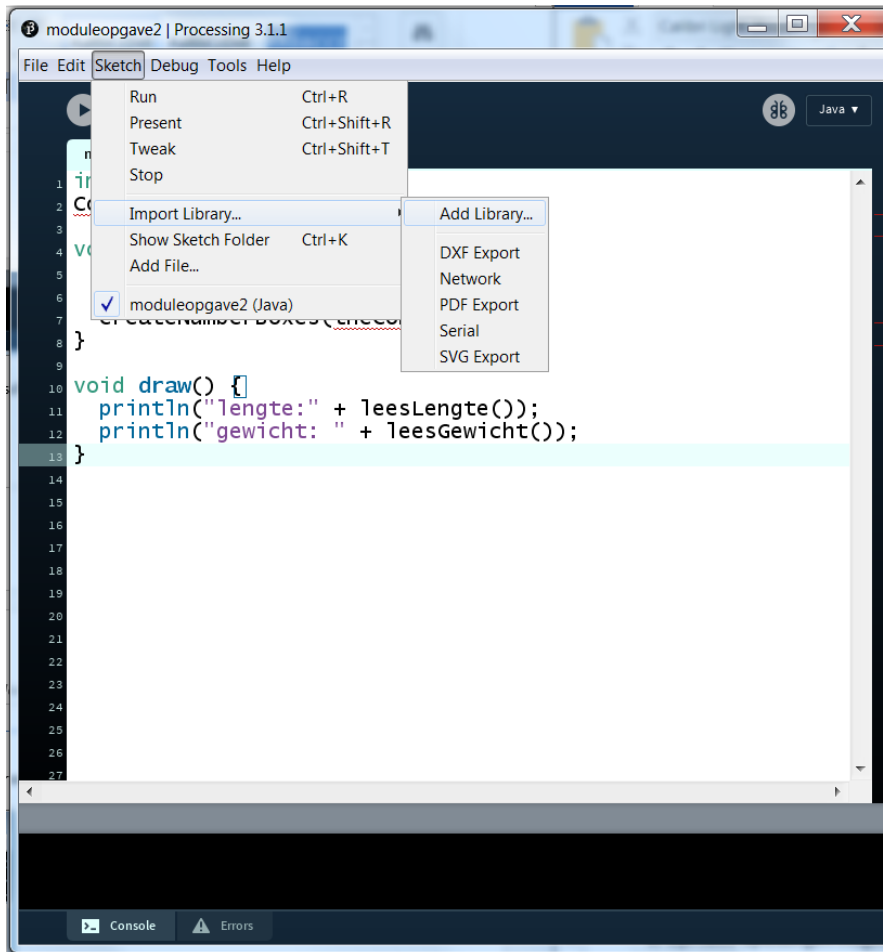
Pak het bestand 'moduleopgave2.zip' uit in een map en open het bestand moduleopgave2.pde in processing. Als het goed is, zijn er twee tabbladen geopend in processing zoals in het onderstaande screenshot te zien is.



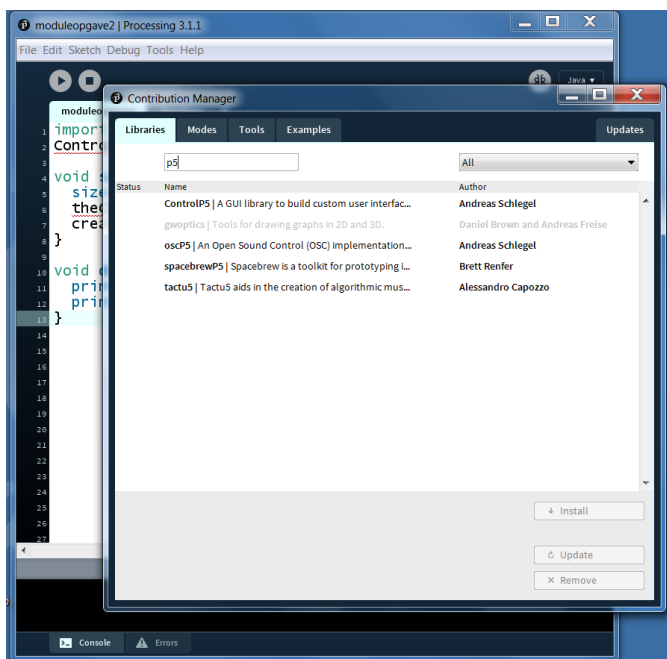
In het tabblad moduleopgave2 kun je je eigen code gaan schrijven. Zorg dat je de code in de methode setup() en de code daarboven niet verwijdert.

## CONTROLP5 INSTALLEREN

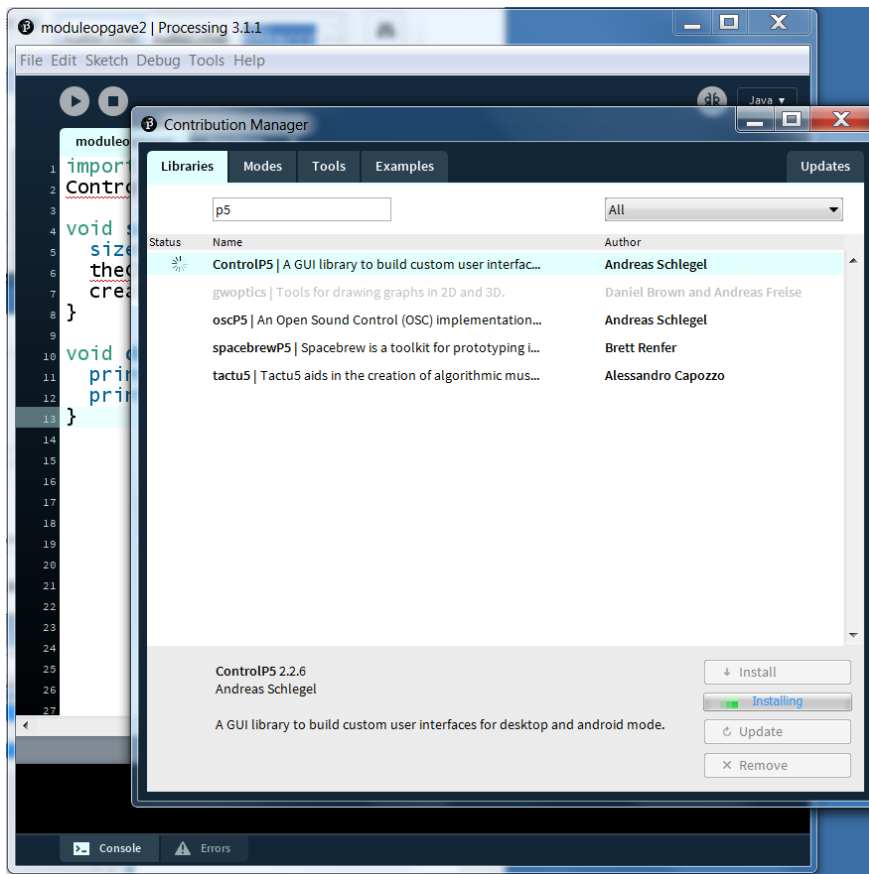
Selecteer in het menu Sketch -> Import Library -> Add Library.



Zoek in het volgende scherm naar "ControlP5"



Selecteer “ControlP5” en klik op “Install”. Sluit de Library Manager als het control goed is geïnstalleerd.



## DE WAARDEN VAN DE NUMBERBOXES GEBRUIKEN

De getallen in de numberboxes kun je gebruiken met de methoden `leesLengte()` en `leesGewicht()`. In de methode `draw()` zie je een voorbeeld van het gebruik van deze twee methoden. Deze code kun je vervangen door je eigen code.

