

Opdracht 1: Huizen

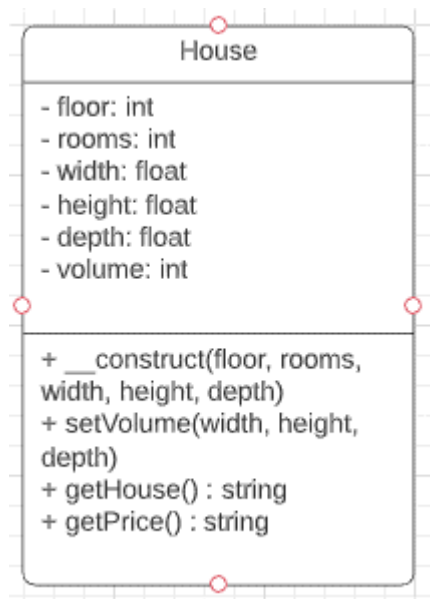
Nodige kennis: class, property, visibility

(video [1 classes](#) + [2 getters & setters](#) + [3 encapsulation](#))

Voor een woonwijk moeten een aantal huizen gebouwd worden.
Per huis kan er van alles verschillen.

De huizen hebben allemaal een
aantal verdiepingen, aantal kamers, een breedte, een hoogte en een diepte. De prijs hangt af
van het aantal kubieke meters.

Bekijk de class diagram hieronder:



Maak 3 huizen aan en print deze uit met alle details met de prijs.

Het resultaat kan er zo uitzien:

Dit huis heeft 2 verdiepingen, 4 kamers en heeft een volume van 100 m3

De prijs van het huis is: 150000

Dit huis heeft 3 verdiepingen, 6 kamers en heeft een volume van 150 m3

De prijs van het huis is: 225000

Dit huis heeft 2 verdiepingen, 3 kamers en heeft een volume van 75 m3

De prijs van het huis is: 112500

Opdracht 2: Herberekening huisprijs

[Publiceren](#)[Bewerken](#)

Nodige kennis: class, property, visibility, array

(video [1 classes](#) + [2 getters & setters](#) + [3 encapsulation](#) + [6 Namespaces](#))

Het blijkt dat de berekening van de prijs bij opdracht 1 niet goed is gegaan. We moeten van elke ruimte weten hoe groot ze zijn om de exacte prijs te bepalen van het huis.

Dit betekent dat een kamer een aparte class moet gaan krijgen, waarbij we de volume gaan bepalen met de lengte, hoogte en breedte.

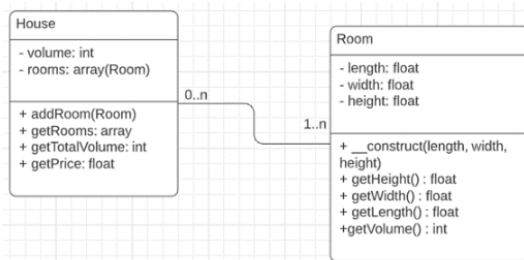
Als we een kamer aanmaken, moeten we er wel voor zorgen dat de kamer aan een huis is gekoppeld. Dit doen we door het object van de kamer in een array te zetten.

Om de prijs te bepalen, zal eerst de grootte van elke kamer berekend moeten worden. De grootte van een kamer hoort bij de kamer, en zal dus in die class moeten gebeuren.

De totale grootte van het huis hoort bij het huis. Je zal dus de array van kamers moeten gebruiken om de totale grootte te bepalen. (loop door de array heen)

Als de totale grootte van het huis is bepaald, kan de prijs worden bepaald.

De class diagram ziet er zo uit:



Het resultaat kan er zo uitzien:

Inhoud Kamers:

- Lengte: 5.2m Breedte: 5.1m Hoogte: 5.5m
- Lengte: 4.8m Breedte: 4.6m Hoogte: 4.9m
- Lengte: 5.9m Breedte: 2.5m Hoogte: 3.1m

Volume Totaal = 298m³

Prijs van het huis is= 894000Euro

Opleveren via je private persoonlijke repository in gitlab. Link inleveren van je repository.

(bijv: https://gitlab.com/techniek-college-rotterdam/sportlaan15/2022sod2a/individuele-opdrachten/voornaam_achternaam/oop-php-prog)

Opdracht 3: Drie op een rij

Nodige kennis: class, property, visibility, inheritance

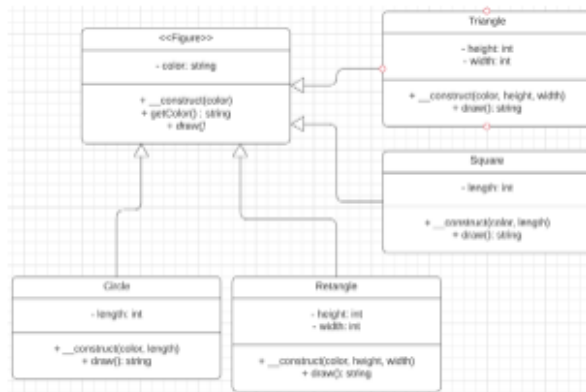
(video [1 classes](#) + [2 getters & setters](#) + [3 encapsulation](#) + [4 Inheritance](#) + [6 Namespaces](#))

Bij deze opdracht gaan we een begin maken van een spel, namelijk drie op een rij. Voor deze versie van drie op een rij gaan we gebruik maken van allerlei soorten figuren:

- Vierkant
- Driehoek
- Rechthoek
- Cirkel

Elk figuur kan een kleur hebben: Rood, groen, blauw, geel, oranje of paars.

Hieronder het Class diagram voor deze opdracht:



Stap 1:

Maak eerst de parent class aan: Figuur

Bepaal de algemene eigenschappen van een figuur.

Stap 2:

Maak de child classes aan voor de soorten. Kijk goed wat je nodig hebt om het figuur te tekenen.

Stap 3:

Zorg dat elk figuur op je scherm staat. Voor nu is het doel om dit met SVG te doen in je html.

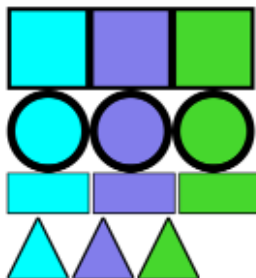
(dus niet met css shapes)

https://www.w3schools.com/graphics/svg_rect.asp ➞

https://www.w3schools.com/graphics/svg_circle.asp ➞

https://www.w3schools.com/graphics/svg_polygon.asp ➞

Resultaat kan er zo uitzien:



Opleveren via je private persoonlijke repository in gitlab. Link inleveren van je repository.