

Naslagmateriaal objectgeoriënteerd programmeren (beschikbaar bij toets)

Notatie van klasse

KlasseNaam
<code>attribuut_1 : type</code> <code>attribuut_2 : type</code> <code>-ditAttribuutIsPrivate : type</code> <code>+ditAttribuutIsPublic : type</code>
<code>methode()</code> <code>methode(parameter_1, parameter_2)</code> <code>methode() : type van teruggeefwaarde</code> <code>setPrivateAttribuut(parameter : type)</code> <code>getPrivateAttribuut() : type</code>

Notatie van object / instantie

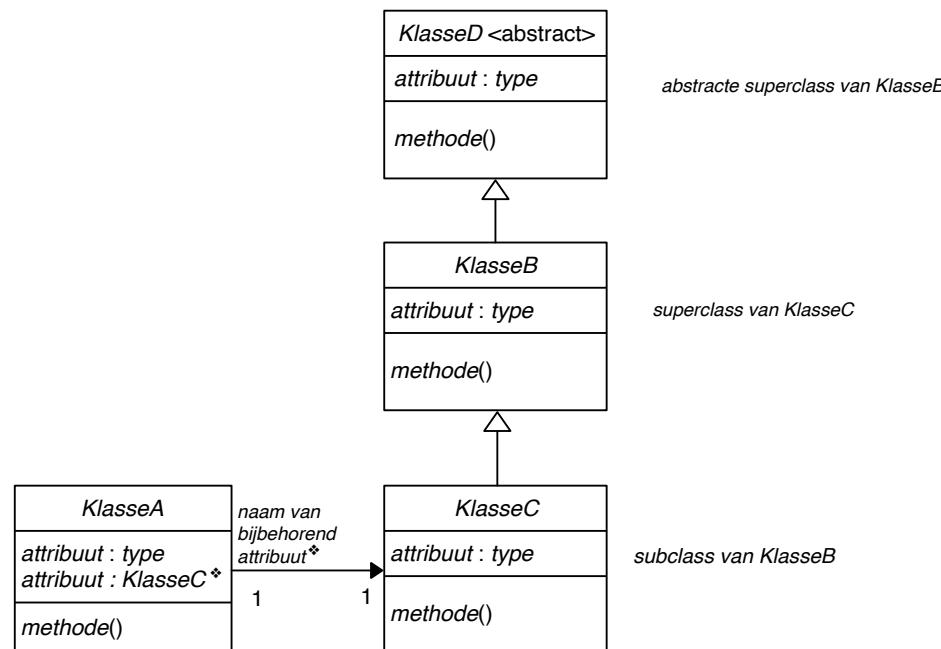
<code>objectnaam : KlasseNaam</code>
<code>attribuut_1 : waarde</code> <code>attribuut_2 : waarde</code>

Afhankelijk van context kun je kiezen of je informatie over toegankelijkheid en types weglaat. Bij de toets wordt dit expliciet aangegeven

Primitieve datatypen in JavaScript zijn o.a.:

- number
- string
- boolean

Ontwerpklassendiagram



* kies één van beide manieren

Naslagmateriaal objectgeoriënteerd programmeren
(beschikbaar bij toets)

Een class in JavaScript

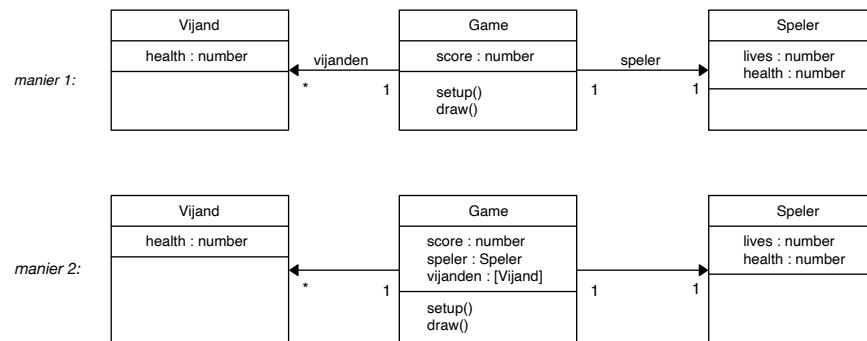
```
class <Naam van de class> {
    #privateAttribuut;
    publicAttribuut;

    constructor(parameter1, parameter2) {
        this.#privateAttribuut = parameter1;
        this.publicAttribuut = parameter2;
    }

    methodenaam() {
        // code die uitgevoerd moet worden

        return <waarde>; // alleen als er een waarde teruggegeven moet worden
    }
}
```

Associatie in JavaScript



```
class Game {
    score;
    vijanden;
    spelers;

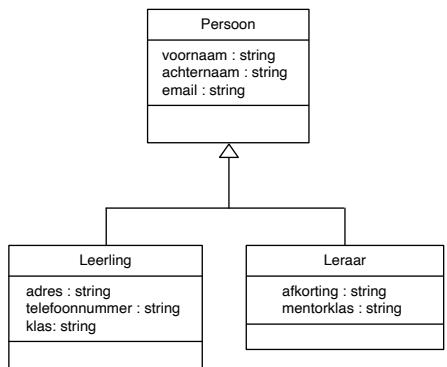
    constructor() {
        this.score = 0;           // een nieuwe Game begint met 0 punten
        this.speler = new Speler();
        this.vijanden = [ new Vijand(), new Vijand(), new Vijand(), new Vijand() ];
    }

    setup() {
        // code voor de setup-methode
    }

    draw() {
        // code voor de draw-methode
    }
}
```

Naslagmateriaal objectgeoriënteerd programmeren
(beschikbaar bij toets)

Overerving in JavaScript



```
class Persoon {
    voornaam;
    achternaam;
    email;

    constructor(voornaam, achternaam, email) {
        this.voornaam = voornaam;
        this.achternaam = achternaam;
        this.email = email
    }
}
```

```
class Leerling extends Persoon {
    adres;
    telefoonnummer;
    klas;

    constructor(voornaam, achternaam, email,
               adres, telefoonnummer, klas) {
        super(voornaam, achternaam, email);
        this.adres = adres;
        this.telefoonnummer = telefoonnummer;
        this.klas = klas
    }
}
```