

Naslagmateriaal objectgeoriënteerd programmeren (beschikbaar bij toets)

Notatie van klasse

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

Notatie van object / instantie

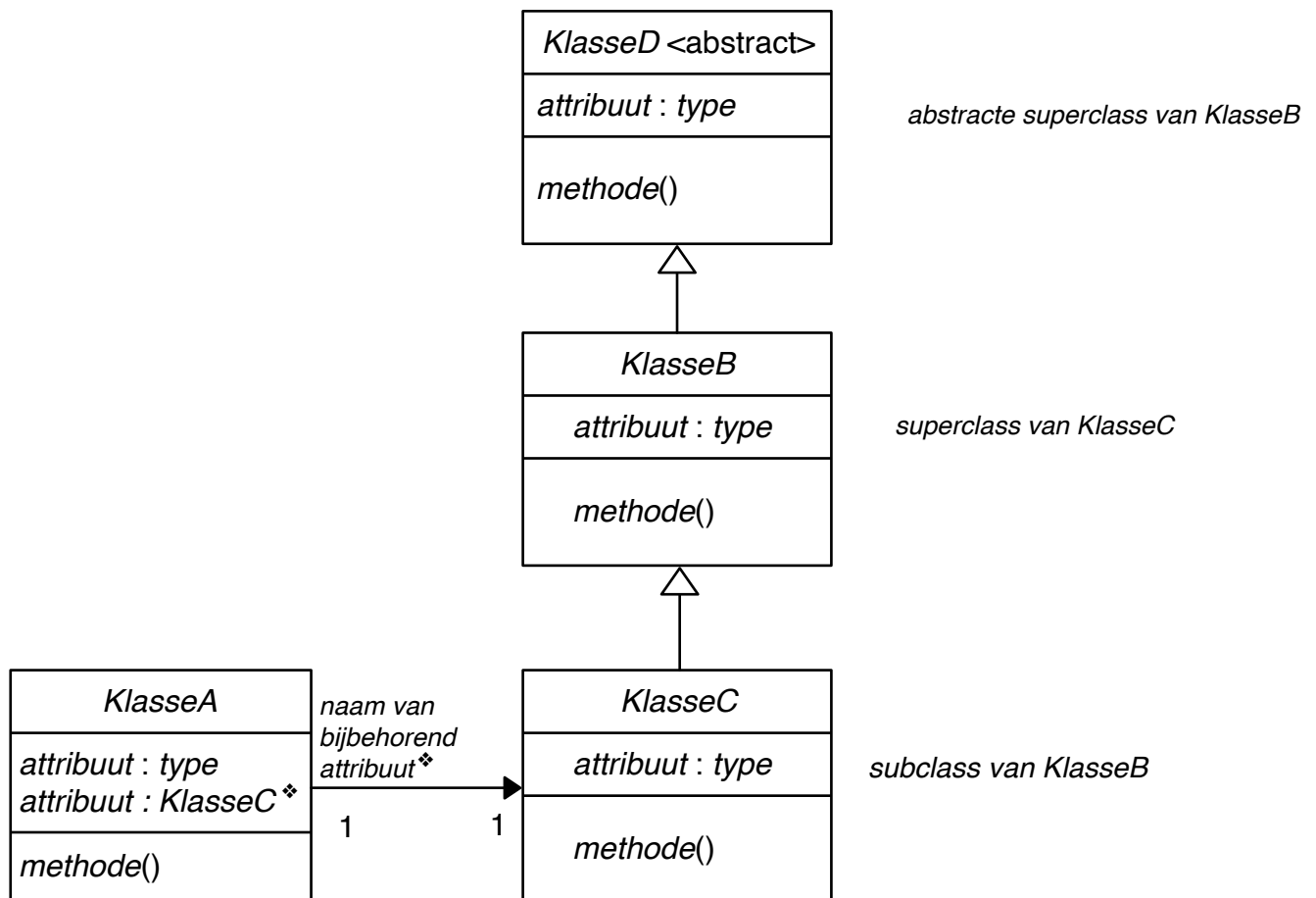
<i>objectnaam : KlasseNaam</i>
<i>attribuut_1 : waarde</i> <i>attribuut_2 : waarde</i>

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



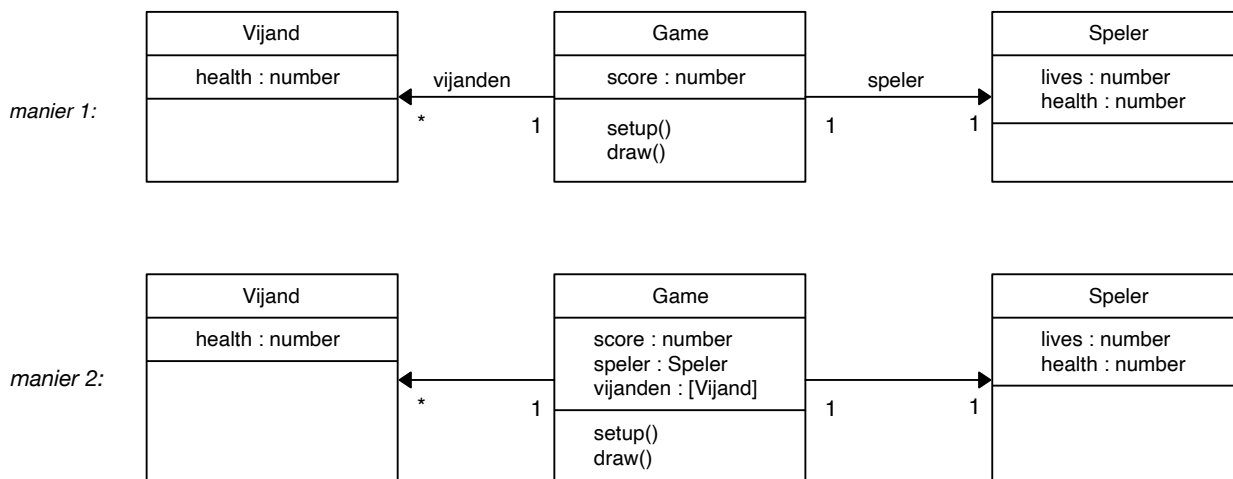
❖ kies één van beide manieren

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  
  }  
}
```

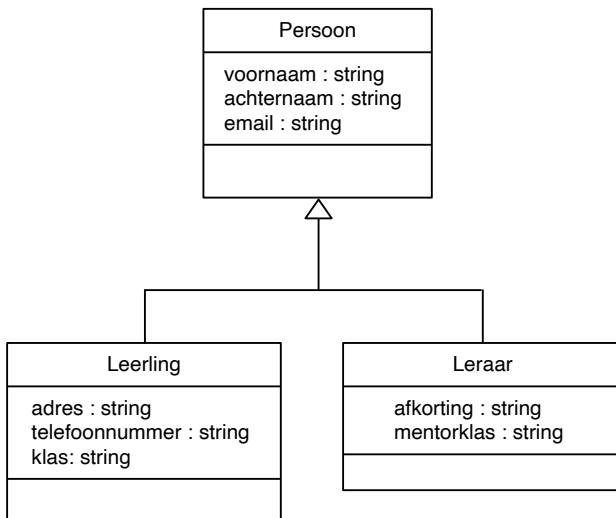
Associatie in JavaScript



```
class Game {  
  score;  
  vijanden;  
  speler;  
  
  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
  }
}
```