Communication and Multimedia Design | Hogeschool van Amsterdam

ttd

OO Libraries

Creating Tomorrow

Object Oriented Programming

Ontstaan in de softwarecrisis

1. Simulaties mogelijk te maken
2. Hergebruik van code mogelijk te maken
   (en dus softwareontwikkeling goedkoper te maken

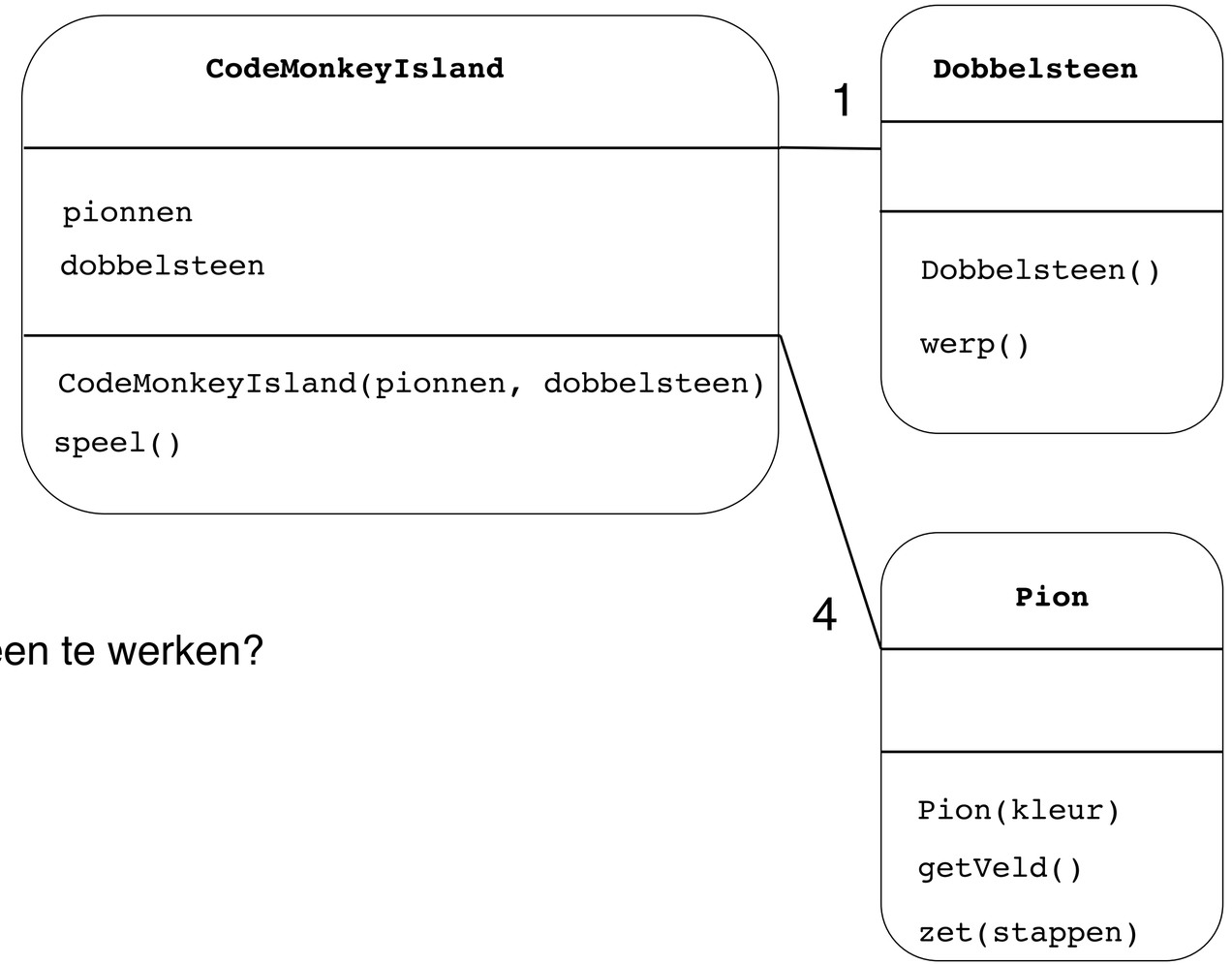
Vergelijk software met een fiets – bestaat uit onderdelen. OO Software ook.

Rollenspel!

Simulatie van een bordspel

De code bestaat uit 3 soorten objecten: **Dobbelstenen, Pionnen en CodeMonkeyIsland** spellen

```
let myDobbelsteen = new Dobbelsteen();

let myPionnen = [new Pion('blauw'),
                 new Pion('rood'),
                 new Pion('geel'),
                 new Pion('groen')
                ];

let codeMonkeyIsland = new CodeMonkeyIsland(myPionnen, myDobbelsteen);
```



Hergebruik van code

Wat moet je weten om met een object van het type Dobbelsteen te werken?

Dobbelsteen
Representeert een zuivere, 6-zijdige dobbelsteen

Attributen

Methoden:
Dobbelsteen – constructor

werp() – simuleert het eenmaal werpen van een dobbelsteen
return: Number uit {1,2,3,4,5,6}

CodeMonkeyIsland
Bewaakt de spelregels van het bordspel

Attributen
pionnen - array objecten van het type Pion
dobbelsteen - Dobbelsteen

Methoden
CodeMonkeyIsland() – constructor, heeft
parameters: pionnen - array objecten van het type Pion
dobbelsteen - Dobbelsteen

speel() – methode die de ‘beurt’ van volgende speler uitvoert

Dobbelsteen
Representeert een zuivere 6-zijdige dobbelsteen

Attributen

Methoden:
Dobbelsteen – constructor

werp() – simuleert het eenmaal werpen van een dobbelsteen
return: Number

Pion
Beeldt de toestand van de pion af op het bord

Attributen

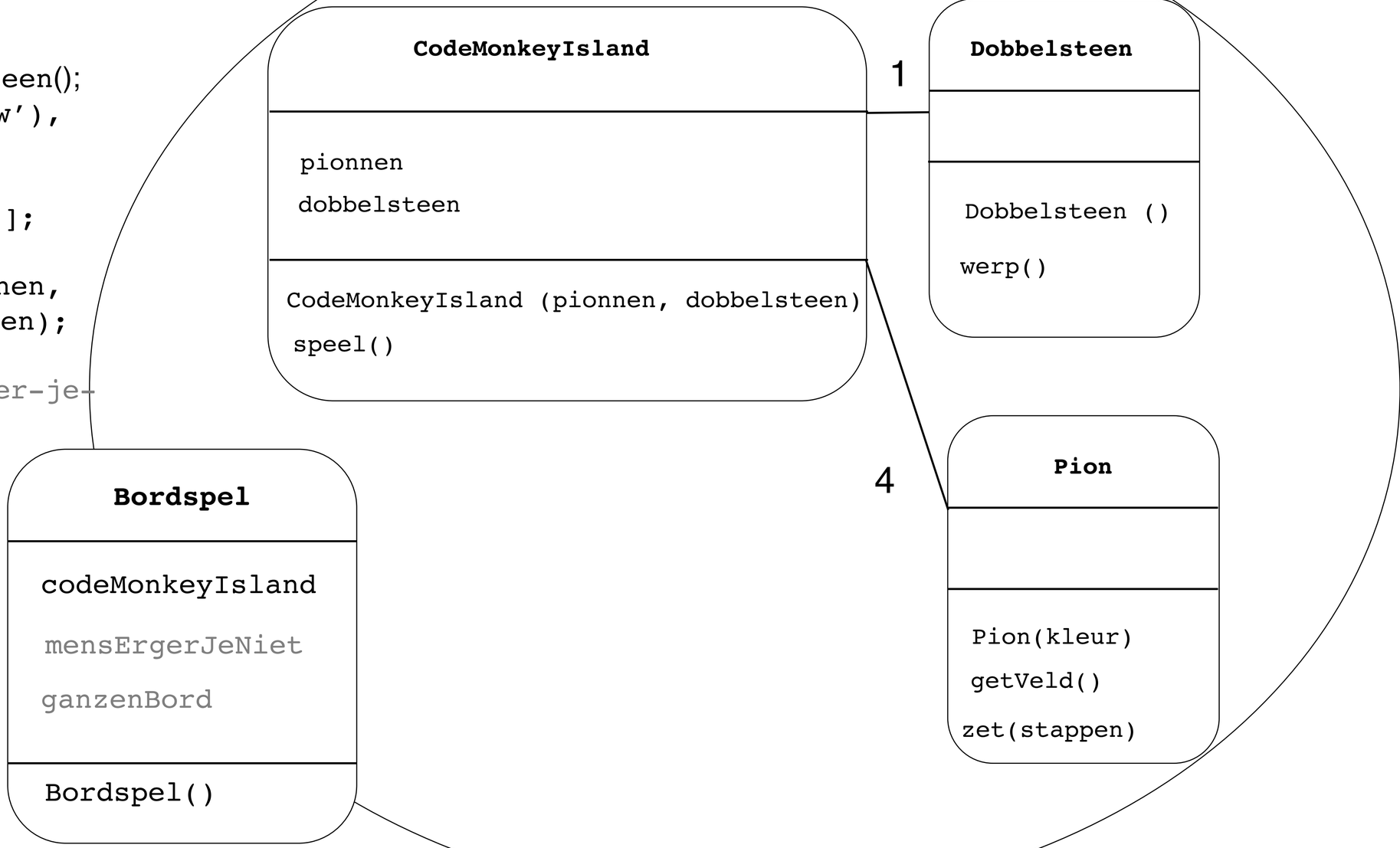
Methoden:
Pion() – constructor, heeft
parameter: kleur - String uit {‘rood’, ’blauw’, ‘geel’, ‘groen}

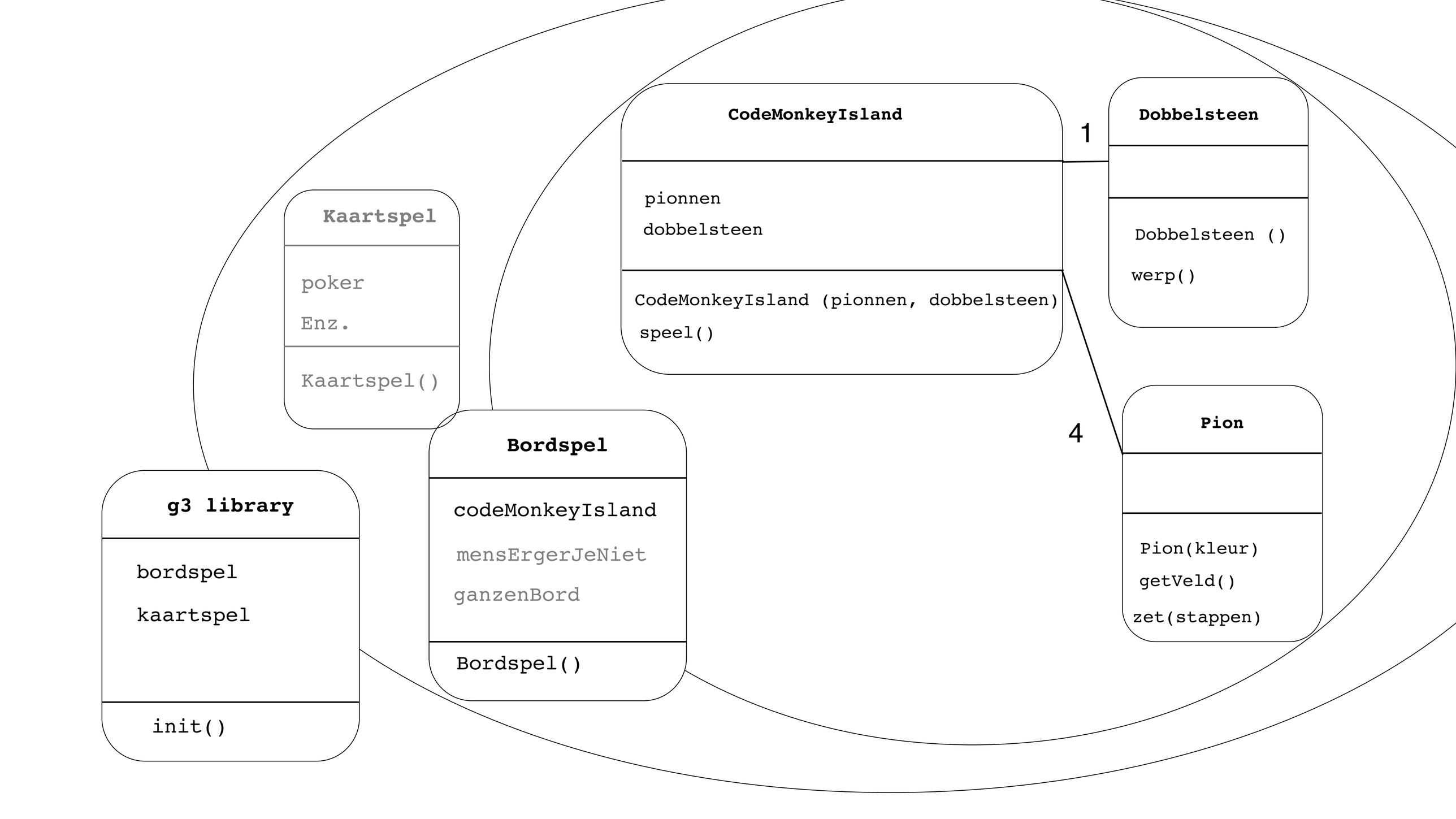
getVeld() – geeft informatie terug over de toestand van de Pion
return: String

zet() – verandert de toestand van de Pion, heeft
parameter: stappen - Number.

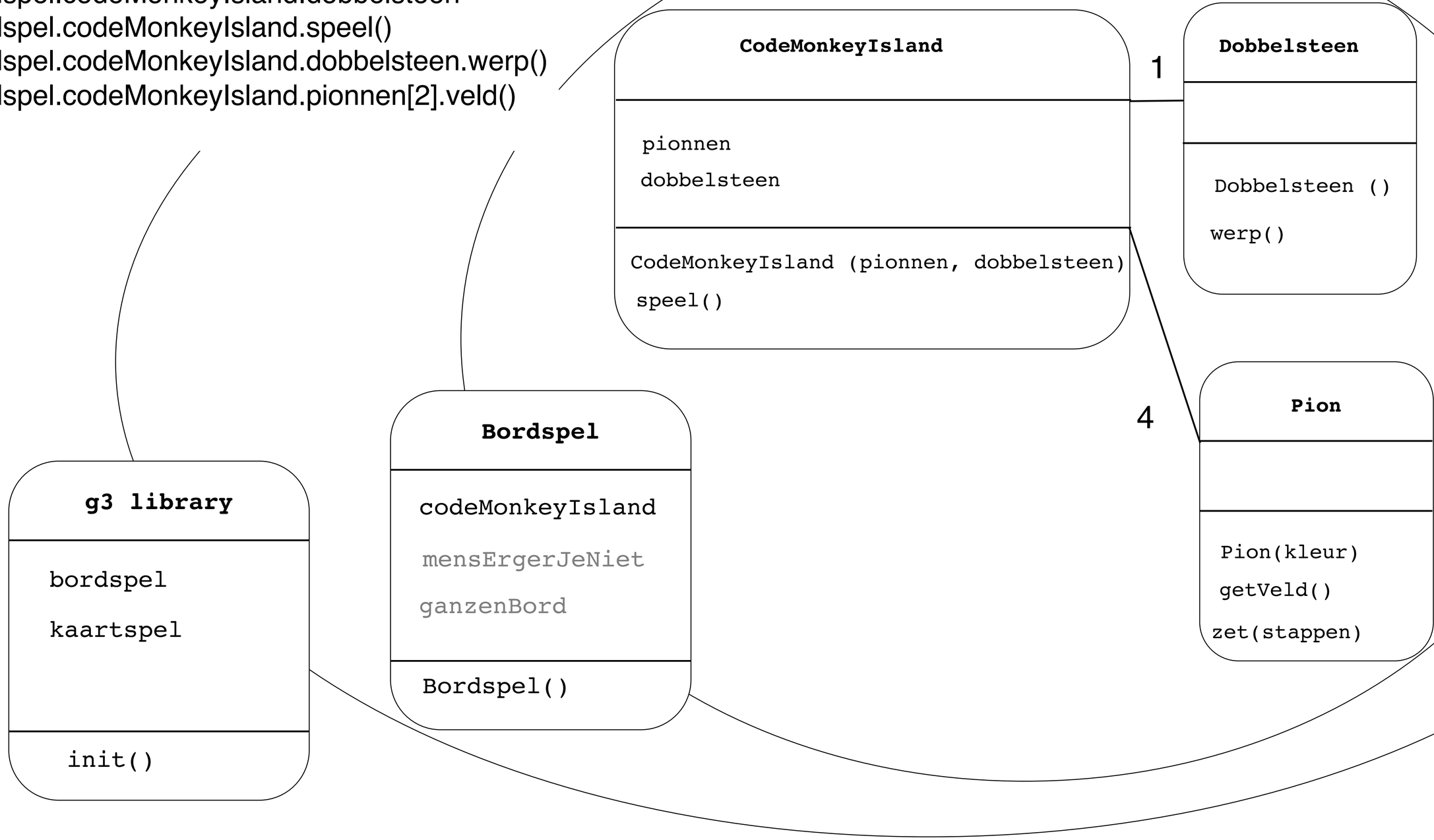
Module Bordspel

```
Bordspel() {
 let myDobbelsteen = new Dobbelsteen();
 let myPionnen = [new Pion('blauw'),
                  new Pion('rood'),
                  new Pion('geel'),
                  new Pion('groen')];
this.codeMonkeyIsland =
     new CodeMonkeyIsland(myPionnen,
                          myDobbelsteen);

enz. voor ganzenbord en mens-erger-je-
niet

}
```




CodeMonkeyIsland
pionnen
dobbelsteen
CodeMonkeyIsland (pionnen, dobbelsteen)
speel()
1
Dobbelsteen
Dobbelsteen ()
werp()
4
Pion
Pion(kleur)
getVeld()
zet(stappen)
Kaartspel
poker
Enz.
Kaartspel()
Bordspel
codeMonkeyIsland
mensErgerJeNiet
ganzenBord
Bordspel()
g3 library
bordspel
kaartspel
init()

g3.bordspel.codeMonkeyIsland.dobbelsteen
g3.bordspel.codeMonkeyIsland.speel()
g3.bordspel.codeMonkeyIsland.dobbelsteen.werp()
g3.bordspel.codeMonkeyIsland.pionnen[2].veld()



Documentatie

g3
Library waarmee verschillende spellen kunnen worden gesimuleerd: (bordspellen, kaartspellen, gokspellen enz.)

g3 is opgebouwd uit modules:

Bordspel (Bevat de API van bordspellen)
codeMonkeyIsland,
mensErgerJeNiet,
ganzebord

kaartspel (Bevat de API van kaartspelen)
poker,
pesten,
klaverjassen

CodeMonkeyIsland
Bewaakt de spelregels van het bordspel

Attributen
pionnen - array objecten van het type Pion
dobbelsteen - Dobbelsteen

Methoden
CodeMonkeyIsland() – constructor, heeft
parameters: pionnen, array objecten van het type Pion
dobbelsteen - Dobbelsteen

speel() – methode die de 'beurt' van volgende speler uitvoert

Dobbelsteen
Representeert een zuivere, 6 zijdige dobbelsteen

Attributen

Methoden:
Dobbelsteen – constructor

werp() – simuleert het eenmaal werpen van een dobbelsteen
return: Number

Vragen?