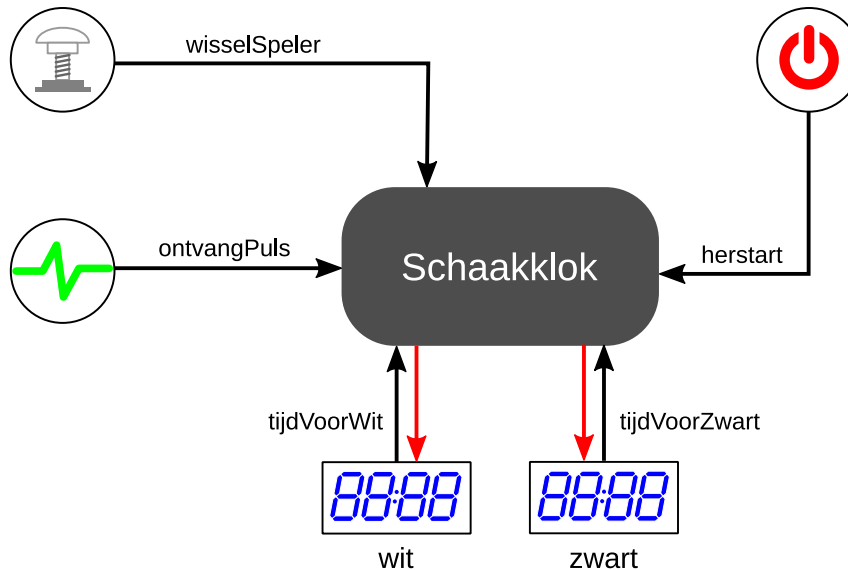


De schaakklok

programmaontwerp voor beginners

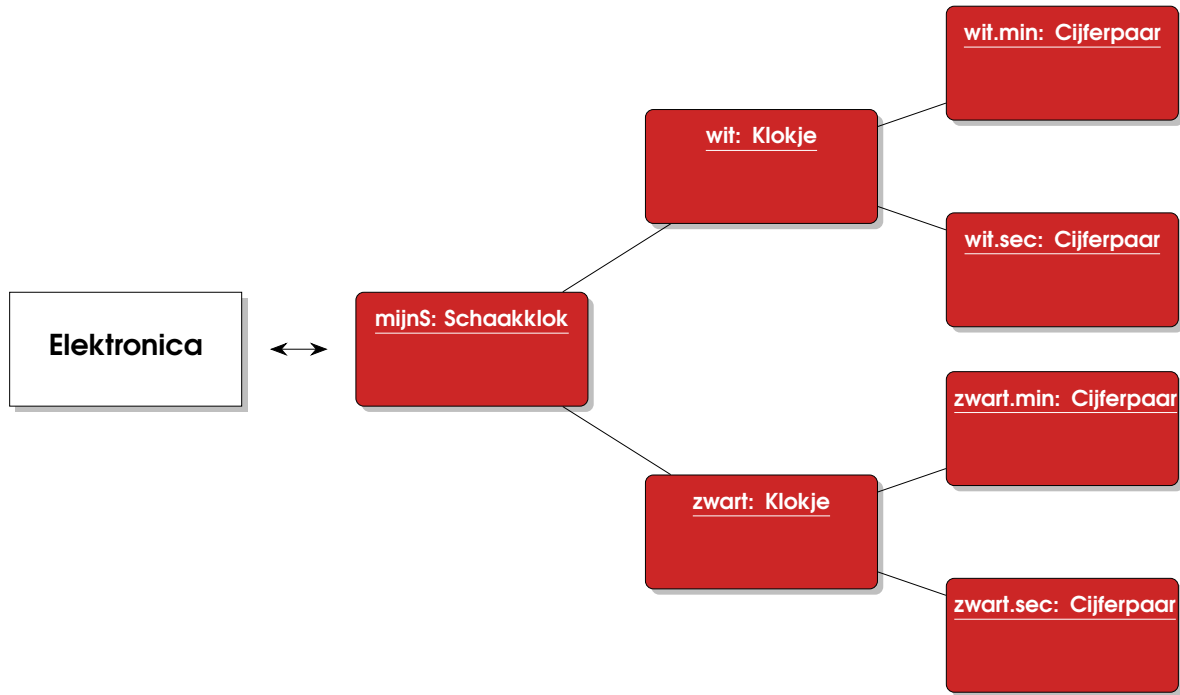


3 inputs — 2 outputs

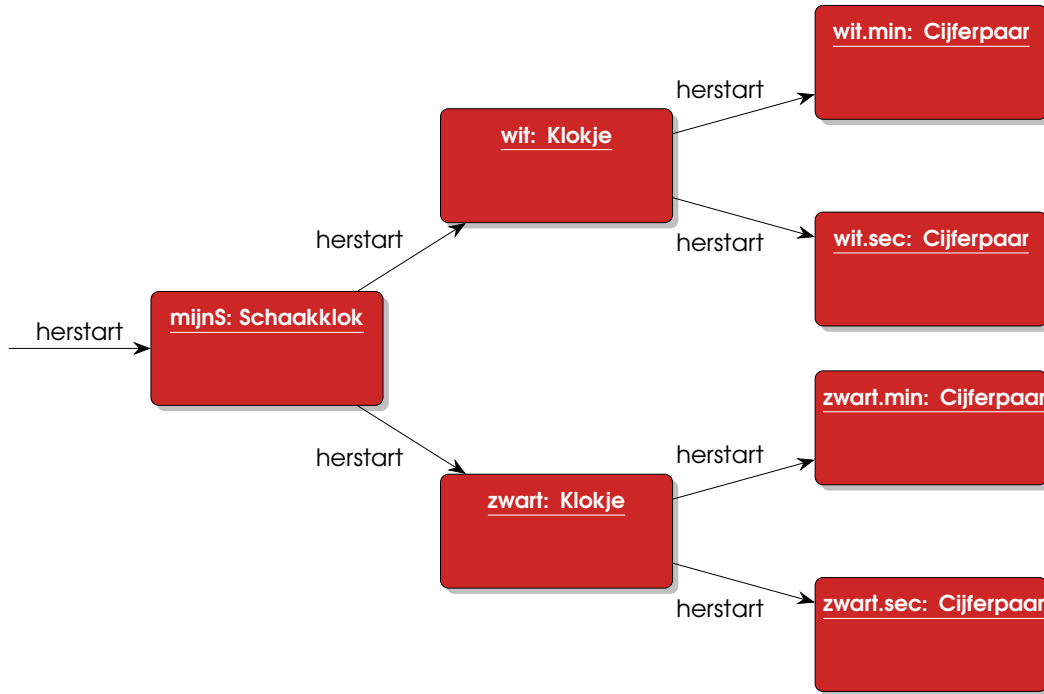


'Output' is een *string* van de vorm 2 cijfers – dubbele punt – 2 cijfers, bijv. 05:23.

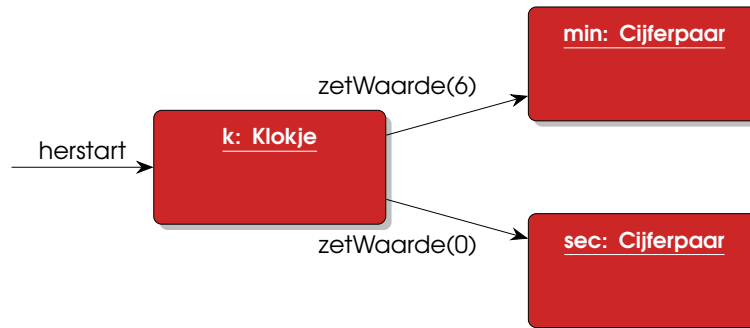
Objecten



Herstart — klokken terug op 06:00



... meer specifiek



Dit schema is geldig voor beide klokjes, zowel voor de witte speler als de zwarte.

Objecten hebben velden

Elk cijferpaar houdt een eigen teller bij

min: Cijferpaar

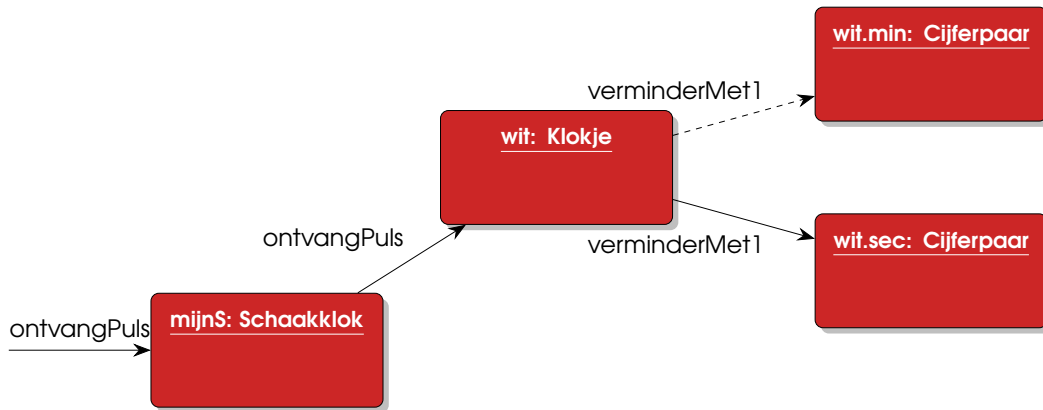
teller

Ook de schaakklok heeft een veld.

mijnS: Schaakklok

speler

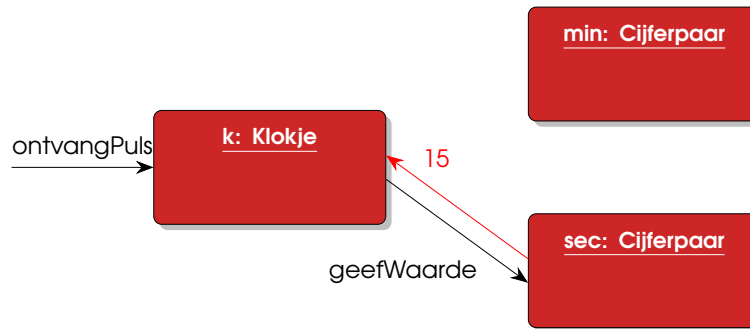
Ontvang puls — (gebeurt elke seconde)



Wanneer wordt *verminderMet1* opgeroepen voor *min* en wanneer voor *sec*?

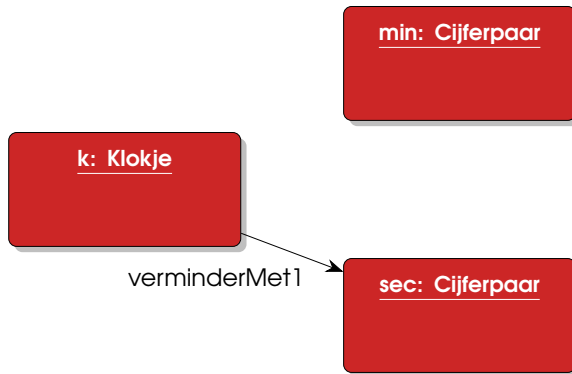
Funciemethoden

Een methode kan ook een waarde teruggeven.

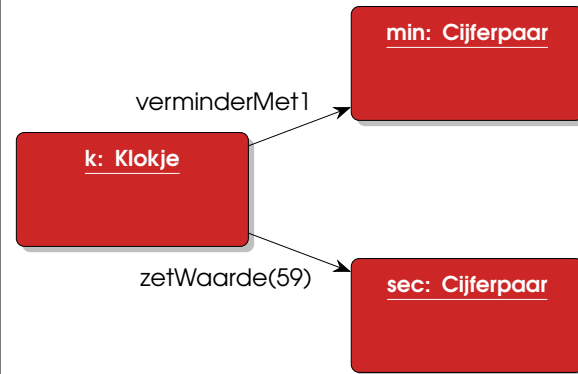


Ontvang puls — Klokje

geefWaarde gaf geen nul terug

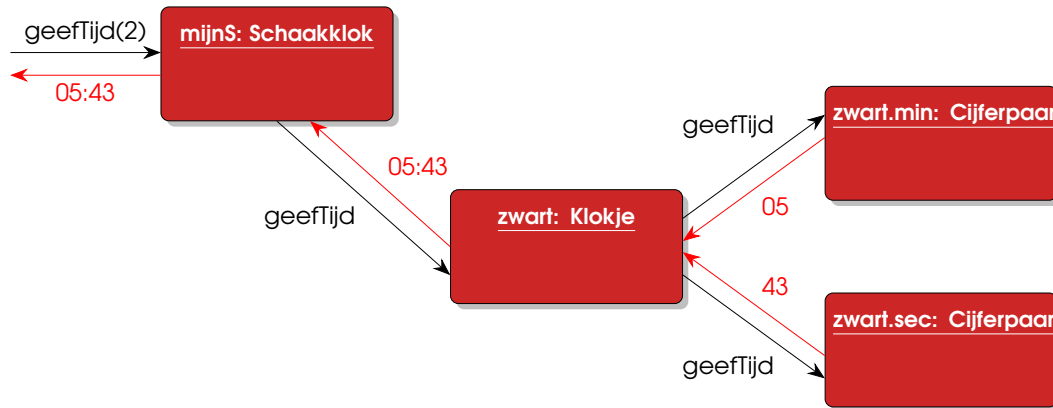


geefWaarde gaf nul terug



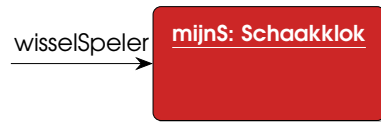
Resterende tijd

We hernoemen *tijdVoorWit* en *tijdVoorZwart* tot *geefTijd(..)*.



Van speler wisselen

Last but not least . . .

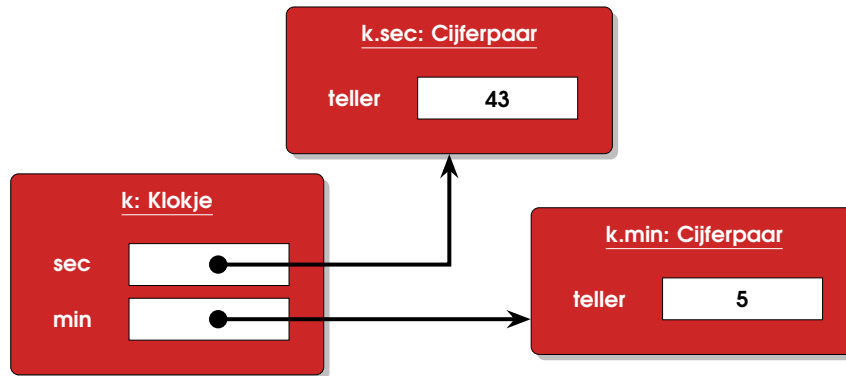


Klassen

- *Klasse* = objecten die hetzelfde gedrag vertonen
- Objecten zijn *instanties* van een klasse



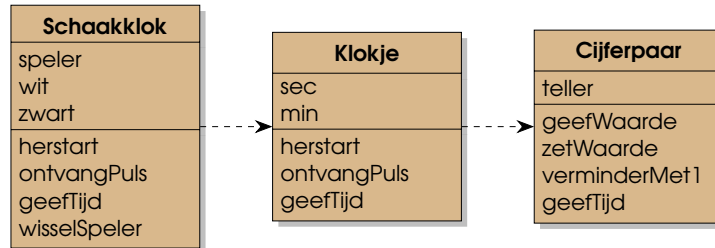
Objecten verwijzen naar andere objecten



Ook de schaakklok heeft twee dergelijke velden *wit* en *zwart*.

Implementaties

- Welke velden moet elke klasse hebben?
- Welke methoden?
- Wat doen deze methoden precies?



Cijferpaar

geefWaarde()

Geef het getal terug dat is opgeslagen in het veld *teller*.

zetWaarde(w)

Stop het getal *w* (de waarde van de parameter) in het veld *teller*.

verminderMet1()

Verminder de waarde die is opgeslagen in *teller* met 1.

geefTijd()

Als het getal dat opgeslagen is in *teller* groter is dan 9,
dan geef je dit getal terug (in de vorm van een string),
anders geef je dit getal terug voorafgegaan door het cijfer 0.

Klokje

herstart()

Roep *zetWaarde(0)* op voor (het object waarnaar verwezen wordt door) *sec* en roep *zetWaarde(6)* op voor *min*.

ontvangPuls()

Roep *geefWaarde()* op voor *sec*.

Als deze waarde nul is,

dan roep je *zetWaarde(59)* op voor *sec* en *verminderMet1()* voor *min*,

anders roep je *verminderMet1()* op voor *sec*.

geefTijd()

Roep *geefTijd()* op voor *sec* en voor *min*. Combineer deze strings tot één geheel door er een dubbele punt tussen te plaatsen. Geef dit resultaat terug.

Schaakklok

herstart()

Roep *herstart()* op voor zowel *wit* als *zwart*.
Plaats de waarde 1 in het veld *speler*.

geefTijd(s)

Als *s* gelijk is aan 1,
dan roep je *geefTijd()* op voor *wit* en geef je het resultaat terug,
anders geef je het resultaat terug van *geefTijd()* voor *zwart*.

ontvangPuls()

Als de waarde van het veld *speler* gelijk is aan 1,
dan roep je *ontvangPuls()* op voor *wit*,
anders roep je *ontvangPuls()* op voor *zwart*.

wisselSpeler()

Als de waarde van *speler* gelijk is aan 1,
dan plaats je 2 in (het veld) *speler*,
anders plaats je 1 in *speler*.

Initialisatie van objecten — constructoren

Cijferpaar

Plaats de waarde 0 in het veld *teller*.

Klokje

Maak een nieuw *Cijferpaar*-object aan en plaats (een verwijzing naar) dit object in *sec*.

Maak een nieuw *Cijferpaar*-object aan en plaats het in *min*.

Schaakklok

Maak een nieuw *Klokje*-object aan en plaats het in *wit*.

Plaats ook een nieuw *Klokje*-object in *zwart*.

Geef *speler* de waarde 1 (t.t.z., plaats de waarde 1 in het *speler*-veld).