

Bestudeer de video van de volgende les geheel:

Modules: (ont)koppeling en cohesie

Experimenteer zelfstandig met de daarin behandelde voorbeeld-programma's. Een link naar de broncode daarvan staat onder de video. In de volgende les kun je hierover vragen stellen.

Opgaven bij de huidige les:

In te leveren uiterlijk 2 dagen vóór de volgende les.

Niveau 1

Maak een programma dat een race tussen twee of meer auto's simuleert. Elke auto is een object van klasse *Car*.

Onveranderlijke eigenschappen, meegegeven aan de constructor en opgeborgen in fields (instance variables) zijn: *brand* (merk), *accelation* (versnelling [m/s²]) en *maxSpeed* (maximale snelheid [m/s]).

Veranderlijke fields zijn *speed* en *position*. Zie opgave in de les voor hoe je deze fields berekent.

Een *Car* heeft de method *drive* met parameter *deltaTime*. In deze method worden snelheid en positie aangepast. Daarnaast kan elke auto z'n positie rapporteren.

Start een eindeloze *while*-lus buiten de klasse minimaal twee auto's, laat ze met maximale snelheid accelereren door in elke iteratie van elke auto *drive* aan te roepen en druk merk, snelheid en positie steeds af, voor alle auto's naast elkaar. Op die manier ontstaat een tabel op je scherm met voor elke auto drie kolommen.

Zorg dat *deltaTime* de ECHTE verstreken kloktijd tussen twee calls naar *drive* is. Gebruik de *time* module om dit te bewerkstelligen. Bestudeer zelfstandig de 'officiële' Python docs voor de werking van deze module. Gebruik alleen modules die bij de standaard Python distributie zitten. Gebruik GEEN timers, game libraries etc.

Niveau 2

Hetzelfde als bij niveau 1, maar nu met een visualisatie met behulp van de *turtle* library. De weg bestaat uit twee evenwijdige lijnen. De auto's zijn verschillend gekleurde blokjes. Geef de gewenste kleur mee aan de constructor van de auto. Kijk als voorbereiding nog eens naar de *turtle*-voorbeeldcode bij les 1.

Niveau 3

Hetzelfde als bij niveau 1, maar nu multi-threaded. Elke auto heeft eigen thread en z'n eigen while loop, die dit maal onderdeel uitmaakt van de *drive* method. Een aparte thread schrijft ca. 2x per seconde de resultaten naar het scherm. Als optioneel extra: Maak ook hierbij een visualisatie met een weg en met de auto's als blokjes of iets mooiers.

--//--