

## Week 11 – Labo voor MCT

### Doelstellingen

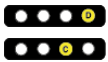
Je bent in staat om Object Georiënteerd Programmeren via de programmeertaal Python toe te passen. Concreet pas je volgende zaken toe:

- Toepassen van logging

*Dit labo-document is specifiek voor studenten MCT. Ben je MIT student? Dan werk je deze week aan andere oefeningen.*

### Afspraken

#### Eindniveau - oefeningen



Ben je een student MCT, dan beheers je de oefeningen tot moeilijkheidsgraad "D"

Ben je een student MIT, dan beheers je de oefeningen tot moeilijkheidsgraad "C"

#### GitHub

Alle oplossingen van week 10 dienen op Github geplaatst te worden. Volg hiervoor de procedure uitgelegd op Leho. Via Github krijg je ook alle bronmateriaal voor deze opgave.

Om je repository in onze Github-classroom aan te maken, klik je op volgende link:  
<https://classroom.github.com/a/Xd5Jn7gC>

Na elke oefening kan je een 'commit & push' doen zodat jouw versie op GitHub steeds aangepast wordt. Geef telkens een gepaste message mee.

Niet afgewerkte oefeningen werk je thuis verder af: voer regelmatig een 'push & commit'-opdracht uit zodat alle oplossingen op je github-repository beschikbaar zijn.

Bij een programmeertaal zoals Python onder de knie krijgen is veelvuldig oefenen essentieel en een noodzakelijke voorwaarde. Daarom vind je in elk labo-document nog twee extra onderdelen. Deze worden als volgt aangeduid.



#### Uitbreidingsoefeningen - eigen onderzoek

Dit onderdeel gaat verder dan de geziene leerstof van deze week. Vaak zijn de opdrachten net iets moeilijker dan hetgeen je in het labo deed. Je zal de Python [documentation](#) en Google nodig hebben voor dit onderzoek.

We motiveren iedereen om dit (thuis) iedere week voor te bereiden. Je onderzoekt in dit onderdeel een onderwerp die de volgende weken terugkomt in de theorie of het labo.

- Oefeningen voor thuis

In dit onderdeel vind je analoge oefeningen zoals je reeds in het labo maakte.

Deze oefeningen hebben dezelfde moeilijkheidsgraden als in het labo. Het is pas door de oefeningen thuis “alleen” te maken dat je je de leerstof eigen maakt. Loop je vast bij een oefening? Herbekijk de theorie, kijk of je een analoge oefening terugvindt die je maakte tijdens het labo. Lukt het nog steeds niet? Kom met je voorbereiding naar het monitoriaat!

## Feedback labo week 10

Lees nog even de algemene opmerkingen uit de theorieles na. Je vindt ze op Leho terug.

Officiële documentatie van Python: <https://docs.python.org/3.9/>

Handige tutorial: <https://docs.python.org/3.9/tutorial/index.html>

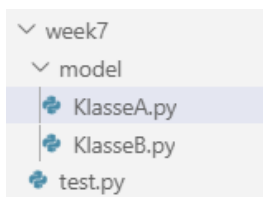
Naming conventions binnen Python: <https://www.python.org/dev/peps/pep-0008/>

## Opgelet!

Vaak gebeuren er fouten tegen het importeren van een klasse.

### Voorbeeld:

Klasse A maak gebruik van klasse B. Klasse A moet daarom klasse B importeren. De testmethode zit één niveau hoger



Klasse A importeert klasse B. Klasse A en klasse B zitten in dezelfde map. We kijken vanuit het standpunt van de geopende projectmap!	<pre>from model.KlasseB import KlasseB # alternatief # from .KlasseB import KlasseB  class KlasseA:      def __init__(self):         self.__var1 = KlasseB()</pre>
Test.py maakt gebruik van de klasseA. We vermelden de submap 'model' waarin hij moet zoeken.	<pre>from model.KlasseA import KlasseA  object_a1 = KlasseA() object_a2 = KlasseA()</pre>

*Opm: wanneer jouw pylint actief is, dan geeft hij bij onderstaande methode een warning. Dit is evenwel **GEEN** uitvoeringsfout!*

```
1  from .KlasseB import KlasseB
2
3  class KlasseA:
4
5      def __init__(self):
6          self.__var1 = KlasseB()
```

## Oefeningen

Vermeld in commentaar telkens de opgave!



### Oef 01

Gegeven onderstaande methode (zie bronmateriaal).  
Vervang alle print-opdrachten door log-opdrachten.

```
def genereer_even_getallen(aantal):  
    even_getallen = []  
  
    while len(even_getallen) != aantal:  
        getal = random.randint(0, 1000)  
        print(f"Gekozen getal: {getal}")  
        if getal % 2 == 0 and getal not in even_getallen:  
            even_getallen.append(getal)  
        print(f"De lijst met even getallen: {even_getallen}")  
    return even_getallen
```

Test even uit wat de verschillende log-levels precies inhouden.  
Hoe kan men het formaat van een log-bericht beïnvloeden?  
Hoe kan men naast het scherm ook naar een bestand loggen?

Meer info vind je alvast hier:

<https://docs.python.org/3.9/howto/logging.html#logging-basic-tutorial>



### Oef 02

Neem oefening 2 uit week9 (Make-up oefening) of gebruik het bronmateriaal.  
Integreer verschillende log-berichten:

- in de verschillende setter-methodes waar er inputvalidatie plaats vindt: log de verkeerd doorgegeven waarde;
- in de methode 'load\_products':
  - log het aangemaakte MakeupProduct-object;
  - log op het einde het aantal ingelezen objecten.



### Oef 03:

Voorbeeld labo-examen. Zie afzonderlijk document.

Kies voor de versie van MCT. Maak dit voorbeeldexamen in max 2,5u.