

Voor- en achternaam:

Groep:

Opleiding **MCT**

Afstudeerrichting/Keuzetraject NMCT

Module **Basic Programming**Lector **Stijn Walcarius**Nagelezen door **Dieter Roobrouck, Christophe Laprudence, Johan Vannieuwnhuysse**Academiejaar **2018-2019**

Semester 1

Datum en uur **21/01/2019, 13u30 – 17u30**

Zittijd 1

LABO EXAMEN 2A**RESULTAAT****/20****ALGEMENE INSTRUCTIES EXAMEN**

Dit examen maakt 70% uit van de evaluatie voor de module Basic Programming.

Schrijf je naam en voornaam bovenaan deze pagina.

Vooraf:

Ga naar Leho om alle bronmateriaal af te halen.

Maak een lege projectmap aan met volgende naam: **<GROEP>_<Naam>_<Voornaam>_Q2A**

Bijvoorbeeld: 1NMCT1_VanDoren_Dirk_Q2A

Maak een nieuw project aan in deze map. In deze map dienen alle bestanden van je examen te komen.

Examen:

Maak de toepassing(en) zoals verder in de opgave beschreven. Lees eerst deze opgave rustig en aandachtig door. Een programma dat afwijkt van deze opgave kan een verlies van punten tot gevolg hebben.

Volg de gevraagde naamgeving indien de opgave dit vermeldt. Je maakt gebruik van de programmeertaal Python.

Indienen:

Als je klaar bent of als de voorziene tijdsduur verlopen is:

- Sla alles op.
- Controleer of je map de juiste naam heeft (zie boven) en of de map alle nodige files bevat.
- Zip of rar de map en roep een docent om toestemming te krijgen om in te dienen via internet.
- Dien je project in op Leho bij de dropbox van Basic Programming. Belangrijk is om elke docent te selecteren.
- Controleer zelf bij de docent of de map correct is ingediend. Dit is je eigen verantwoordelijkheid!

Als dat in orde is geef je deze ingevulde opgave aan de docent af en teken je het blad met aanwezigheden.

Opgelet:

Iedere vaststelling van onregelmatigheid (o.a. GSM, spieken) wordt conform het OER gemeld aan de betrokken student en aan de voorzitter van de examencommissie.

Veel succes!

OPGAVE 1

Bestudeer het bestand 'snelheidsmetingen.json'. Dit bestand is afkomstig van Open Data Kortrijk en lijst alle snelheidsovertredingen per maand in 2012 rond Kortrijk op. Je vindt het bestand ook in het bronmateriaal terug.

We maken een kleine console-applicatie in Python die het doorzoeken van al deze informatie vereenvoudigt.

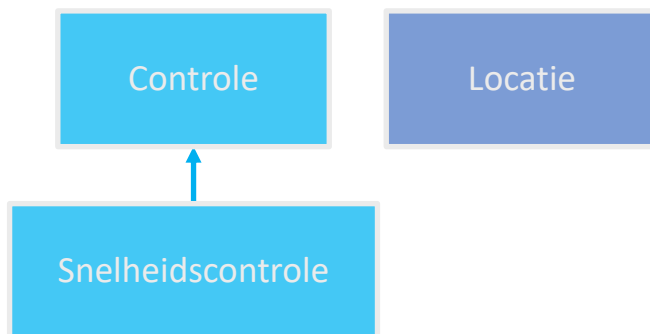
Stappenplan: je vetrekt van het bronmateriaal (terug te vinden op Leho). Je wijzigt de folderstructuur niet. Het bestand start.py mag niet gewijzigd worden, en moet vlekkeloos werken (nadat je de nodige referenties toegevoegd hebt).

De volgende klasse is gegeven:

- De klasse **Locatie**. Deze klasse houdt een straat, gemeente, latitude (breedtegraad) en longitude (lengtegraad) bij.

Maak volgende klassen aan:

- klasse **Controle** die een locatie (object van de vorige klasse; opgelet: hier is klasse-associatie van toepassing), een jaar en maand bijhoudt;
- klasse **Snelheidscontrole**, die erft van de klasse Controle, aanvullend het aantal controles, het aantal gecontroleerde voertuigen en het aantal overtredingen bijhoudt.



- Klasse **SnelheidscontroleRepository** die instaat voor het inlezen van het json-bestand, en kan zoeken/filteren in een list van objecten van de klasse Snelheidscontrole.

Voorzie de klassen Controle en Snelheidscontrole van de methodes `__init__()`, `__str__()` en de nodige property-methodes/setter-methodes.

Waar vermeld houd je rekening met overerving!

Je past op de juiste plaatsen **exception handling** toe:

- Is de property een geheel of komma getal? Raise dan een exception indien de value een foutief gegevenstype is.

Voeg volgende zaken aan de klasse Locatie toe:

- Objecten van de klasse Locatie moeten op hun gelijkheid kunnen getest worden. Indien straat en gemeente identiek zijn, dan hebben we te maken met een gelijkheid tussen beide objecten. Voeg hiervoor de juiste methode toe aan de klasse Locatie.
- Objecten van de klasse Locatie moeten kunnen gesorteerd kunnen worden. De sortering gebeurt op basis van de gemeente. Bij een zelfde gemeente wordt de straat vergeleken.

Specifiek voor de klasse Controle:

- Voorzie de klasse van de nodige private attributen:
 - o *locatie, maand, jaar*
- De constructor
 - o heeft 3 parameters: *locatie, jaar, maand*
- Maak voor elk attribuut een publieke property aan.
 - o Locatie en jaar hebben zowel een getter als een setter.
 - Voeg in de setter controle toe.
 - o De property maand heeft enkel een getter.
 - Deze zet het maandnummer om naar de maandnaam: bv: januari, februari, enz
- Objecten van de klasse Controle moeten op hun gelijkheid kunnen getest worden. Indien de locatie, maand en jaar identiek zijn, hebben we te maken met een gelijkheid tussen beide controles. Voeg hiervoor de juiste methode toe.
- Objecten van de klasse Controle moeten kunnen gesorteerd worden.
 - o [Enkel voor MCT:] De sortering gebeurt op basis van de locatie, jaar, maand (in deze volgorde).
 - o [Enkel voor MIT:] De sortering gebeurt op basis van de locatie.

Specifiek voor de klasse Snelheidscontrole:

- Deze klasse erft van Controle
- Voorzie de klasse van volgende extra private attributen:
 - o *aantal_controles, aantal_snelheidsovertredingen en aantal_gecontroleerde_voertuigen*
- De constructor
 - o heeft 6 parameters: *locatie, jaar, maand, aantal_controles, aantal_snelheidsovertredingen en aantal_gecontroleerde_voertuigen*
- Maak voor de attributen een publieke property aan. Doe telkens een controle of de waarde een geheel getal is en groter dan nul is.
- Voeg een extra get-property 'overtredingsgraad' toe dat de overtredingsgraad berekent & teruggeeft. De overtredingsgraad wordt bekomen door het aantal overtredingen te delen door het aantal gecontroleerde voertuigen maal 100.
- Houdt het totale aantal objecten van deze klasse bij. Voorzie ook de static methode 'geef_totaal_aantal_controles' om dat aantal terug te geven.
- Methode 'info()' die een string teruggeeft met uitgebreide info:
 Snelheidscontrole op ROGGEELAN te KORTRIJK in de maand augustus 2012:
 - aantal gevoerde controles: 1
 - aantal gecontroleerde voertuigen: 122
 - aantal overtredingen: 16
 - overtredingsgraad: 13.1%
- Objecten van de klasse Snelheidscontroles moeten kunnen gesorteerd worden.
 - o [Enkel voor MCT:] De sortering gebeurt op basis van de straat en de bijhorende overtredingsgraad (in deze volgorde)
 - o [Enkel voor MIT:] De sortering gebeurt op basis van de straat.

Specifiek voor de klasse SnelheidscontroleRepository:

- Een static methode 'inlezen_controles' die in staat is om het bronbestand in te lezen en een list van snelheidscontroles terug te geven. Verwerk hierin exceptionhandling. Doe waar nodig conversie van string naar float/int.
- een static methode 'zoek_deel_straat' met parameters een verzameling van objecten van de klasse Snelheidscontrole én een deel van de gezochte straatnaam. Geef een list met enkel deze controles uit de gezochte straat terug.
- [EXTRA VOOR MCT]
 Een static methode 'geef_gevaarlijke_straten' met parameters een verzameling van objecten van de klasse Snelheidscontrole en een minimum overtredingsgraad. Doorloop alle snelheidscontroles. Geef een lijst van unieke straatnamen terug waar de overtredingsgraad groter is dan het doorgegeven minimum.

Maak gebruik van het test-bestand start.py om de verschillende functionaliteiten uitvoerig te testen.