

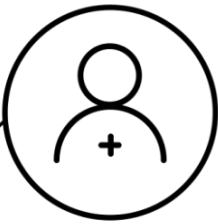


Eindopdracht

Hbo-bachelor ICT/Geo BI



Python Fundamentals (15 EC)



Opdrachtbeschrijving en deelopdrachten

Randvoorwaarden, structuur en beoordeling



NOVI
HOGESCHOOL

Inhoud

Integrale eindopdracht	3
Algemene opdrachtbeschrijving	3
Casus	4
Deelopdrachten	9
Deelopdracht 1. Plannen, analyseren en ontwerpen	9
Deelopdracht 2. Coderen en testen	9
Deelopdracht 3. Analyse uitvoeren en projectverslag schrijven	10
Randvoorwaarden	10
Beoordelingscriteria	11

Eindopdracht Python Fundamentals (15 EC)

Integrale eindopdracht

De leerlijn Python Fundamentals behandelt de belangrijkste basisprincipes van Python. Je doet basisvaardigheden op waarmee je repeterende werkzaamheden kunt automatiseren, handige scripts kunt schrijven en daarmee allerlei processen kunt vereenvoudigen.

Om deze leerlijn af te ronden dien je de volgende **leeruitkomst** aan te tonen:

De student past belangrijke basisprincipes van Python toe in een zelfgeschreven Python-applicatie, en automatiseert hiermee repeterende werkzaamheden en/of processen.

Algemene opdrachtbeschrijving

Voor deze eindopdracht ga je een Python (versie 3.9) applicatie ontwikkelen voor onderstaande casus. De nadruk ligt op het ontwerpen, bouwen en testen van de applicatie met de gewenste functionaliteiten. De benodigde invoerbestanden met een beschrijving vind je in Teams. In de code neem je (zinvol) commentaar op. Naast de code lever je een projectverslag op dat de volgende indeling heeft:

1. Plan van aanpak (overzicht activiteiten & planning)
2. Verantwoording, bestaande uit
 - Beschrijving en ontwerp van de analysefunctie
 - Ontwerp van de 2 zoekfuncties voor bedrijven
 - Toelichting gemaakte keuzes
 - Testverslag
3. Projectevaluatie

De sourcecode wordt als ZIP-bestand ingeleverd en het projectverslag als PDF-document.

Casus

Zorg voor de aarde en het klimaat krijgt steeds meer aandacht, van overheden, burgers en ook het bedrijfsleven. Bedrijven mogen tegenwoordig niet onbeperkt gassen uitstoten en betalen belasting voor hun uitstoot. Tevens kunnen ze boetes krijgen wanneer hun feitelijke en afgesproken uitstoot niet overeenkomen

Schone Lucht B.V. controleert in opdracht van de overheid of bedrijven zich houden aan de afgesproken uitstootwaarden van 4 verschillende gassen (CO_2 , CH_4 , NO_2 en NH_3). Dagelijks ontvangt de organisatie een bestand met (door een aardobservatiesatelliet) waargenomen concentraties van deze gassen in een gebied van 100 km bij 100 km (10.000 km^2). De concentratie van elk gas is bepaald per vierkante kilometer, dus het bestand bevat $100 \times 100 \times 4$ (gassen) = 40.000 meetwaarden. Voor alle bekende bedrijven wordt handmatig gecontroleerd of ze zich aan de gemaakte afspraken houden en als dat niet zo is wordt een boete berekend en opgelegd. Hierbij worden regelmatig vergissingen gemaakt. Een deel van de bedrijven wordt periodiek bezocht door inspecteurs die ter plaatse metingen doen en een bezoekrapport opmaken. Deze rapporten worden gearchiveerd, maar soms vergeet een inspecteur een rapport af te geven of wordt een rapport niet definitief gemaakt.

Recent heeft de directeur van Schone Lucht B.V. besloten dat er een Python applicatie ontwikkeld moet worden die de volgende functionaliteiten bevat:

1. Inlezen permanente gegevens
2. Tonen inspecteurs
3. Tonen bedrijven
4. Tonen bezoekrapporten
5. Inlezen metingen bestand
6. Plotten van meetgegevens
7. Bepalen en tonen boetes
8. Analyseren meetgegevens
9. Beheer bedrijfsgegevens
10. Beheer bezoekrapporten
11. Opslaan gewijzigde gegevens

Hieronder zijn de gewenste functionaliteiten gespecificeerd.

Specificatie

Na het starten van de applicatie moet een keuzemenu getoond worden waarmee de gebruiker een bepaalde functionaliteit kan uitvoeren. Je mag alle functies in 1 groot menu stoppen, maar kunt er ook voor kiezen om submenu's te gebruiken of functies te combineren in één menukeuze.

Zorg ervoor dat de applicatie niet crasht bij een ongeldige menukeuze. Het is niet nodig om een grafische interface voor het menu te maken; een eenvoudige opsomming met nummers voor de verschillende menukeuzes volstaat.

Een van de menukeuzes is het (gecontroleerd) beëindigen van de applicatie.

1. Inlezen permanente gegevens

De gegevens van inspecteurs, bedrijven en bezoeken zijn in tekstbestanden opgeslagen en moeten na het starten van de applicatie worden ingelezen. De ingelezen gegevens worden geconverteerd naar objecten van representatieve klassen. Je mag zelf kiezen of dit automatisch gebeurt of dat de gebruiker dit (per bestand) opstart via een menufunctie.

Details van de gegevens kun je vinden in de toelichting van de invoerbestanden. Denk aan het toevoegen van de juiste controles bij het inlezen en laat de applicatie niet crashen als een bestand niet aanwezig is of (technisch) onjuiste gegevens bevat.

2. Tonen inspecteurs

Bij deze functie kan de gebruiker een overzicht van de gegevens van inspecteurs op het scherm tonen. Dit overzicht moet gegenereerd worden vanuit de inspecteur objecten en niet vanuit het tekstbestand (om te voorkomen dat gegevens meerdere keren worden ingelezen!).

3. Tonen bedrijven

Bij deze functie kan de gebruiker een overzicht van de gegevens van bedrijven op het scherm tonen. Dit overzicht moet gegenereerd worden vanuit de bedrijfsobjecten en eventuele wijzigingen die door de gebruiker en/of de applicatie zijn ingevoerd moeten correct getoond worden.

4. Tonen bezoekerapporten

Bij deze functie kan de gebruiker een tweetal overzichten laten genereren door het systeem. Het betreft de bezoekerapporten per inspecteur of per bedrijf. Van elk bezoekerapport wordt de datum van het bezoek, de datum van het rapport, de opmerkingen en de status getoond. Afhankelijk van het type overzicht komen daar nog enkele gegevens bij. Deze overzichten moeten gegenereerd worden vanuit de rapportobjecten en eventuele wijzigingen die door de gebruiker en/of de applicatie zijn ingevoerd moeten correct getoond worden.

Overzicht bezoekerapporten per inspecteur

De gebruiker voert de code van een inspecteur in en optioneel een begin- en einddatum. Het systeem controleert of die code bestaat en of de datums geldig zijn (als ze zijn ingevuld). Vervolgens worden alle bezoekerapporten van deze inspecteur waarvan de bezoekdatums binnen de geselecteerde periode vallen, aflopend gesorteerd op datum, op het scherm getoond. Naast bovengenoemde gegevens moet per bezoekerapport ook de naam van het bezochte bedrijf worden getoond.

Overzicht bezoekerapporten per bedrijf

De gebruiker selecteert een bedrijf met één van de twee zoekfuncties voor bedrijven (zie functionaliteit 5, beheer bedrijfsgegevens). Vervolgens worden alle bezoekerapporten van dit bedrijf, aflopend gesorteerd op datum, op het scherm getoond. Naast bovengenoemde gegevens moet per bezoekerapport ook de naam van de inspecteur die het rapport heeft opgesteld worden getoond.

5. Inlezen metingenbestand

Met deze functie kan de gebruiker het metingenbestand inlezen. De data staat in een csv-bestand dat naar één of meerdere NumPy arrays moet worden geconverteerd. Denk weer aan het toevoegen van de juiste controles.

Daarna moet de totale uitstoot CO₂-equivalent per km² (dus voor elke x- en y-waarde) worden bepaald. Dit is een gewogen uitstoot omdat het ene gas vervuilerder is dan het andere. De berekening gaat als volgt:

$$\text{Totale uitstoot} = c1 \times \text{uitstoot CO}_2 + c2 \times \text{uitstoot CH}_4 + c3 \times \text{uitstoot NO}_2 + c4 \times \text{uitstoot NH}_3$$

Voor deze opdracht geldt: $c1 = 1$, $c2 = 25$, $c3 = 5$ en $c4 = 1000$.

De wegingsfactoren $c1$ tot en met $c4$ worden als constanten in de applicatie vastgelegd.

6. Plotten van meetgegevens

Met deze functie kan de gebruiker een plot maken van de meetgegevens per gas. De gebruiker kiest een van de 4 gassen waarvan de uitstoot is gemeten en het systeem maakt een plot van de waarden voor alle coördinaten. Maak hiervoor gebruik van de functionaliteit van het Matplotlib package.

De gebruiker moet ook een plot kunnen maken van de totale uitstoot CO₂-equivalent.

7. Bepalen en tonen boetes

Met deze functie wordt voor elk bekend bedrijf berekend wat de omgevingsuitstoot is. Daarbij wordt de totale uitstoot op de locatie van het bedrijf volledig meegeteld, op de 8 aanliggende locaties voor de helft en op de 16 locaties daaromheen voor een kwart. De omgevingsuitstoot is dus een gewogen gemiddelde op een 5 km bij 5 km vierkant, waarbij het bedrijf in het middelste vierkant van 1 km² staat.

Vervolgens wordt per bedrijf gecontroleerd of de totale berekende uitstoot minder is dan de maximale afgesproken uitstoot. Als een bedrijf meer heeft uitgestoten dan afgesproken wordt een boete opgelegd die evenredig is met de overschrijding. De boete per eenheid overschrijding is vastgelegd als een constante in de applicatie en is voor deze opdracht € 1.000,--. Boetes worden afgerond op hele euro's.

Tenslotte moet de berekende omgevingsuitstoot en de eventuele boete van elk bedrijf in het corresponderende bedrijfsobject worden vastgelegd.

N.B. De afstand tussen 2 bedrijven is minimaal 5 km en alle bedrijven liggen minimaal 2 km van de rand af. De omgevingsvierkanten van 2 bedrijven overlappen elkaar dus nooit!

8. Analyseren meetgegevens

Met deze functie wordt de gebruiker ondersteund bij het opsporen van onbekende vervuilende bedrijven. De functie moet onderzoeken of er locaties zijn met een hoge concentratie van gassen waar nog geen bedrijf is geregistreerd. Het metingenbestand bevat minimaal 1 zo'n locatie en de functie moet de coördinaten van deze locatie en de gemeten concentraties tonen.

In de verantwoording moet functioneel beschreven worden hoe het programma deze analyse uitvoert en moet een ontwerp van het zoekalgoritme worden opgenomen (bijvoorbeeld m.b.v. een PSD).

9. Beheer bedrijfsgegevens

Bij deze functie moet de gebruiker de gegevens van een nieuw bedrijf kunnen invoeren of de gegevens van een bestaand bedrijf kunnen wijzigen. Het betreft: de unieke code, de naam en het adres (straat, postcode en plaats), de locatie (de coördinaten) en de maximaal toegestane totale uitstoot. Er zijn enkele bedrijven die ook regelmatig door een inspecteur worden bezocht en voor deze bedrijven moet tevens de frequentie (het geplande aantal bezoeken per jaar, dit is een geheel getal tussen 1 en 12) en de naam van de contactpersoon worden vastgelegd en kunnen worden gewijzigd.

Van een bestaand bedrijf kan de code en de locatie niet gewijzigd worden. Je moet bij een nieuw bedrijf kunnen aangeven of het wel of niet geïnspecteerd wordt, en de status van een bestaand bedrijf moet van 'niet geïnspecteerd' naar 'wel geïnspecteerd' kunnen wijzigen. Het omgekeerde mag niet.

Denk weer aan het toevoegen van de juiste controles (de postcode bestaat bijvoorbeeld uit 4 cijfers en 2 letters en de locatie moet binnen het 100 bij 100 grid liggen).

Als je de gegevens van een bestaand bedrijf wil wijzigen moet je het bedrijf kunnen vinden via één van de 2 zoekfuncties: zoeken op locatie en zoeken op (een deel van de) naam.

Zoeken van een bedrijf in de buurt van een locatie

De gebruiker voert een locatie in (een tweetal coördinaten) en het systeem geeft de code van het bedrijf dat het dichtst bij die locatie ligt terug. De afstand tussen de ingevoerde locatie en de locatie van een bedrijf is het minimale aantal verplaatsingen tussen beide locaties. Als je van een locatie naar één van de 8 aangrenzende locaties gaat (dus recht of schuin) is dat 1 verplaatsing. Als er twee of meer bedrijven zijn die op dezelfde afstand liggen van de ingevoerde locatie wordt de eerste die gevonden is geselecteerd.

Zoeken van een bedrijf met (een deel van) de naam

De gebruiker voert (een deel van) de naam van een bedrijf in en het systeem geeft de code van het bedrijf waarvan de naam (gedeeltelijk) overeenkomt terug. Als er geen overeenkomst is toont het systeem een melding 'niet gevonden'. Als er twee of meer bedrijven zijn waarvan de naam (gedeeltelijk) overeenkomt wordt de eerste die gevonden is geselecteerd.

10. Beheer bezoekerapporten

Bij deze functie moet de gebruiker een nieuw bezoekerapport kunnen aanmaken of de gegevens van een bestaand bezoekerapport kunnen wijzigen. Het betreft: de code van de inspecteur, de code van het bedrijf, de (geplande) bezoeksdatum, de datum dat het rapport is opgesteld, de opmerkingen en de status ('voorlopig' of 'definitief').

Van een bestaand bezoekerapport kunnen de code van de inspecteur en de code van het bedrijf niet gewijzigd worden en als het rapport definitief is gemaakt kan er niets meer gewijzigd worden. Bij een

nieuw rapport hoeven de datum van het rapport en de opmerkingen nog niet te zijn ingevuld en wordt de status altijd 'voorlopig'. Het rapport kan alleen 'definitief' worden gemaakt als alle gegevens zijn ingevoerd. Denk weer aan het toevoegen van de juiste controles (de datum van het rapport mag bijvoorbeeld niet vóór de bezoeksdatum liggen).

11. Opslaan gewijzigde gegevens

Voordat het programma wordt afgesloten moeten alle nieuwe en gewijzigde bedrijfsgegevens en nieuwe en gewijzigde bezoekrapporten permanent worden opgeslagen in tekstbestanden, zodat ze de volgende dag weer kunnen worden ingelezen door de applicatie (de gegevens van de inspecteurs kunnen niet gewijzigd worden en het tekstbestand met hun gegevens hoef je dus niet te updaten). Je mag zelf kiezen of je hiervoor aparte menukeuzen maakt of het wegschrijven automatisch laat doen als je de menukeuze 'beëindigen applicatie' kiest.

Deelopdrachten

In de eerste deelopdracht tref je de voorbereidingen om de applicatie te gaan bouwen. In de tweede deelopdracht programmeer en test je de applicatie. Afsluitend schrijf je het projectverslag, waarin je onder andere het project evalueert en de door jou gemaakte keuzes bespreekt.

Deelopdracht 1. Plannen, analyseren en ontwerpen

Op basis van de eerdergenoemde casus is het handig om eerst een beeld te creëren van hoe je aan de slag zal gaan. Dit ga je als volgt doen:

- Allereerst maak je een plan van aanpak
- Vervolgens bedenk je hoe je de meetgegevens kunt analyseren
- Vervolgens maak je een ontwerp van de belangrijkste onderdelen

Hieronder lees je omschrijving van iedere stap uit bovenstaande stappenplan:

Maken Plan van aanpak

Het Plan van aanpak bevat een overzicht en een planning van uit te voeren activiteiten. Het is niet de bedoeling dat er een compleet plan van aanpak wordt opgeleverd (met probleembeschrijving, opdrachtomschrijving, risicoanalyse, etc.). Onderdelen die aanwezig moeten zijn in je plan van aanpak zijn: *1. Een overzicht van activiteiten & 2. Planning van de activiteiten*. N.B. Dit kan in 1 tabel!

Analyse meetgegevens

Bedenk hoe je een onbekend vervuילend bedrijf kan opsporen met behulp van de meetgegevens.

Ontwerpen applicatie

Maak op basis van de systeemeisen een ontwerp voor de menustructuur. Bedenk welke functies je wilt combineren en welke functies automatisch (dus zonder tussenkomst van de gebruiker) worden uitgevoerd. Maak een ontwerp voor de analysefunctie van de meetgegevens en de 2 zoekfuncties voor bedrijven (bijvoorbeeld m.b.v. een PSD).

Bedenk welke (samengestelde) datatypen geschikt zijn om bepaalde functionaliteiten te realiseren.

Op te leveren:

- Plan van aanpak
- Beschrijving en ontwerp van de analysefunctie
- Ontwerp van de 2 zoekfuncties voor bedrijven

Deelopdracht 2. Coderen en testen

Coderen

Als de specificatie duidelijk is en de ontwerpen gemaakt kan er gestart worden met de ontwikkeling van de Python applicatie. Probeer dit zo gestructureerd mogelijk aan te pakken door eerst te bekijken welke onderdelen separaat kunnen worden gecodeerd en getest.

Denk aan het gebruik van zinvolle namen voor de variabelen, functies, klassen en methoden, het toevoegen van zinvol commentaar en anticipeer op hergebruik van bepaalde onderdelen.

Je mag in jouw code Engelse of Nederlandse variabele- en functienamen gebruiken, zolang je maar consistent bent in het gebruik. Ook zorg je voor duidelijk en voldoende commentaar in de code. In de verantwoording kun je gemaakte keuzes toelichten.

Testen

Tijdens en na de ontwikkeling van de applicatie dient de werking getest te worden. In het testverslag (onderdeel van de verantwoording) beschrijf je hoe je dit hebt gedaan. Je kunt dit bijvoorbeeld doen door de testresultaten toe te voegen aan de testgevallen in het (vooraf gemaakte) testplan.

Op te leveren:

- De python sourcecode van de applicatie
- Toelichting op de gemaakte keuzes
- Testverslag

Deelopdracht 3. Projectverslag schrijven

Verantwoording

Je neemt het plan van aanpak en alle verplichte onderdelen van de verantwoording op in het projectverslag. Tevens neem je de gerealiseerde begin- en einddatums van de activiteiten uit je plan van aanpak op en geef je een toelichting op eventuele verschillen. Vermeld ook het aantal uren dat je aan de totale opdracht hebt besteed.

Projectevaluatie

Aan het eind evalueer je het project. Je vergelijkt de uitvoering met de planning en geeft aan welke functies makkelijker/moeilijker waren dan gedacht. Uiteraard mag je ook onderdelen noemen waar je trots op bent en/of zaken die je in de toekomst anders zou aanpakken. De evaluatie wordt als laatste hoofdstuk opgenomen in het projectverslag (voor indeling zie Algemene opdrachtbeschrijving).

Op te leveren:

- Projectverslag

Randvoorwaarden

Hieronder vind je een aantal randvoorwaarden waaraan je code/programma en verslaglegging moet voldoen. Deze randvoorwaarden hebben een verplicht karakter.

- Beperk spellings- en grammaticale fouten tot een absoluut minimum, laat je werk nog een keer door iemand anders nalezen voor je het inlevert.
- Draag zorg voor een lezersvriendelijke lay-out (gebruik bv. paginanummering).
- Vermeld je naam en studentnummer duidelijk op de voorpagina van je verslag.
- De omvang van het verslag beslaat maximaal **2.000 woorden** (exclusief inhoudsopgave, samenvatting, inleiding en eventuele bijlagen).
- De applicatie moet ontwikkeld worden met behulp van Python 3.9.

Beoordelingscriteria

De eindopdracht wordt beoordeeld op basis van de volgende beoordelingscriteria. Per criterium kent de beoordelaar een aantal punten toe (cijfer 1-10). De weging van deze cijfers staat vermeld in de tabel. Het eindcijfer voor de opdracht als geheel is het gewogen gemiddelde van de cijfers per criterium en dient ten minste 5,5 te zijn om deze opdracht af te ronden.

Criterium nummer	Leeruitkomsten en beoordelingscriteria	Weging
	De student past belangrijke basisprincipes van Python toe in een zelfgeschreven Python-applicatie, en automatiseert hiermee repeterende werkzaamheden en/of processen.	
Criterium 1	De student schrijft een deugdelijk plan van aanpak dat bestaat uit een overzicht van de activiteiten en de planning ter voorbereiding op het systematisch programmeren.	5%
Criterium 2	De student maakt ontwerpen (bijvoorbeeld PSD's) voor de analysefunctie en de 2 zoekfuncties om een bedrijf te selecteren.	20%
Criterium 3	De student maakt een Python applicatie, waarin alle gewenste functionaliteiten zijn opgenomen.	50%
Criterium 4	De student test de applicatie en verwerkt de resultaten in een testverslag.	10%
Criterium 5	De student schrijft een begrijpelijke verantwoording waarin de gemaakte keuzes helder worden toegelicht.	10%
Criterium 6	De student evalueert het project waarbij wordt gereflecteerd op de moeilijkheden en onderdelen die goed gingen tijdens het project.	5%
	Totaal	100%