

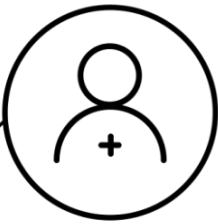


Eindopdracht

*Hbo-bachelor ICT/Geo BI
Versie 3.0*



Python Fundamentals (15 EC)



*Opdrachtbeschrijving en deelopdrachten
Randvoorwaarden, structuur en beoordeling*



NOVI
HOGESCHOOL

Inhoud

Integrale eindopdracht	
Algemene opdrachtbeschrijving	3
Casus	4
Deelopdrachten	7
Deelopdracht 1. Plannen	7
Deelopdracht 2. Coderen	7
Deelopdracht 3. Project evaluatie	8
Quicksan	9
Beoordelingscriteria	10

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 allerhande 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 van de applicatie met de gewenste functionaliteiten. De applicatie wordt ontwikkeld op basis van de casusgegevens die je terugvindt in teams. In de code neem je (zinnig) commentaar op. Naast de code lever je een projectverslag op dat de volgende indeling heeft:

1. Plan van aanpak
2. Analyse van meetgegevens
3. Projectevaluatie
4. Verantwoordingsdocument

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 onbepaald gas 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 bij 100 km (10.000km^2). De concentratie van elk van de 4 gassen is bepaald per vierkante kilometer, dus het bestand bevat in totaal $100 \times 100 \times 4$ (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 gearhiveerd, 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. Inlezen metingen bestand
3. Analyseren meetgegevens
4. Tonen bezoekrapporten

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 je 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. Je mag zelf kiezen of dit automatisch gebeurt of dat de gebruiker dit (per bestand) opstart via een menufunctie.

Een inspecteur heeft een unieke code, een naam en een standplaats. De gegevens van inspecteurs hoeven niet gewijzigd te kunnen worden.

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. Inlezen metingenbestand

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

Voor elk bekend bedrijf wordt berekend wat de totale uitstoot CO₂-equivalent is, omdat het ene gas vervuilerder is dan het ander. Daarbij worden de concentraties 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 uitstoot per gas wordt dus berekend als een gewogen gemiddelde op een 5-bij-5 vierkant, waarbij het bedrijf in het midden staat. De totale uitstoot op een coördinaat kan berekend worden met deze formule:

$$\text{Totale uitstoot} = c1 \times \text{uitstoot gas1} + c2 \times \text{uitstoot gas2} + c3 \times \text{uitstoot gas3} + c4 \times \text{uitstoot gas4}$$

De wegingsfactoren c1 tot en met c4 zullen als constanten in de code staan.

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.

Voor het berekenen van de uitstoot van een bedrijf en het maken van een 5-bij-5 vierkant kan het volgende gedaan worden:

- Maak een grid van 100X00
- Vul de meetwaarden in
- Bereken de totale uitstoot over een 5x5 grid van elk bedrijf.

3. 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 het analyserapport moet functioneel beschreven worden hoe het programma deze analyse uitvoert

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.

Overzicht bezoekenrapporten 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 bezoekenrapporten van deze inspecteur waarvan de bezoeksdatum binnen de geselecteerde periode vallen, aflopend gesorteerd op datum, op het scherm getoond. Naast bovengenoemde gegevens moet per bezoekenrapport ook de naam van het bezochte bedrijf worden getoond.

Overzicht bezoekenrapporten per bedrijf

De gebruiker selecteert een bedrijf met één van de twee zoekfuncties voor bedrijven. Vervolgens worden alle bezoekenrapporten van dit bedrijf, aflopend gesorteerd op datum, op het scherm getoond. Naast bovengenoemde gegevens moet per bezoekenrapport ook de naam van de inspecteur die het rapport heeft opgesteld worden getoond.

Deelopdrachten

In de eerste deelopdracht tref je de voorbereidingen om de applicatie te gaan bouwen. In de tweede deelopdracht programmeer 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

Het Plan van aanpak bevat een gedegen omschrijving van jouw interpretatie van de casus en welke stappen je gaat ondernemen om de applicatie te ontwikkelen die de vereiste functionaliteiten bevatten. Het is niet de bedoeling dat er een compleet plan van aanpak wordt opgeleverd (met probleembeschrijving, opdrachtoomschrijving, risicoanalyse, etc.). Onderdelen die aanwezig moeten zijn in je plan van aanpak zijn: *1. Een omschreven interpretatie van de casus & 2. Een omschrijving van de aanpak.*

Op te leveren:

- Plan van aanpak (overzicht activiteiten & planning)

Verantwoordingsdocument:

- In je verantwoordingsdocument schrijf je een verantwoording voor de 1^e deelopdracht. Denk daarbij aan een toelichting van gemaakte keuzes en planning.

Deelopdracht 2. Coderen

Coderen

Zodra je plan van aanpak is geschreven, ga je de specificaties (de genoemde functionaliteiten in de casus) voor de python applicatie ontwikkelen volgens de geldende richtlijnen. Pak dit zo gestructureerd mogelijk aan door eerst te bekijken welke onderdelen separaat kunnen worden gecodeerd. Denk aan het gebruik van zinvolle namen voor de variabelen, functies, klassen en methoden. Voeg zinvol commentaar toe en anticipeer op hergebruik van bepaalde onderdelen. Daarna ga je één van de handelingen automatiseren.

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.

Op te leveren:

- De python sourcecode van de applicatie

Verantwoordingsdocument:

- In het verantwoordingsdocument beargumenteer je de keuze voor de wijze waarop je gecodeerd hebt, de keuze voor bepaalde variabelen, functies, klassen, methoden en keuzes in het automatiseren van een handeling. Daarnaast geef je een toelichting over de waarde van jouw applicatie; wat doet het, waarvoor dient het en hoe verhoudt dit zich tot de casus?

Deelopdracht 3. Project evaluatie

Evaluatie en verslag

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:

- Projectevaluatie

Quickscan

Hieronder vind je de randvoorwaarden waar de eindopdracht aan moet voldoen. De beoordelaar kan op basis van deze eisen de eindopdracht teruggeven en zal deze pas nakijken als aan de eisen is voldaan.

Algemene eisen
Het projectverslag is ingeleverd als PFD
Het document bevat geen bronnen of verwijzingen buiten het document (linkjes etc.)
De eindopdracht is goed leesbaar zonder storende aanwezigheid van grammatica- en spellingsfouten.
De sourcecode is aangeleverd d.m.v. een ZIP-bestand van maximaal 50 MB. (geen .rar)
De omvang van het verslag beslaat maximaal 2.000 woorden(exclusief inhoudsopgave, samenvatting, inleiding, bibliografie en eventuele bijlagen).
Indien je een herkansing inlevert na het krijgen van feedback, is het ingevulde 'Template herkansingsfeedback' bijgevoegd.
Inhoudelijke eisen
Alle deelopdrachten zijn uitgewerkt en de gevraagde deelproducten zijn aanwezig.
De applicatie is geprogrammeerd met Python
De applicatie start op zonder te crashen.

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 minst 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 logische omschrijving/interpretatie van de casus en de geplande stappen ter voorbereiding op het systematisch programmeren.	10%
Criterium 2	De student maakt correct gebruik van controlestructuren, functies en modules en gebruiker input wanneer de applicatie hierom vraagt.	30%
Criterium 3	De student ontwikkelt een werkend algoritme/script voor het vinden van het niet geregistreerde bedrijf.	30%
Criterium 4	De student schrijft een begrijpelijke verantwoording waarin de gemaakte keuzes helder worden toegelicht.	20%
Criterium 5	De student evalueert het project waarbij wordt gereflecteerd op de moeilijkheden en onderdelen die goed gingen tijdens het project.	10%
	Totaal	100%