

Novi Python Fundamentals

Eindopdracht: Schone Lucht B.V.



Naam: Hayk Abrahamian

Docent: Rein Fernhout

Woorden: 1998 (ex-inhoudsopgave en inleiding)

Inleverdatum: 30-04-2024

## Inhoudsopgave

1. Inleiding	3
2. Plan van aanpak	3
a. Omschrijving/Interpretatie van de casus	3
b. Omschrijving van de aanpak	5
c. Verantwoordingsdocument 1e deelopdracht	8
3. Projectevaluatie	9
4. Verantwoordingsdocument	10

## Inleiding

In de integrale eindopdracht van de leerlijn Python Fundamentals worden vaardigheden toegepast waarmee een Python-applicatie ontwikkeld kan worden. Het doel van deze integrale eindopdracht binnen de Python Fundamentals leerlijn is het creëren van een Python-applicatie die de kernprincipes van Python benut om repetitieve taken en processen te automatiseren. Dit omvat ook het begrijpen en toepassen van concepten zoals bestandsverwerking, werken met datastructuren en implementeren van foutcontrolemechanismen.

Onder de eindopdracht valt ook het opstellen van een plan van aanpak, analyse van meetgegevens, projectevaluatie en verantwoordingsdocument. Hiermee wordt inzicht geboden in de aanpak en de gemaakte keuzes tijdens het ontwerpen en bouwen van de applicatie. Los van het schrijven van code, moet er ook beargumenteerd worden voor de keuzes die er gemaakt zijn en het documenteren van de codes met zinvolle opmerkingen om de leesbaarheid en onderhoudsvriendelijkheid te verbeteren.

### 1. Plan van aanpak

#### **Omschrijving/Interpretatie van de casus**

In deze Python eindopdracht duiken wij in de wereld van milieuregulaties en gaan wij aan de slag voor Schone Lucht B.V. die ingeschakeld wordt om de emissie-waarden van grote bedrijven te meten/onderzoeken. Het doel is om te controleren of bedrijven zich houden aan de afgesproken uitstootwaarden van vier gassen (CO<sub>2</sub>, CH<sub>4</sub>, NO<sub>2</sub> en NH<sub>3</sub>). Als bedrijven de afgesproken uitstoot overschrijden dan worden ze beboet. Schone Lucht B.V. geeft aan ons de opdracht om een applicatie te ontwikkelen die de controle op de uitstoot van gassen kan automatiseren en vereenvoudigen. De applicatie moet een gebruiksvriendelijke interface hebben waarin de gebruiker verschillende functionaliteiten kan selecteren, zoals het inlezen van gegevens, het analyseren van meetgegevens en het tonen van rapporten.

Kortom, de directeur van Schone Lucht B.V. heeft besloten dat wij een Python-applicatie moeten ontwikkelen met de volgende functionaliteiten:

1. Inlezen van permanente gegevens zoals inspecteurs, bedrijven en bezoekerapporten.

Het inlezen van gegevens van inspecteurs, bedrijven en bijbehorende bezoekerapporten vanuit tekstbestanden naar de applicatie. Hierbij worden de gegevens gecontroleerd om ervoor te zorgen dat ze correct worden verwerkt en geen fouten veroorzaken in de

applicatie. De gebruiker zal een foutmelding krijgen wanneer de gegevens niet op een correcte wijze worden geupload.

## 2. Inlezen van meetgegevens uit een CSV-bestand.

Hierbij worden meetgegevens van gassen ingelezen vanuit een CSV-bestand. De gegevens worden omgezet naar een geschikt formaat, zoals een NumPy-array, om verdere verwerking mogelijk te maken. Deze gegevens worden gebruikt om de uitstoot van verschillende gassen en het identificeren van mogelijke vervuilende bedrijven te analyseren.

## 3. Analyseren van de meetgegevens om onbekende vervuilende bedrijven op te sporen.

Deze functie omvat het analyseren van de meetgegevens om locaties te identificeren waar hoge concentraties van gassen worden gedetecteerd, maar waar nog geen geregistreerde bedrijven zijn. Dit kan helpen bij het opsporen van mogelijke vervuilende bedrijven die nog niet bekend zijn bij de autoriteiten.

## 4. Tonen van bezoekerapporten per inspecteur of per bedrijf.

Deze functionaliteit stelt gebruikers in staat om bezoekerapporten op te vragen op basis van inspecteurs of bedrijven. Hierbij worden details van de bezoekerapporten, zoals de datum van het bezoek, de datum van het rapport, opmerkingen en status, getoond. Dit biedt een overzicht van inspecties die zijn uitgevoerd en de bevindingen die zijn gerapporteerd.

## Omschrijving van de aanpak

Om de Python-applicatie voor de integrale eindopdracht van de Python Fundamentals leerlijn succesvol te voltooien, is het belangrijk om een stapsgewijze aanpak op te stellen. Er is gekozen om de code in het Engels te schrijven maar dat de opmerkingen in de code in het Nederlands worden beschreven. De belangrijkste basisonderdelen om deze applicatie neer te zetten zijn het gebruik maken van:

- Versie: Python 3.9
- Unit-testen
- Github
- Downloaden van de nodige packages: os, numpy, matplotlib
- Importeren van modules/packages
- Uploaden en analyseren van de bestanden
- Constanten opstellen
- Functies bepalen
- Klassen opstellen
- Variabelen en methoden zorgvuldig formuleren

Hieronder is het plan van aanpak beschreven:

### Stap 1: Analyse van de casus

Het eerste en belangrijkste onderdeel van het project is het grondig analyseren van de casus. Dit omvat het begrijpen van de vereisten van de applicatie, wat zijn de minimale voorwaarden en hoe baken je de opdracht af.

- Casus, deelopdrachten, QuickScan en beoordelingscriteria doornemen
- Schrijven van plan van aanpak
- Planning

### Stap 2: Verzamelen van benodigde informatie

Na het analyseren van de casus is het belangrijk om alle benodigde informatie te verzamelen. Dit omvat onder andere het bestuderen van de geleverde documentatie en het begrijpen van de gegevensstructuren.

- Hoe zijn de geleverde gegevens zoals bedrijven.txt, inspecteurs.txt, bezoekrapporten en gassen.csv gestructureerd?
- Op welke wijze kunnen deze gegevens in het programma geüpload worden?
- Op welke wijze kunnen de meegeleverde modules (menu, metingen en inspecteurs) toegepast worden en zijn er efficiëntere manieren om te coderen?

### **Stap 3: Ontwerp van de applicatie**

Op basis van de verzamelde informatie wordt een ontwerp voor de applicatie gemaakt. Dit omvat het identificeren van de verschillende modules, functionaliteiten die nodig zijn en het opstellen van een structuur voor de code.

- Welke modules moeten in het menu verwerkt worden en hoe kun je het beste via het menu doorgestuurd worden naar deze modules?
- Welke functies, klassen en definities moeten er gebruikt worden zodat er door middel van Python codes berekeningen gemaakt kunnen worden
- Op welke wijze zullen er overzichten opgesteld worden van inspecteurs
- Welke formules gaan er gebruikt worden om emissies te berekenen en dit ook te kunnen koppelen aan bedrijf/boete/locatie/onderzoek onbekende bedrijven?

### **Stap 4: Implementatie van de applicatie**

Na het voltooien van het ontwerp begint de eigenlijke ontwikkeling van de applicatie. Dit omvat het schrijven van de code voor elke module en functionaliteit, het testen van de code om fouten op te sporen en het aanbrengen van eventuele correcties of aanpassingen.

- Samenhang van verschillende modules bepalen en daarmee de eerste codes schrijven door de modules en functionaliteiten te importeren
- Functies schrijven die overzicht kunnen geven van bedrijven en inspecteurs maar ook matplotlib gebruiken om locaties van de hoge emissiewaardeplaatsen te tonen
- Een aparte menu opstellen om de bezoekrapporten per bedrijf/inspecteur te kunnen checken
- Functies/formules schrijven die de boete kunnen berekenen en ook een lijst kunnen geven van locaties waar hoge emissiewaarden aanwezig zijn (om zo te kunnen onderzoeken of er onbekende vervuilende bedrijven zijn).

### **Stap 5: Testen en debuggen**

Zodra de implementatie is voltooid, wordt de applicatie grondig getest om ervoor te zorgen dat deze correct werkt en voldoet aan de vereisten. Het gaat hier voornamelijk om unit-testen.

- Er moet eerst code geschreven worden die de gebruiker erop wijst wanneer er foute invoer wordt gegeven of dat de gegevens niet op de juiste wijze zijn geüpload
- In welke delen van de code zijn de plaatsen die het meest gevoelig zijn voor fouten, dit moet uitgebreid getest worden
- Alle opties en functionaliteiten moeten langsgegaan worden om vast te stellen dat het programma niet crasht en naar behoren werkt

- Zo veel mogelijk bugs uit het programma halen

## Stap 6: Documentatie en rapportage

Na het succesvol testen van de applicatie wordt alle code gedocumenteerd, inclusief commentaar en documentatie voor gebruikers. Daarnaast stel je nog projectevaluatie en verantwoordingsdocument op.

- ReadMe opstellen zodat gebruikers weten hoe ze zelfstandig het programma kunnen starten
- Commentaar in het programma verwerken zodat de functies en de werkingen duidelijk toegelicht kunnen worden
- Terugkijken naar je ervaring en projectevaluatie opstellen
- Verantwoordingsdocument schrijven om je keuzes te verantwoorden

**Planning** Het bovenstaande proces wordt verdeeld over 4/5 weken, waarbij elke week een van de stap(pen) afgerond moet worden.

Week 1	<ul style="list-style-type: none"> <li>• Analyse van de casus</li> <li>• Verzamelen van benodigde informatie.</li> <li>• Ontwerp van de applicatie</li> </ul>
Week 2	<ul style="list-style-type: none"> <li>• Implementatie van de applicatie</li> </ul>
Week 3	<ul style="list-style-type: none"> <li>• Implementatie van de applicatie</li> </ul>
Week 4	<ul style="list-style-type: none"> <li>• Laatste implementaties doorvoeren</li> <li>• Testen van de code</li> </ul>
Week 5	<ul style="list-style-type: none"> <li>• ReadMe opstellen</li> <li>• Documentatie schrijven</li> <li>• Projectevaluatie</li> <li>• Verantwoordingsdocument</li> <li>• Laatste test uitvoeren</li> <li>• Indiening van de opdracht.</li> </ul>

## Verantwoordingsdocument 1e deelopdracht

Tijdens de eerste deelopdracht van de integrale eindopdracht voor de Python Fundamentals leerlijn heb ik de volgende keuzes gemaakt en stappen ondernomen:

## Omschrijving van de casusinterpretatie:

Bij het interpreteren van de casus van Schone Lucht B.V. heb ik de nadruk gelegd op het identificeren van de belangrijkste functionele vereisten, zoals het controleren van de uitstoot van bedrijven, het analyseren van meetgegevens en het tonen van bezoekerapporten. Door de casus zorgvuldig te analyseren, kon ik een duidelijk beeld krijgen van de functionaliteiten die de Python-applicatie moest hebben.

## Omschrijving van de aanpak:

Mijn aanpak bestond uit verschillende stappen om de ontwikkeling van de Python-applicatie te plannen en te organiseren:

1. Analyse van de casus: Ik heb de casus grondig geanalyseerd om de functionele vereisten te begrijpen en te identificeren welke functionaliteiten nodig waren voor de applicatie.
2. Planning: Op basis van de casusanalyse heb ik een plan van aanpak opgesteld waarin ik de benodigde activiteiten en een tijdlijn voor de ontwikkeling van de applicatie heb vastgelegd.
3. Keuze van tools en technieken: Ik heb besloten om Python (versie 3.9) te gebruiken voor de ontwikkeling van de applicatie, samen met de NumPy-bibliotheek voor gegevensverwerking en Matplotlib voor de visuele weergave.
4. Selectie van functionaliteiten: Ik heb de gewenste functionaliteiten uit de casus gespecificeerd en bepaald hoe deze in de applicatie zouden worden geïmplementeerd.
5. Risicobeoordeling: Ik heb potentiële risico's geïdentificeerd en kwam die vooral tegen tijdens het testen. De foutmeldingen en instructies zijn toegevoegd zodat de gebruiker zonder een derde weet hoe hij het programma moet gebruiken.

Door deze aanpak kon ik een goede basis leggen voor de ontwikkeling van de Python-applicatie, waarbij ik de nadruk legde op het begrijpen van de casus, het plannen van de activiteiten en het identificeren van mogelijke risico's.

## 3. Projectevaluatie

Tijdens de ontwikkeling van het Python-project voor de integrale eindopdracht van de Python Fundamentals leerlijn heb ik verschillende ervaringen opgedaan en waardevolle lessen geleerd.

**Vergelijking met de planning:** Over het algemeen verliep alles redelijk volgens het plan maar omdat ik hiernaast ook nog bezig ben met het afronden van Fronted, heb ik hier en daar wel wat vertragingen opgelopen. Ik denk dat ik wel geslaagd ben om de meeste functionaliteiten binnen de geplande tijdlijnen te implementeren. Echter, sommige functies



hebben mij meer tijd gekost dan verwacht, vooral de gedeelten van boete berekenen en onbekende locatie weergeven hebben mij wel extra tijd en aandacht gekost.

**Makkelijkere en moeilijker functies:** Het uploaden van gasgegevens en het analyseren van meetgegevens verliepen over het algemeen soepeler dan verwacht. Deze taken waren goed gedefinieerd en relatief eenvoudig te uploaden. Aan de andere kant waren de functies voor het verwerken en tonen van bezoekrapporten wel wat lastiger. Het was ook even rommelen met de gas en boete formules in code verwerken.

**Trots en leerpunten:** Ik ben trots op hoe alles mooi stapsgewijs verliep, ik week weinig af van mijn plan van aanpak en dit heeft ervoor gezorgd dat ik eerst de structuur op orde en dan de technische kanten van het programma.

Een belangrijke les die ik geleerd heb is het belang van een grondige analyse van de casus en meerdere Python bronnen en forums gebruiken om te achterhalen hoe bepaalde analyses geautomatiseerd kunnen worden. Als ik op tijd gebruik had gemaakt van andere externe bronnen (los van Edhub) dan kon ik het project ook eerder afronden.

**Toekomstige aanpak:** In de toekomst ga ik vaker gebruik maken van educatieve YouTube video's en meer vragen stellen in Python forums

## 4. Verantwoordingsdocument Schone Lucht B.V.

### **Wijze van coderen:**

- De code is geschreven met het oog op leesbaarheid en gebruiksvriendelijkheid. Door het gebruik van functies en commentaar is de code gemakkelijk te begrijpen en te onderhouden.
- Er wordt gebruik gemaakt van ingebouwde Python-functies en bibliotheken zoals NumPy voor gegevensverwerking en Matplotlib voor analyse-visualisaties.
- Waar mogelijk zijn algemene programmeerprincipes toegepast, zoals het vermijden van herhaalde code (DRY - Don't Repeat Yourself) en het gebruik van passende naamgeving van de variabelen.

### **Selectie van variabelen, functies, klassen en methoden:**

- Variabelen zijn zorgvuldig benoemd om hun functie en doel in de code duidelijk te maken.
- Functies zijn ontworpen om specifieke taken uit te voeren volgens het principe van functionaliteitscheiding. Elke functie heeft een duidelijk omschreven doel en voert een specifieke actie uit.

- Klassen zijn gebruikt wanneer een groep gerelateerde functies en variabelen een geheel vormt. Dit helpt bij het organiseren van de code en het bevorderen van herbruikbaarheid.
- Methoden binnen klassen zijn ontworpen om de functionaliteit van die klasse aan te vullen en te verbeteren.

#### **Automatisering van handelingen:**

- De applicatie automatiseert verschillende handelingen die relevant zijn voor de casus, zoals het inlezen van permanente gegevens en meetgegevens, het analyseren van meetgegevens en het tonen van bezoekrapporten.
- Door automatisering worden repetitieve taken verminderd en wordt de efficiëntie van het proces verbeterd.
- Automatisering draagt bij aan het verminderen van menselijke fouten en verhoogt de nauwkeurigheid en betrouwbaarheid van het systeem.

#### **Waarde van de applicatie:**

- De applicatie biedt een geautomatiseerde oplossing voor het controleren van de uitstoot van bedrijven, zoals vereist in de casus van Schone Lucht B.V.
- Het stelt de gebruiker in staat om permanente gegevens en meetgegevens te uploaden en te analyseren om vervuilende bedrijven op te sporen en bezoekrapporten van inspecteurs te bekijken.
- De applicatie draagt bij aan nauwkeurigere controle van bedrijfsuitstoot, waardoor naleving van regelgeving wordt verbeterd.