Matthew Jim, studentnummer: 1812418

V2B

**Verslag Programming Inleveropdracht 1: FSM Programmeren (individueel)**

**Inleiding:**

Voor de opdracht Programming Inleveropdracht 1: FSM Programmeren (individueel) heb ik een koffie machine gesimuleerd, gebruik makend van een Finite State Machine. Ondanks de koffie machine niet alles kan, kan de koffie machine wel genoeg om een simulatie uit te voeren. De code is in Python gemaakt met een OOP (Object Oriented Programming)methode.

**De Koffie Machine (FSM):**

Diagram, schematic

Description automatically generated

**Class Handeling**

In de koffie machine heb je verschillende opties waar je uit kunt kiezen. De koffie machine kan iemand iets laten kopen, de koffie machine aanvullen, kijken hoeveel ingredienten die heeft en weggaan. Om er makkelijk over heen te loopen totdat er een geldig antwoord uit komt heb ik een enumeratie gemaakt van de verschillende opties.

Text

Description automatically generated

**Class Buyer**

Om iets te kopen van de koffie machine heb je een koper nodig. Hievoor heb ik een aparte class gemaakt met een naam en een hoeveelheid geld. Met dat geld kan die gene verschillende soorten koffie kopen in de koffie machine.

**Class KoffieMachine**

In de koffie machine heb je verschillende attributen zoals de hoeveelheid water in ml en hoeveelheid koffie bonen in gram. In deze class worden alle mogelijkheden uitgewerkt in de Handeling functies en worden de states waar die op dat moment ook is aangepast.

In het begin word er gevraagd wat je wilt doen. Je kunt dat kiezen uit de verschillende soorten handelingen. Voor elke functie is gekeken naar verschillende types input. Bijvoorbeeld wanneer er een interger verwacht word kan het antwoord ook alleen een interger zijn.



**Buy**

Wanneer je kiest om iets te kopen krijg je allemaal opties van dranken waar je uit kunt gaan kiezen en een optie om weer terug te gaan naar het main menu. Als je niet naar het main menu gaat, dan heb je alle dranken en de ingredienten in een namedtuple. Ik heb ze in een namedtuple gedaan zodat ik de verschillende ingredienten makkelijk kan accessen met de naam in plaats van de index. Waneer je het drankje hebt word er daarna in de check() functie gekeken of er nog wel genoeg ingredienten in de koffie machine zit. Als dat zo is word het geld afgeschreven van de koper en het drankje gemaakt.



Text

Description automatically generated

Text

Description automatically generated

**Fill & Remaining**

Als je ervoor koos om de machine te vullen word er voor elk ingredient gevraagd hoeveel je ervan wilt gaan toevoegen. Hier is het alleen mogelijk om intergers in te vullen, als je dat niet doet dan word het opnieuw gevraagd. In remaining krijg je te zien hoeveel van elk ingredient de koffie machine heeft. Daarna gaat het weer terug naar het main menu.

Text

Description automatically generatedText

Description automatically generated

**Discussie:**

Ik ben tevreden met de code die ik heb gemaakt. Alles ging wel redelijk goed totdat ik het moest gaan koppelen met de FSM. Want het lukte niet op de manier hoe ik het had gewild. In de toekomstige versie zouden de current\_states veranderd kunnen worden met een FSM.run() functie.