

SAMENGEVAT VOOR VANDAAG

- Vorm een groep.
- Team leader gaat langs bij Noah met laptop.
- Implementeer scenario 1 (enkel tekst tonen) met contracten.
- Implementeer eventueel ook al scenario 2.

6 Opgave TO groep

6.1 De probleemstelling

Quetzal, de chocoladebar, is op zoek naar een e-shop. Gebruikers kunnen via jullie programma chocolademelk bestellen die ze een kwartiertje later kunnen komen halen. Elke chocolademelk bestaat uit room en melk. Daarbij wordt volgens de wensen van de klant iets toegevoegd: één of meerdere shots chocolade (gekozen uit wit, melk, bruin, zwart), honing, chilipeper of marshmallows. Op basis van de combinatie wordt de uiteindelijke prijs bepaald. Een klant kan zoveel toevoegen aan een chocolademelk als hij/zij wil. Op tijd en stond moet de stock van de chocoladebar aangepast kunnen worden. Het moet ook mogelijk zijn om een gesorteerde versie van alle producten, bestellingen en gebruikers op te vragen.

Werknemers moeten kunnen toegevoegd en verwijderd worden. Elke werknemer is anders en afhankelijk van de ervaring kan die meer of minder werk verzetten. Deze workload wordt uitgedrukt in credits. Een bepaalde werknemer kan zijn workload – dit is een vast aantal credits – verwerken in 1 tijdseenheid.

De workload van een chocolademelk wordt bepaald door alle toevoegingen. Reken op 5 credits per chocolademelk en 1 credit per chocoladeshot, honing, Een chocolademelk met twee soorten chocola en 1 portie honing komt dan neer op 8 credits.

Een werknemer werkt bestelling per bestelling af. Indien nodig kan die een aantal opeenvolgende tijdseenheden na mekaar werken aan dezelfde bestelling. Werknemer A met 10 credits werkt een chocolademelk van 8 credits in 1 keer af, maar werknemer B met 3 credits doet het in 3 keer ($3+3+2$). Per tijdseenheid kan een werknemer zijn volledige workload van 1 bestelling verwerken. In ons voorbeeld doet werknemer A 1 tijdseenheid over de chocolademelk, terwijl werknemer B er 3 nodig heeft. We bekijken het systeem telkens aan het begin van het tijdsinterval.

6.2 Analyse

De volgende objecten dienen als bouwstenen voor de oplossing. Stel voor elk een ADT op.

- Chocolademelk: krijgt een id¹ en heeft een basisprijs van 2 EUR. Zoeksleutel: id.
- Chocoladeshot: bevat een id en kan witte, melk-, bruine, of zwarte chocolade zijn. Kost 1 EUR en heeft een vervaldatum. Zoeksleutel: id.
- Honing: bevat een id en kost 0,50 EUR en heeft een vervaldatum. Zoeksleutel: id.
- Marshmallow: bevat een id en kost 0,75 EUR en heeft een vervaldatum. Zoeksleutel: id.
- Chilipeper: bevat een id en kost 0,25 EUR en heeft een vervaldatum. Zoeksleutel: id.
- Gebruiker: bevat een id, een voornaam, een achternaam en een e-mailadres. Zoeksleutel: e-mailadres.
- Stocks: er is een stock van chocoladeshots, een stock van honingporties, een stock van marshmallows en een stock van chilipeperporties. Dit moeten tabellen van objecten zijn, geen tellers.
- Bestelling: bevat een id, een gebruikersid, een timestamp (tijd en datum), een chocolademelk-id en een veld om bij te houden of die al afgehaald is of niet. Blijft altijd bewaard. Zoeksleutel: id.
- Werknemer: heeft een id, een voornaam, een achternaam en een bepaalde workload uitgedrukt in credits. Zoeksleutel: id.

Op basis van deze analyse gaan we werken naar een ontwerp dat zal leiden tot het eindproduct. We doen dat in drie stappen:

1. een contract opstellen voor elk object uit de analyse (opdracht 5),
2. een basisimplementatie van een deel van de objecten m.b.v. een basis-scenario (opdracht 6.1),
3. een basisimplementatie van alle objecten m.b.v. een uitgebreid scenario (opdracht 6.2),

¹dit is een uniek getal dat overeenkomt met dit object. Je mag daar een getal van maken (1, 2, 3, ...). Die moet niet uniek zijn voor heel het programma. Je kan een gebruiker hebben met id 1 en een chocolademelk met id 1, maar uiteraard wel per type: er is maar 1 gebruiker met id 1

7 Opdracht 5: De eerste ADT's

We stellen een ADT op voor alle gegeven structuren uit de opgave.

1. Maak een nieuw project aan in PyCharm.
2. Maak een python bestand per ADT. Start de naam met een hoofdletter!
3. Zet bovenaan in commentaar waarvoor je een ADT opstelt.
4. Welke data bevat jouw ADT? Maak variabelen met goede namen aan en initialiseer die op goede waarden.
5. Welke functionaliteit gaat jouw ADT aanbieden? Voeg een contract toe aan elk ADT. Een contract bestaat uit verschillende functies. Een contract bestaat uit volgende zaken:
 - de naam van de functie,
 - de input parameter(s) met uitleg,
 - de beschrijving van de functie (in een docblock),
 - de output parameter(s) (je geeft die terug via een return van een tuple),
 - de pre- en postcondities. Precondities zijn voorwaarden waar de oproeper van je methode zich aan moet houden (bv n moet een strikt positief geheel getal zijn). Een postconditie is een belofte die je maakt die waar is nadat de methode werd opgeroepen (bv er zit nu 1 element meer in de lijst). Die kunnen achteraf nog bijgevoegd worden, maar als je een veronderstelling maakt, kan die hier dus al gezet worden.

Je doet dit door een Python functie te definiëren met als body 'pass'. Dat is een functie die op dit moment nog niets doet, maar dan krijg je geen errors.

7.1 Een voorbeeld

Merk op dat dit slechts een voorbeeld is dat zeker niet volledig is, maar in dit voorbeeld komt alles aan bod wat nodig is, telkens toegepast op dat voorbeeld. Vergeet ook niet dat het hier maar over 1 functie gaat en je er vermoedelijk meerdere zal nodig hebben per ADT (en voor sommige zelfs geen). Dat gaat waarschijnlijk nog groeien van zodra we het prototype gaan bouwen, maar het is een goed startpunt waar we dan op verder kunnen werken de volgende weken. De bedoeling is om eerst het probleem wat te kneden en pas daarna tot een oplossing te komen m.a.w. eerst denken, dan doen ;-).

```
# ADT Stack
## data
items = []

## functionaliteit
def push(item):
    """
    voeg een item toe aan de top van de stack

    preconditie: er is nog plaats op de stack
    postconditie: de stack is 1 item groter en de top bevat het toegevoegde
        item

    :return: geeft True terug asa het toevoegen gelukt is
    """
    pass
```

8 Opdracht 6: Een prototype

Soft deadline : dinsdag 11/2/2025 om 22u

Hard deadline : donderdag 13/2/2025 om 22u

Om snel een inzicht te krijgen in wat de klant precies wil, werkt men binnen agile development met snel gemaakte prototypes voor een vereenvoudigd probleem. We vereenvoudigen alles en reduceren het tot werknemers die bestellingen afhandelen. We vergeten dus even dat het gaat om chocolademelk en verschillende andere ingrediënten en we verwaarlozen ook even de stock. Let op: aan deze implementatie zal dus nog veel gesleuteld moeten worden, maar het geeft een beter inzicht hoe alles aan elkaar zal hangen.

8.1 Implementeer volgende klassen.

De methodes moeten nog niet volledig klaar zijn en mogen gewoon afdrukken wat er gebeurt. Voor de ketting en binaire zoekboom gebruik je de code van TOi (van ING). Als je die nog niet hebt, gebruik je een gewone lijst voor de ketting en een dict voor de boom.

- Werknemer
- Bestelling
- Circulaire dubbelgelinkte ketting (gebruik ADT lijst uit de cursus voor het contract)
- Binaire zoekboom
- MyStack
- MyQueue

8.2 Implementeer het volgende scenario. Zorg dat elk van volgende stappen een regel op het scherm afdrukt.

Voor het tijdstip gebruik je gewoon een teller op dit moment.

1. Maak een werknemer A met een workload van 10 credits en plaats die in de ketting.
2. Voeg die toe aan de stack.
3. Maak een werknemer B met een workload van 3 credits en plaats die in de ketting.
4. Voeg die toe aan de stack.

5. Maak een bestelling van 5 credits op tijdstip 0.
6. Voeg die toe aan de queue.
7. Maak een bestelling van 8 credits op tijdstip 1.
8. Voeg die toe aan de queue.
9. Zet de tijd op 2.
10. Lees de eerste bestelling uit de queue.
11. Lees de eerste werknemer uit de stack.
12. Geef de bestelling aan de werknemer.
13. Lees de tweede bestelling uit de queue.
14. Lees de tweede werknemer uit de stack.
15. Geef de bestelling aan de werknemer.
16. Verhoog de tijd.
17. Verlaag de workload van elke werknemer met zijn maximum workload.
18. Beslis of de werknemer klaar is of niet. Zo ja, dan komt die terug op de stack en plaats je die bestelling in de binaire zoekboom.
19. Verhoog de tijd.
20. Verlaag de workload van elke werknemer met zijn maximum workload.
21. Beslis of de werknemer klaar is of niet. Zo ja, dan komt die terug op de stack en plaats je die bestelling in de binaire zoekboom.

Je code ziet er dus bijvoorbeeld als volgt uit:

```
r = Reservatiesysteem()
r.werknemers = LinkedChain()
werknemerA = Werknemer("A","",10) # lege achternaam
r.werknemers.insert(1,werknemerA)
...
```

Vermoedelijk heb je bij het implementeren van dit scenario een aantal nieuwe methodes ontdekt. Voeg die toe aan je klassen met een gepast contract.

8.3 Implementeer volgende klassen tegen donderdag 22u

We hebben nu een redelijk goed idee van hoe de basis klassen samenwerken. We voegen nu de overige toe. Implementeer volgende klassen. De methodes moeten nog niet volledig klaar zijn en mogen gewoon afdrukken wat er gebeurt.

- Chocolademelk
- Chocoladeshot
- Honing
- Marshmallow
- Chilipeper
- Gebruiker
- Stocks

8.4 Implementeer het volgende scenario. Zorg dat elk van volgende stappen een regel op het scherm afdrukt.

We breiden het scenario uit.

1. *Maak een gebruiker Els en plaats die in de ketting van alle gebruikers.*
2. *Maak een gebruiker Tom en plaats die in de ketting van alle gebruikers.*
3. *Maak een werknemer A met een workload van 10 credits en plaats die in de ketting.*
4. *Voeg die toe aan de stack.*
5. *Maak een werknemer B met een workload van 3 credits en plaats die in de ketting.*
6. *Voeg die toe aan de stack.*
7. *Els bestelt een chocolademelk zonder toevoegingen op tijdstip 0.*
8. *Voeg die toe aan de queue.*
9. *Tom bestelt een chocolademelk met 2 chocoladeshots en 1 portie honing op tijdstip 1.*
10. *Voeg die toe aan de queue.*
11. *Zet de tijd op 2.*
12. *Lees de eerste bestelling uit de queue.*

13. Lees de eerste werknemer uit de stack.
14. Geef de bestelling aan de werknemer.
15. Lees de tweede bestelling uit de queue.
16. Lees de tweede werknemer uit de stack.
17. Geef de bestelling aan de werknemer.
18. Verhoog de tijd.
19. Verlaag de workload van elke werknemer met zijn maximum workload. *Indien nodig pas je de stock aan.*
20. Beslis of de werknemer klaar is of niet. Zo ja, dan komt die terug op de stack en plaats je die bestelling in de binaire zoekboom.
21. Verhoog de tijd.
22. Verlaag de workload van elke werknemer met zijn maximum workload. *Indien nodig pas je de stock aan.*
23. Beslis of de werknemer klaar is of niet. Zo ja, dan komt die terug op de stack en plaats je die bestelling in de binaire zoekboom.

9 Opdracht 7: voorbereiding groepswork

9.1 Tijdens de les van dinsdag

Deadline: groep kiezen tegen woensdag 12/2/2025 om 22u

Deadline: contracten insturen tegen donderdag 13/2/2025 om 22u

1. Je schrijft je in in een groep op Blackboard (via de groepsset 'TOg').
2. Team leaders maken een groep aan door langs te gaan met hun laptop bij Noah tijdens de les van dinsdag (via volgende github classroom <https://classroom.github.com/a/OoXrfILv>). Kies als groepsnaam het nummer van je groep op BB met 2 cijfers gevolgd door een underscore en dan de voornamen van de leden gescheiden door underscores (bv 01_John_Jane_Jelly_Jimmy).) Je doet dit enkel als je op BB een groep hebt met 4 studenten.
3. In de repo zit een map prototypes. Maak daarin een map met je eigen naam. Vanaf woensdag zet elke student zijn of haar contracten uit de vorige opdracht(en) in die map. Je vertrekt dus van je eigen python code en verwijdert alle bodies van alle methodes zodat enkel de naam van de methode en de docblocks met de contracten blijven staan. Het gaat ENKEL over de contracten van de klassen Werknemer, Bestelling, Chocolademelk, Chocoladeshot, Honing, Marshmallow, Chilipeper, Gebruiker en Stocks(en eventueel nog andere klassen die jij nodig acht), maar NIET van de queue, stack, ketting, bst, heap, hashmap, 2-3-boom, 2-3-4-boom en roodzwartboom. Je moet dus ook niet het scenario insturen. Dat dient alleen maar om te weten te komen welke methodes je nodig hebt.

SAMENGEVAT VOOR VANDAAG

- Vorm een groep.
- Team leader gaat langs bij Noah met laptop.
- Implementeer scenario 1 (enkel tekst tonen) met contracten.
- Implementeer eventueel ook al scenario 2.