**Feedback op de oplossing van groep:**

Groep: David scalais Thomas Ghysels Thomas Urkens Daan Th. **(Beoordeling David door Daan)**

Opdracht: Quetzal shop

**Wat vond je van deze oplossing?**

Deel A: STELLINGEN

*Zet achter elke stelling een kruisje in het voor jou passende hokje. Licht je antwoord toe. Feedbackformulieren waarin geen toelichtingen gegeven worden gaan je medestudent niet vooruit helpen en worden dan ook als niet afgegeven beschouwd.*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Volledig akkoord | Akkoord | Neutraal | Niet akkoord | Helemaal niet akkoord | Toelichting |
| De design laat toe de implementatie vlot te gebruiken. | x |  |  |  |  | Alle functies zijn duidelijk en simpel |
| De design is modulair, de modules hebben duidelijke contracten |  | x |  |  |  | De contracten zijn straight tot he point. |
| In de design wordt gesproken over WAT, niet HOE (geen verwijzingen naar implementatie) | x |  |  |  |  | Enkel wat en niet hoe. |
| Duidelijke naamkeuze (van variabelen, constanten, functies, …) |  | x |  |  |  | Namen zijn duidelijk maar lang. |
| Zijn er methodes voorzien om makkelijk van ADT te wisselen? Je wil bv wisselen van 2-3-boom naar een hashmap. |  |  |  | x |  | Geen specifieke methodes voorzien maar is wel mogelijk om te implementeren. |
| Er is object-georienteerd gewerkt | x |  |  |  |  | Klassen en init functies zijn er steeds. |
| Deze oplossing komt in aanmerking als voorbeeld voor de andere studenten |  | x |  |  |  | Het design is simpel maar effectief. |

Deel B: OPEN VRAGEN

*Beantwoord kort onderstaande open vragen. Probeer in je antwoord zo concreet mogelijk te zijn.*

Flexibiliteit

*Een design is pas flexibel als het goed tegen wijzigingen kan. Stel nu dat we de volgende wijzigingen doen. Geef de klassen die* ***toegevoegd*** *moeten worden (enkel bij naam, geen methodes of attributen) en de klassen die vermoedelijk moeten* ***aangepast*** *worden in dit ontwerp. Baseer je dus enkel op de contracten van dit ontwerp. Als je dat niet kan, duid dan goed aan waar het niet duidelijk is wat er moet gebeuren en/of waar er moet aangepast worden.*

|  |
| --- |
| We implementeren een AVL-boom en willen die gebruiken om de chocoladeshots op te slaan. |
| *We hebben al een contract die alle parameters aanduidt, we moeten dus enkel die parameters nog gebruiken in een klasse voor de AVL-boom.* |

|  |
| --- |
| We willen dat bestellingen nu gesorteerd worden op gebruikersid. |
| *We implementeren een gesorteerde array bij de klasse gebruiker die alle id’s bijhoudt.* |

|  |
| --- |
| Er zijn nu ook light chocoladeshots, met minder suiker, maar enkel voor de witte en melkchocolade. |
| *We voegen een parameter toe aan de klasse van de chocoladeshots die bepaald of de chocolade light is of niet.* |
| TOPS - volgende onderdelen van de oplossing vond ik bijzonder goed: |
| *De klasse gebruiker omdat er redelijk veel methodes in deze functie zitten die ook elk veel nut hebben.* |

|  |
| --- |
| TIPS - welke wijzigingen (qua structuur, inhoud, vormgeving, klasses die je mist, methodes die je mist, …) die je nog niet vermeldde bij één van de bovenstaande vragen, kan je suggereren? Licht toe. |
| Meer functionaliteit aan de ‘werknemer’ klasse omdat er nu niet veel meer bij kan gebeuren dan nieuwe werknemers aanmaken en ze verwijderen. |