**Feedback op de oplossing van groep:**

Groep: David scalais Thomas Ghysels Thomas Urkens Daan Th.

Opdracht:

**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.*

*Feedback van Thomas G voor Thomas Urkens*

|  | Volledig akkoord | Akkoord | Neutraal | Niet akkoord | Helemaal niet akkoord | Toelichting |
| --- | --- | --- | --- | --- | --- | --- |
| De design laat toe de implementatie vlot te gebruiken. | x |  |  |  |  | Goed geordend, De uitleg erbij is duidelijk. |
| De design is modulair, de modules hebben duidelijke contracten | x |  |  |  |  | Veel contracten die duidelijk uitgelegd zijn. |
| In de design wordt gesproken over WAT, niet HOE (geen verwijzingen naar implementatie) | x |  |  |  |  | Het geeft wat het is en dit is zeer duidelijk |
| Duidelijke naamkeuze (van variabelen, constanten, functies, …) | x |  |  |  |  | Namen geven duidelijk weer over wat het gaat. |
| Zijn er methodes voorzien om makkelijk van ADT te wisselen? Je wil bv wisselen van 2-3-boom naar een hashmap. |  |  |  |  | x | Er zijn methodes voorzien waarin het kan veranderen. |
| Er is object-georienteerd gewerkt | x |  |  |  |  | Er zijn verschillende classes. |
| Deze oplossing komt in aanmerking als voorbeeld voor de andere studenten | x |  |  |  |  | Zeer duidelijk opgesteld en goed te volgen. |

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. |
| --- |
| *Er zou een class moeten toegevoegd worden waardoor je de class chocoladeshots zou kunnen opslaan in een AVL-boom.* |

| We willen dat bestellingen nu gesorteerd worden op gebruikersid. |
| --- |
| *Er is in de class een self.gerbuikersid. Dit zou dus wel moeten lukken als je gebruik maakt van dit.* |

| Er zijn nu ook light chocoladeshots, met minder suiker, maar enkel voor de witte en melkchocolade. |
| --- |
| *Een nieuwe parameter light chocoladeshots toevoegen. En deze parameter zal dan een true of false geven als je dit kiest of niet.* |
| TOPS - volgende onderdelen van de oplossing vond ik bijzonder goed: |
| *Bij de bestelling gebruik maken van veel verschillende parameters.*  *Hierdoor is er veel meer mogelijkheid om iets te doen.* |

| 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. |
| --- |
| *Bij de chocolade shots nog een verschil maken tussen welke soort chocolade je kiest.* |