# Examen Object Georiënteerd Programmeren & Databanken

### Dinsdag 22 januari 2019

Naam: Groep:

**De eerste 2 vragen beantwoord je zonder een computer. Je mag enkel gebruikmaken van je handgeschreven samenvatting van maximum 4 bladzijden. Voor dit deel heb je maximaal 1u, na het indienen van dit gedeelte krijg je de opgave van vraag 3 en kan je starten aan de computer. Het totale examen duurt maximaal 4u.**

## Van probleemstelling naar klasse diagram (20% van totaal, teken je klasse diagram op de daartoe voorziene plaats)

Genereer een gedetailleerd klasse diagram startend van volgende probleem beschrijving: definieer de benodigde klassen, hun attributen (type en naam), de signatuur van hun methodes en de relaties tussen de verschillende klassen.

Je wordt gevraagd om een softwaresysteem te ontwikkelen dat Alma helpt om bij te houden wie wat voor eten en drinken koopt en hoeveel geld er wordt uitgegeven.

Klanten kunnen studenten of medewerkers zijn; alle klanten hebben een uniek identificatienummer dat wordt berekend door het aantal letters in hun naam te tellen. Indien dit nummers reeds bestaat wordt het verhoogd tot een uniek nummer is gevonden. Klanten hebben ook een portemonnee (“wallet”) die hun budget voorstelt. Je moet in staat zijn om nieuwe klanten aan het systeem toe te voegen met hun achternaam en een bedrag in hun portemonnee. Je moet natuurlijk ook deze klantinformatie kunnen opvragen.

Alle verkochte items hebben een id, een naam, een type (drank = 1, sandwich = 2, snoep = 3), een prijs, een gereduceerde studentenprijs (lager dan de normale prijs, maar niet een vast tarief) en een vervaldatum. Je moet in staat zijn om de artikelprijzen op te vragen samen met andere relevante artikelinformatie die nodig is in het kader van aankopen, bijvoorbeeld of een bepaald artikel is verlopen. Alma wil ook graag nieuwe items kunnen maken door alle benodigde informatie aan te leveren of door deze informatie uit te lezen uit een tekstbestand.

Een aankoop bevat een lijst met items en de klant die de aankoop heeft gedaan. Het Alma systeem houdt ook alle aankopen bij die ooit zijn gedaan. Wanneer een nieuwe aankoop wordt gedaan, moet het systeem controleren of de klant een student is om na te gaan of de gereduceerde studententarief van toepassing is, en moet het controleren of de verzameling van items die de klant wenst te kopen betaalbaar is door de klant. Het wil ook controleren of een bepaald artikel in een aankoop zit en het moet alle vervallen artikelen van een bepaalde aankoop kunnen verwijderen. Je moet van een aankoop kunnen opvragen of deze met succes is afgerond, welke klant de aankoop heeft gedaan en welke artikelen erin zijn opgenomen.

## Stukjes code (schrijf antwoord op opgave, 10% van totaal)

What is het resultaat van een oproep van isThisTheHeaviest() met k3 als parameter. Neem aan dat volgende Koe objecten in de collection animals zitten (de waarde tussen haakjes is hun gewicht): k1 (83), k2 (102), k3 (75) en k4 (102).

public boolean isThisTheHeaviest(Koe k)

{

Koe zwaarste = null;

double maxWeight = 0.0;

for (Koe koe : animals)

{

if (koe.getWeight() >= maxWeight)

{

maxWeight = koe.getWeight();

heaviest = koe;

}

}

return zwaarste == k;

}

Result = resultaat =

# Examen Object Georiënteerd Programmeren & Databanken

### Dinsdag 22 januari 2019

Naam: Groep:

(b). Welke fouten geeft de compiler voor volgende code. Duid de lijnen aan waar zich een fout bevindt en verbeter ze in de code. Geef met een pijl aan welke code logisch fout is (syntax = OK, maar geeft niet resultaat dat je wenst).

public class Koe

{

private final String naam;

private final double gewicht;

private Koe moeder;

public Koe(String stukNaam, double startGewicht, Koe moe)

{

naam = "Koe" + stukNaam;

gewicht = startGewicht;

moe = moeder;

}

public void voeder(int hoeveelheid)

{

gewicht += hoeveelheid ^ 0.25;

}

public String toString()

{

String str = "Koe met naam=";

str += naam + "\n, heeft als gewicht=" + gewicht;

if (moeder != null)

{

str + "en als moeder " + moeder.toString();

}

}

public Koe kweek()

{

return new Koe("dochter van " + naam, Math.*random*() \* 20.0, this);

}

}

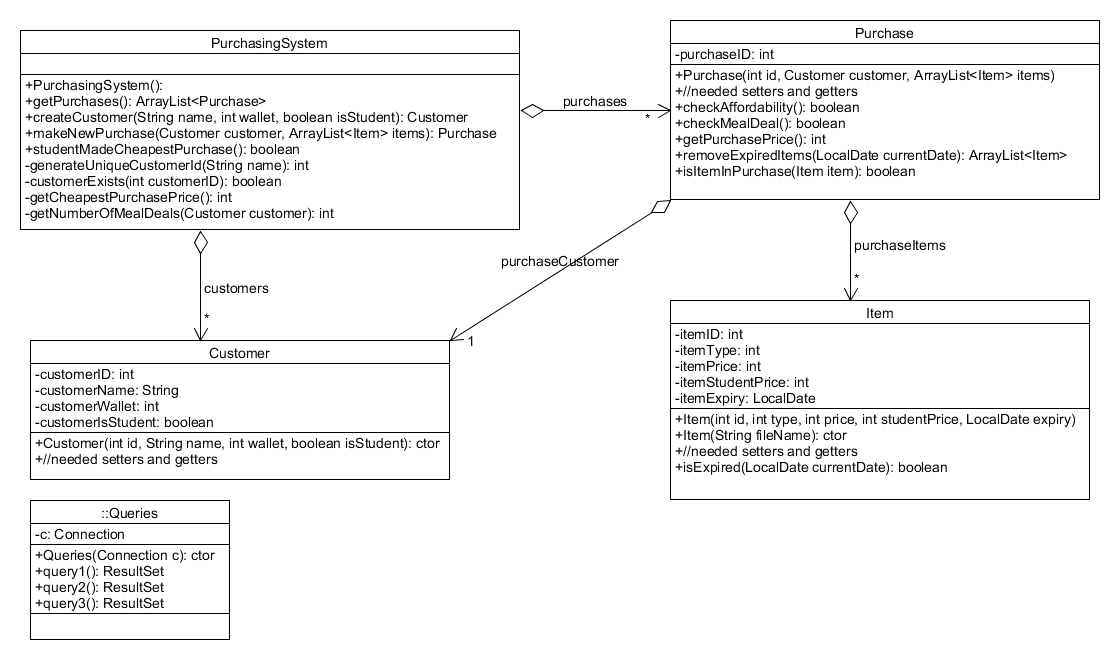
**Teken hier je klasse diagram**

Naam: Groep:

## Implementatie van het klasse diagram (70% van totaal)

Je gaat met je browser naar http://toledo.kuleuven.be/examen en selecteer examen OOP&DB. In de “Read me first” sectie, vind je extra informatie over de Toledo examenomgeving en de regels die hier gelden. Je kan alleen surfen naar deze site, alle andere netwerkactiviteit is verboden. Ga dan naar de eigenlijke Examen sectie en selecteer de assignment 22012019. Daar vind je de zip file waar je van vertrekt. In deze zip file vind je ook een pdf bestand met de probleemstelling uit vraag 1 als extra info. Het paswoord om dit bestand te unzippen krijg je van je begeleider. Na het unzippen, kan je dit bestand best verwijderen. Naast de assignment is er ook een zip bestand met een gedeelte van de javadoc documentatie. Je kan die dan ook ergens lokaal op je computer uitpakken en gebruiken.

Je start van volgend klasse diagram. ***Je mag de signatuur van de gegeven methodes NIET wijzigen, noch het type van de gegeven attributen. Het is wel mogelijk dat je extra attributen of methoden moet toevoegen om te voldoen aan alle vereisten.***



Test 1 controleert de constructor van Customer met behulp van de methode PurchasingSystem.createCustomer. Nadien wordt de klant ook toegevoegd aan de customers collectie.

Test 2 controleert de getters en setters van Customer en Item.

Test 3 controleert de berekening van de Item prijzen voor studenten en niet-studenten.

Test 4 controleert of een aankoop betaalbaar is (wat betekent dat de klant genoeg geld heeft om ervoor te betalen) door checkAffordability op te roepen.

.

Test 5 controleert de basislogica van PurchasingSystem: of een nieuwe aankoop kan worden gedaan door makeNewPurchase op te roepen.

Test 6 controleert of een prijsvermindering van toepassing is met makeNewPurchase: als een klant een drankje en een broodje koopt, hoeft hij alleen de studentenprijs te betalen, ongeacht de status.

Test 7 controleert of het mogelijk is om een nieuw item aan te maken uit een bestand door de constructor op te roepen met de bestandsnaam als parameter. Zie item.txt voor details over de structuur.

Test 8 controleert of de goedkoopste aankoop in het systeem is gedaan door een student.

Test 9 controleert of een bepaald item in een bepaalde aankoop zit met behulp van de methode Purchase.isItemInPurchase.

Test 10 verwijdert alle vervallen items van een aankoop en geeft een lijst terug van de items die verwijderd zijn.

De volgende tests omvatten queries die je moet uitvoeren op de databank PurchasingSystem.db. Je mag de structuur van deze databank niet wijzigen.

|  |  |  |
| --- | --- | --- |
| **Table: Customers** | | |
| Column | Data type | Constraints |
| id | INTEGER | PRIMARY KEY AUTOINCREMENT  UNIQUE  NOT NULL |
| customerID | INTEGER |  |
| name | STRING |  |
| student | BOOLEAN |  |
| wallet | INTEGER |  |

|  |  |  |
| --- | --- | --- |
| **Table: Items** | | |
| Column | Data type | Constraints |
| id | INTEGER | PRIMARY KEY AUTOINCREMENT  UNIQUE  NOT NULL |
| type | INTEGER |  |
| price | INTEGER |  |
| studentPrice | INTEGER |  |

|  |  |  |
| --- | --- | --- |
| **Table: Purchases** | | |
| Column | Data type | Constraints |
| customerID | INTEGER | REFERENCES Customers (id) |
| itemID | INTEGER | REFERENCES Items (id) |

Je kan de queries uitproberen in SQLiteStudio. Bij elke vraag staat aangegeven wat de correcte resultset moet zijn. Als je de juiste query gevonden hebt, kan je de string kopiëren naar de Queries klasse op de juiste plaats. Zorg er voor dat de string geen speciale karakters als newlines of tabs bevat. Zo kan je dan ook de query testen van uit BlueJ.

Test 11 controleert de query die een lijst van alle informatie van items teruggeeft die nooit gekocht zijn.

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | 1 | 2 | 1 |

Test 12 controleert de query die een gesorteerde lijst van unieke item types die gekocht zijn door de klant customerID = 6.

|  |
| --- |
| 1 |
| 2 |

Test 13 controleert de query die een lijst geeft met alle informatie van items die enkel door studenten gekocht werden.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 2 | 1 |

Indienen doe je ook via Toledo. Je klikt op de assignment die voor jou actief is. Je zipt de ***HELE MAP*** waar je BlueJ project staat en dit bestand laad je op naar Toledo. ***Zorg er voor dat je NIET het originele zip bestand upload waar je van vertrokken bent, anders haal je simpelweg een 0 voor dit gedeelte van het examen . Zorg er ook voor dat dit enkel je JAVA code is en NIET de Javadoc tools***

Indien de upload gelukt is moet je nog een laatste stap uitvoeren: het examen verlaten. Klik daarvoor op de knop “Verlaat examen”, je moet verklaren dat je alles ingediend hebt wat je wou indienen. Hierna heb je geen toegang meer tot het examen.