



Tentamen OOP Make IT Work

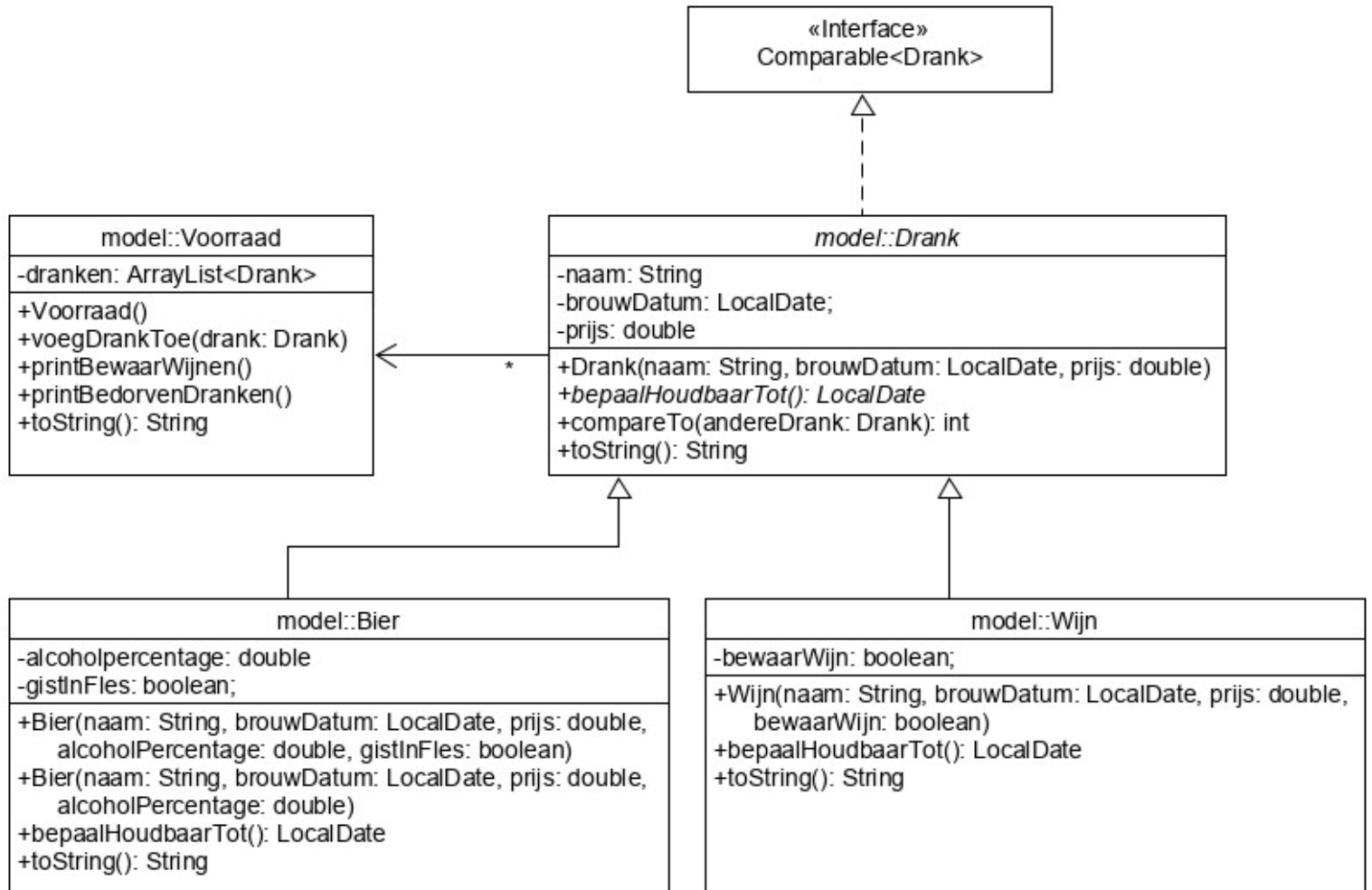
Naam	
Datum en tijd	18 November 2021 09:30-12:00

Opmerkingen

1. Je krijgt voor deze opgave een startproject genaamd Drankenhandel.
2. Je krijgt een sql-script om een database te maken met een user. Deze staat in de map resources van het startproject.
3. Upload na afloop een zip van je hele project naar Blackboard.
4. Deze toets bestaat uit 8 pagina's.
5. Het gebruik van het internet is toegestaan.
6. Daarnaast is het gebruik van het boek, je eigen aantekeningen en al het materiaal op je eigen laptop (inclusief uitwerkingen van practicumopgaven) toegestaan.
7. Als je het tentamen inlevert checkt de docent of de upload in Blackboard is gelukt. Verlaat het lokaal niet voordat die check is uitgevoerd en je toestemming hebt gekregen!

Klassendiagram Slijterij 't Lammetje

De voorraad van Slijterij 't Lammetje is de afgelopen tijd flink gegroeid. Er moet software ontwikkeld worden om de voorraad bij te houden en hiervoor is het volgende klassendiagram opgezet voor de klassen in de package model.





Algemene aanwijzingen

- De `SlijterijLauncher`-klasse is al voor je gemaakt en bevat veel testcode. Maak hier gebruik van! De voorbeeldoutput is ook gebaseerd op deze testcode.
- In het klassendiagram zijn geen getters en setters opgenomen. Voeg zelf de getter(s) en setter(s) toe die je nodig denkt te hebben. Je mag niet standaard alle getters en setters opnemen, maar alleen degene die je nodig hebt.
- In het klassendiagram staan geen klassen die nodig zijn voor het maken van een connectie met een database en ook geen DAO-klassen. Een `DBAccess` klasse en een `AbstractDAO` klasse zijn wel al aanwezig in je project.
- Zorg dat je `toString()` methodes dezelfde output genereren als in de outputvoorbeelden.
- De interface `Comparable` bestaat natuurlijk al in Java, die mag je dus niet zelf maken.
- Merk op dat datums als datatype de `LocalDate` klasse gebruiken. Je kunt dan handig gebruik maken van de methodes van een `LocalDate` object, zoals `plusYears()` en `isBefore()`.
- Als je zelf nog zaken wil toevoegen (omdat je denkt dat ze nodig zijn) die niet in het klassendiagram staan, geef dat dan duidelijk met commentaar aan in de code

Deel I: Bouw klassenstructuur

Run de applicatie. De output moet er als volgt uit zien.

Welkom bij Slijterij 't Lammetje, geschreven door: <Naam>

Stap 0: Verander de welkomstboodschap

Verander de boodschap zodat je eigen naam verschijnt.

Stap 1: Abstracte klasse `Drank` (15 punten)

Maak de abstracte klasse `Drank` volgens het klassendiagram. Houd hierbij rekening met de onderstaande eisen:

- LET OP: Het implementeren van de interface `Comparable` en de methode `compareTo()` hoef je pas in stap 6 te doen.
- De methode `bepaalHoudbaarTot()` is abstract.
- Override de methode `toString()`, zodat deze de naam, de prijs en de houdbaarheidsdatum van de drank teruggeeft in één string. Gebruik voor de houdbaarheidsdatum de methode `bepaalHoudbaarTot()`. De string moet er als volgt uitzien (voorbeeld):

```
Onbekende drank: 3,66 euro per fles  
Houdbaar tot: 2021-03-24
```

De houdbaarheidsdatum wordt default in Amerikaanse opmaak (yyyy-mm-dd) weergegeven. Je kunt hiervoor de `toString()` methode van de `LocalDate` klasse gebruiken. Je kunt dit pas testen in stap 2.



Stap 2: Subklasse Bier (15 punten)

Maak de subklasse `Bier` volgens het klassendiagram. Houd hierbij rekening met de volgende eisen:

- Implementeer de constructors op zo'n manier dat je dubbele code zoveel mogelijk voorkomt.
- Als voor het attribuut `gistInFles` geen waarde aan de constructor wordt meegegeven, dan moet dit attribuut de waarde `false` krijgen.
- Override de methode `bepaalHoudbaarTot()` en implementeer deze als volgt:
 - Als `gistInFles` op `true` staat, dan is het bier tot 16 jaar vanaf de `brouwDatum` houdbaar. Tip: Gebruik de methode `plusYears()` uit de `LocalDate` klasse.
 - Als `gistInFles` op `false` staat, dan is de houdbaarheid afhankelijk van het alcoholpercentage:
 - Als het alcoholpercentage lager of gelijk is aan 5%, dan is het bier tot 9 maanden vanaf de `brouwDatum` houdbaar.
 - Als het alcoholpercentage hoger is dan 5%, dan is het bier tot 2,5 jaar vanaf de `brouwDatum` houdbaar.
- Override de methode `toString()` en implementeer deze zoals het voorbeeld hieronder. Voorkom dubbele code en hergebruik zoveel mogelijk.

Gebruik `testStap2()` om je implementatie te testen. De output moet er als volgt uitzien.

```
--- Test stap 2 ---
Hertog Jan Karakter: 3.66 euro per fles
Houdbaar tot: 2022-06-18
Alcoholpercentage: 7.5%
Gist in fles: Nee

Amstel Pilsener: 1.98 euro per fles
Houdbaar tot: 2018-08-30
Alcoholpercentage: 5.0%
Gist in fles: Nee

La Trappe Dubbel: 6.5 euro per fles
Houdbaar tot: 2031-02-06
Alcoholpercentage: 4.28%
Gist in fles: Ja
```

*De houdbaarheid van de Hertog Jan Karakter is 7 maanden na de dag van het tentamen, mocht ik vergeten zijn dat aan te passen.



Stap 3: Subklasse Wijn (10 punten)

Maak de subklasse `Wijn` volgens het klassendiagram. Houd hierbij rekening met de volgende eisen:

- Override de methode `bepaalHoudbaarTot()` en implementeer deze als volgt:
 - Als `prijs` lager is dan 35 euro:
 - Als `bewaarWijn` op `true` staat, dan is de wijn 10 jaar vanaf de `brouwDatum` houdbaar.
 - Anders is de wijn tot 2 jaar vanaf de `brouwDatum` houdbaar.
 - Als de `prijs` groter of gelijk is aan 35 euro:
 - Als `bewaarWijn` op `true` staat, dan is de wijn tot 100 jaar vanaf de `brouwDatum` houdbaar.
 - Anders is de wijn tot 10 jaar vanaf de `brouwDatum` houdbaar.
- Override de methode `toString()` en implementeer deze zoals de output hieronder.
 - Voorkom dubbele code en hergebruik zoveel mogelijk.

Gebruik `testStap3()` om je implementatie te testen. De output moet er als volgt uitzien.

```
--- Test stap 3 ---
Château La Tour Carnet 2014: 35.99 euro per fles
Houdbaar tot: 2024-01-16
Bewaarwijn: Nee

Clos Du Clocher 2016: 74.99 euro per fles
Houdbaar tot: 2116-06-08
Bewaarwijn: Ja

Amarosso Verona 2015: 8.75 euro per fles
Houdbaar tot: 2017-04-22
Bewaarwijn: Nee

Domaine Benastra Rouge 2015: 16.5 euro per fles
Houdbaar tot: 2025-10-30
Bewaarwijn: Ja
```

Stap 4: Klasse Voorraad (10 punten)

Maak de klasse `Voorraad` volgens het klassendiagram.

- Let OP: De methodes `printBewaarwijnen()`, `printBedorvenDranken()` en `toString()` zijn pas relevant in Deel II en hoeven nu nog niet geïmplementeerd te worden.

Gebruik `testStap4()` om je implementatie te testen. De output moet er als volgt uitzien.

```
--- Test stap 4 ---
Bieren en wijnen zijn aan de voorraad toegevoegd!
```

Deel II: Gebruik klassen, bestand en database

Stap 5: Sorteer en print alle dranken (10 punten)

Klanten van Slijterij 't Lammetje bladeren vaak door de catalogus die in de winkel hangt, deze moet regelmatig bijgewerkt worden. De catalogus bevat de dranken gesorteerd op alfabetische volgorde. Voer de volgende stappen uit:

- Zorg dat de klasse `Drank` de interface `Comparable` implementeert (zie het klassendiagram) en override vervolgens de methode `compareTo()`.
- Override de methode `toString()` in de klasse `Voorraad` en implementeer deze zoals de output hieronder.
 - Let op: Sorteer de dranken alfabetisch op naam.

Gebruik `testStap5()` om je implementatie te testen. De output moet er als volgt uitzien. In het kader van papierbesparing zijn niet relevante delen van de output vervangen met puntjes, dit moet in je applicatie uiteraard niet het geval zijn.

```
--- Test stap 5 ---
Amarosso Verona 2015: 8.75 euro per fles
Houdbaar tot: 2017-04-22
Bewaarwijn: Nee

Amstel Pilsener: 1.98 euro per fles
Houdbaar tot: 2018-08-30
Alcoholpercentage: 5.0%
Gist in fles: Nee

Château La Tour Carnet 2014: 35.99 euro per fles
...

Clos Du Clocher 2016: 74.99 euro per fles
...

Domaine Benastra Rouge 2015: 16.5 euro per fles
...

Hertog Jan Karakter: 3.66 euro per fles
...

La Trappe Dubbel: 4.28 euro per fles
...
```



Stap 6: Print alle bewaarwijnen (10 punten)

Er is veel vraag naar bewaarwijnen, daarom wil Slijterij 't Lammetje regelmatig een lijst publiceren van alle bewaarwijnen die ze op voorraad hebben:

- Implementeer de methode `printBewaarWijnen()` in de klasse `Voorraad`.

Gebruik `testStap6()` om je implementatie te testen. De output moet er als volgt uitzien (de volgorde van de wijnen in de output kan verschillen):

```
--- Test stap 6 ---
Clos Du Clocher 2016: 74.99 euro per fles
Houdbaar tot: 2116-06-08
Bewaarwijn: Ja

Domaine Benastra Rouge 2015: 16.5 euro per fles
Houdbaar tot: 2025-10-30
Bewaarwijn: Ja
```

Stap 7: Print alle bedorven dranken naar een bestand (10 punten)

Bedorven dranken moeten zo snel mogelijk uit de voorraad gehaald worden, een optie die zo'n lijst uitprint kan daarom niet ontbreken. Deze lijst moet in een bestand gezet worden, zodat de lijst uitgeprint kan worden en meegenomen de winkel in. Bedorven dranken hebben een houdbaarheidsdatum die voor de huidige datum liggen. Tip: Gebruik de methode `isBefore()` van de `LocalDate` klasse. De static methode `now()` van de `LocalDate` klasse geeft de huidige datum.

- Implementeer de methode `printBedorvenDranken()` in klasse `Voorraad`.
- Zorg dat in de methode een tekstbestand gemaakt wordt met de naam 'bedorven_dranken.txt' waarin de bedorven dranken worden weggeschreven. Sla dit bestand op in de folder 'resources' van je project.

Gebruik `testStap7()` om je implementatie te testen. De output in het bestand moet er als volgt uitzien (de volgorde van de wijnen in de output kan verschillen):

```
--- Test stap 7 ---
Amarosso Verona 2015: 8.75 euro per fles
Houdbaar tot: 2017-04-22
Bewaarwijn: Nee

Amstel Pilsener: 1.98 euro per fles
Houdbaar tot: 2018-08-30
Alcoholpercentage: 5.0%
Gist in fles: Nee
```



Stap 8: Haal een lijst met wijnen uit de database (10 punten)

Er is een begin gemaakt met een database genaamd 'Slijterij'. Daarin staat een tabel 'Wijn' met een aantal wijnen. Maak een WijnDAO klasse die het mogelijk maakt om alle wijnen die goedkoper zijn dan een gegeven prijs uit de database te halen. De tabel Wijn heeft een primary key idWijn, die op autonummering staat. De klasse Wijn heeft geen eigenschap idWijn en die hoeft je niet toe te voegen in de klasse. Je hoeft de idWijn daarom niet op te halen uit de tabel, want je hebt de idWijn niet nodig voor het instantiëren van Wijn objecten.

- Maak de klasse WijnDAO met methode `getWijnGoedkoperDanPrijs(double prijs)`. In deze methode moeten de juiste records uit de tabel worden gehaald en als `ArrayList<Wijn>` worden gereturned.
- Gebruik de klasse en de methode om in stap 8 in de main de wijnen uit de database te halen en uit te printen.

Let op de code conventions (10 punten)

- Zorg dat je naam en het doel bij elke klasse bovenin staan (ICC #1).
- Gebruik de juiste inspringing (indentation) bij de lay-out (ICC #2).
- Let op juist gebruik hoofdletters en kleine letters (ICC #3).
- Gebruik goede namen (ICC #4).
- Vermijd magic numbers (ICC#5).
- Voeg waar nodig commentaar toe die inzicht geven in je code (ICC#7).
- Vermijd dode code (ICC #8).
- Denk aan encapsulation (ICC #9).