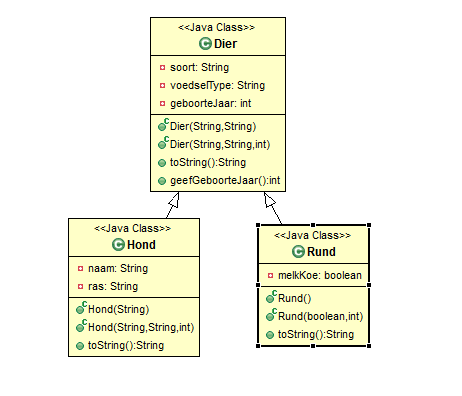
Oefeningen Hoofdstuk 10

**Oefening 1**



Maak op basis van bovenstaand klassediagram de klassen Dier, Hond en Rund aan. Vermijd dubbele code!!!!

Bij het uitvoeren van onderstaand programma

**public** **class** TestDier {

**public** **static** **void** main(String[] args) {

Dier d1 = **new** Dier("olifant", "herbivoor");

Hond h1 = **new** Hond("Lassie");

Hond h2 = **new** Hond("Bessie", "Terrier", 2010);

Rund r1 = **new** Rund(**true**, 2013);

System.***out***.println("d1 : " + d1);

System.***out***.println("h1 : " + h1);

System.***out***.println("h2 : " + h2);

System.***out***.println("r1 : " + r1);

}

}

zou je volgende output moeten krijgen.

d1 : Dit is een olifant (herbivoor).

h1 : Dit is een hond (carnivoor). Het is Lassie de straathond.

h2 : Dit is een hond (carnivoor). Het is Bessie de Terrier.

r1 : Dit is een Rund (herbivoor). Het is een melkkoe

**Klasse Dier**

Zorg ervoor dat bij de creatie van een object een controle gebeurt op de waarde van het voedselType: als dit begint met een ‘h’ of ‘H’, wordt het ingesteld op ‘herbivoor’, begint het met een ‘c’ of ‘C’, dan wordt het ingesteld op ‘carnivoor’, begint het met een ‘o’ of ‘O’, dan wordt het ingesteld op ‘omnivoor’. Is de beginletter een andere dan voorgenoemde, dan wordt het voedselType ingesteld op ‘onbekend’.

Zorg eveneens voor een methode om het voedselType op te vragen en te wijzigen (hierbij gebeurt eveneens voornoemde controle). Deze code komt slechts 1 keer voor in de klasse!

**Oefening 2**

**public** **class** TestDrank {

**public** **static** **void** main(String[] args) {

Drank d1 = **new** Drank("Fanta lemon", 1.6);

Drank d2 = **new** Drank("Fanta light");

Drank d3 = **new** Drank();

System.***out***.println("Gegevens van mijn drankjes ...");

System.***out***.println("mijn eerste drankje " + d1.getNaam());

System.***out***.println(" mijn 2de drankje " + d2.getNaam() + "kost " +   
 d2.getPrijs());

d3.printDrankGegevens();

d3.setNaam("Cola light");

d3.wijzigPrijs(1.7);

d3.printDrankGegevens();

}

}

* Maak op basis van bovenstaand programma de klasse Drank.   
  De prijs van een drankje is minimum 1 euro : bouw hiervoor de nodige controles in in de klasse Drank.
* Bier is een bijzonder soort drank: het heeft bijkomend een alcohol-gehalte (vb 7.3) en een omschrijving (vb: abdijbier). Maak deze klasse.
* Breid de main uit met de nodige opdrachten om een bier-object te creëren en zijn gegevens af te drukken. De uitbreidingen dienen als laatste regels toegevoegd te worden in de main.
* Laat in deze oefening de volgende begrippen aan bod komen : datahiding, inheritance, constructor overloading, method overriding, polymorfisme en static.

**Oefening 3**

Vertrek van de klassen Dier, Hond en Rund uit oefening1.

* Maak van de klasse Dier een abstracte klasse en doe de nodige aanpassingen in de klassen Hond en Rund.
* Maak een abstracte methode maakGeluid() aan in de klasse Dier.
* Maak een uitvoerbaar programma. In de main methode van deze klasse moet het volgende gebeuren: er wordt een array gecreëerd met die 1 hond met naam “Sloeber”, één melkkoe en één vleeskoe bevat.   
  Zorg ervoor dat je volgende output bekomt

Dit is een hond (carnivoor). Het is Sloeber de straathond.

Ik maak volgende geluid WAFWAF

Dit is een Rund (herbivoor). Het is een melkkoe.

Ik maak volgende geluid BEUBEU

Dit is een Rund (herbivoor). Het is een vleeskoe.

Ik maak volgende geluid BEUBEU

**Oefening 3**

Gegeven de klassen Gast en TestHotel. Vertrek van deze klassen om onderstaande te realiseren.

1. Een gast-object kan ook aangemaakt worden met alleen een nummer. De naam en land worden dan beide "onbekend".
2. Het nummer, de naam en het land kunnen als argumenten bij de main meegegeven worden. Declareer een object gast5 op deze manier.
3. Een kamer is een klasse met als kenmerken een nummer en type. Bij de defaultconstructor wordt de nummer van kamer “Z000” en type “standaard”. Alle kenmerken van een kamer kunnen opgevraagd en gewijzigd worden. Op basis van het kamernummer worden volgende gegevens afgedrukt : “blok Z – verdiep 0 – kamer 0 – type standaard”. Definieer de klasse Kamer.
4. Iedere gast heeft steeds één kamer.
5. Breid de main uit waardoor je een kamer-object kan aanmaken dat aan een gast kan toegekend worden. Druk de gegevens van deze gast af.
6. Een gaststudent is een soort gast. Naast een nummer, naam, land volgt hij aan een bepaalde hogeschool een bepaalde opleiding.…. Definieer de klasse GastStudent.
7. Een werknemer is een andere soort gast. Hij heeft als bijkomend kenmerk het beroep en de sector waarin hij tewerkgesteld is. Definieer de klasse GastWerknemer.
8. Voorzie in alle klassen een methode ingaveGegevens() die je gebruikt om alle gegevens van de gasten in te geven.
9. Breid de main uit met de nodige opdrachten om ofwel een student ofwel een werknemer te creëren. De aangemaakte studenten of werknemers worden in een array (max 10) van het type Gast bijgehouden. Na het beëindigen van de ingave, wordt de array overlopen en worden alle gegevens van de gasten getoond.
10. Voorzie een variabele en de nodige methoden om het aantal gecreëerde gasten te kunnen bepalen. Druk het aantal gewone gasten af, het aantal gast-studenten en het aantal gast-werknemers en dit zowel in absolute cijfers en als in procenten.