**Opgave 5: Class diagrams and Java code - Advanced**

**5.1 Domeinmodel**

Maak een domein-klassediagram op basis van de onderstaande kassabon met de aanvullende informatie uit de praktijk van Praxa.  
Ga uit van het domein klassediagram dat we in de opgave 1 (of 3) hebben opgesteld en gebruik daar klassen en associaties uit. Gebruik dezelfde klassennamen.

Onderken en teken op basis van de behandelde leerstof nieuwe domeinklassen, attributen en associaties.

* Ken alle variabelen toe als attributen aan een domeinklasse.
* Specificeer iedere associatie volledig (multiplicity, naam en/of rollen).

PRAXA Amersfoort

Amsterdamseweg 44

Ruilen met bon binnen 8 dagen

In de originele verpakking

|  |  |
| --- | --- |
| 1 \* 159014 Schroeven 4\*25 100st | 8,95 |
| 3 \* 159012 Schroeven 4\*35 100st | 29,85 |
| 1 \* 500154 Bosch boorset | 199,95 |
|  | ====== |
| Totaal incl. BTW | 238,75 |
| BTW: 38,12 |  |
|  |  |
| Betaald met Pinnen | 225,00 |
| Betaald Contant | 15,00 |
| Wisselgeld | 1,25 |

11-08-20xx 14:11 Filiaal 51

Medewerker 12 Kassa 1 Kassabonnr 786549

Aanvullende informatie:

* Deze Bosch boorset is van 9-15/08/20xx afgeprijsd van 239,95 naar 199,95.
* De aankoop is voor een deel contant betaald, maar voor een deel met een pinpas.   
  Iedere (deel)betaling met een bepaalde betaalwijze moet apart vastgelegd worden, onder andere omdat anders geen goede kascontrole kan worden uitgevoerd aan het eind van de dag (hoeveel moet er contant in kas zitten, hoeveel aan retourbonnen, …).
* Momenteel zijn alleen de volgende betaalwijzen toegestaan: contant, pinpas, retourbon.
* Indien er een (deel)betaling voldaan wordt met een pinpas, dan moeten de volgende gegevens ook worden opgeslagen: periodeNummer, transactieNummer, bankId, kaartNummer.

**5.2 Implementatiemodel afleiden uit Java-code**

Bestudeer de onderstaande code van een applicatie voor wielrenners die uit zes klassen bestaat.  
Maak aan de hand ervan een UML Klassediagram en teken de meest passende UML-concepten.  
Pas daarbij de vertaalregels toe die in de les behandeld zijn.

NB Neem wel alle attributen en associaties op, maar je hoeft niet alle methoden op te nemen.

**public class Wielrenner extends Persoon implements Beweegbaar {**

private ArrayList<Fiets> fietsen = new ArrayList<Fiets>();

private Pet pet;

public Wielrenner(String naam, char geslacht,

int leeftijd, double lengte) {

super(naam, geslacht, leeftijd, lengte);

}

public void voegToeFiets(Fiets fiets) {

fietsen.add(fiets);

}

public void verwijderFiets(Fiets fiets) {

fietsen.remove(fiets);

}

public void verwijderFiets(int index) {

fietsen.remove(index);

}

@Override

public void beweegt() {

}

public Pet getPet() {

return pet;

}

public void setPet(Pet pet) {

this.pet = pet;

}

public ArrayList<Fiets> getFietsen() {

return fietsen;

}

public void setFietsen(ArrayList<Fiets> fietsen) {

this.fietsen = fietsen;

}

}

**public class Persoon {**

protected String naam;

protected char geslacht;

protected int leeftijd;

protected double lengte;

public Persoon(String naam, char geslacht,

int leeftijd, double lengte) {

this.naam = naam;

this.geslacht = geslacht;

this.leeftijd = leeftijd;

this.lengte = lengte;

}

public void loopt() {

System.out.println(this.getNaam() + " loopt.");

}

public void lacht() {

System.out.println(this.getNaam() + " lacht.");

}

public String getNaam() {

return naam;

}

public void setNaam(String naam) {

this.naam = naam;

}

public char getGeslacht() {

return geslacht;

}

public void setGeslacht(char geslacht) {

this.geslacht = geslacht;

}

public int getLeeftijd() {

return leeftijd;

}

public void setLeeftijd(int leeftijd) {

this.leeftijd = leeftijd;

}

public double getLengte() {

return lengte;

}

public void setLengte(double lengte) {

this.lengte = lengte;

}

}

**public class Fiets implements Beweegbaar {**

private int serienummer;

private double lengte;

private double hoogte;

private Wiel [] wielen;

public Fiets(int serienummer, double lengte,

double hoogte, double radiusWiel,

String bandtypeWiel) {

this.serienummer = serienummer;

this.lengte = lengte;

this.hoogte = hoogte;

maakWielen(radiusWiel, bandtypeWiel);

}

private void maakWielen(double radius, String bandtype) {

setWielen(new Wiel[2]);

getWielen()[0] = new Wiel(radius, bandtype);

getWielen()[1] = new Wiel(radius, bandtype);

}

@Override

public void beweegt() {

}

public int getSerienummer() {

return serienummer;

}

public void setSerienummer(int serienummer) {

this.serienummer = serienummer;

}

public double getLengte() {

return lengte;

}

public void setLengte(double lengte) {

this.lengte = lengte;

}

public double getHoogte() {

return hoogte;

}

public void setHoogte(double hoogte) {

this.hoogte = hoogte;

}

public Wiel[] getWielen() {

return wielen;

}

public void setWielen(Wiel[] wielen) {

this.wielen = wielen;

}

}

**public class Pet {**

private double lengte;

private String kleur;

public Pet(double lengte, String kleur) {

this.lengte = lengte;

this.kleur = kleur;

}

public double getLengte() {

return lengte;

}

public void setLengte(double lengte) {

this.lengte = lengte;

}

public String getKleur() {

return kleur;

}

public void setKleur(String kleur) {

this.kleur = kleur;

}

}

**public class Wiel implements Beweegbaar {**

private double radius;

private String bandtype;

public Wiel(double radius, String bandtype) {

this.radius = radius;

this.bandtype = bandtype;

}

@Override

public void beweegt() {

}

public double getRadius() {

return radius;

}

public void setRadius(double radius) {

this.radius = radius;

}

public String getBandtype() {

return bandtype;

}

public void setBandtype(String bandtype) {

this.bandtype = bandtype;

}

}

**public interface Beweegbaar {**

public void beweegt();

}