Oefeningen Hoofdstuk 8

**Oefening 1**

1. Maak een klasse Hond a.d.h.v. de volgende informatie:

* Een hond heeft een bepaalde grootte, is van een bepaald ras, heeft een naam en een elektronische chip (String) (het nummer van de elektronische chip moet een ISO-nummer zijn van 15 cijfers. Bv. 962052851260321).
* De grootte van de hond wordt in cm uitgedrukt. Een hond is klein, als de grootte minder dan 70 cm is, anders is de hond groot.
* Een hond kan blaffen, zitten en lopen (= eigenschap activiteit).
* Voorzie een methode blaf(). Als een kleine hond blaft, verschijnt er op het scherm ‘wafwaf’. Als een grote hond blaft, verschijnt er op het scherm ‘WOOAAFF’.   
  Vergeet niet de activiteit aan te passen.
* Voorzie een methode loop(). Als een kleine hond loopt, verschijnt er op het scherm ‘tippel tippel’. Als een grote hond loopt, verschijnt er op het scherm ‘ZOEF ZOEF’.
* Voorzie een methode zit(). Als de hond zit, verschijnt er op het scherm de hond ‘zit’.

Opmerking: de eigenschappen mogen van buiten de klasse niet toegankelijk zijn.

1. Maak de nodige methodes om de eigenschappen van een Hond-object aan te passen en op te vragen.
2. Laat het begrip constructor overloading tot uiting komen in je oplossing. Duid via commentaar lijnen aan in het programma waar je dit begrip toepast.
3. Maak als hoofdprogramma een klasse HondApp die een main bevat.   
   Test al je methoden uit.
4. Druk in de main het aantal hond-objecten af. Zorg ervoor dat je hiervoor gebruik maakt van de klasse Hond. Doe hierin de nodige aanpassingen.

**Oefening 2**

1. Maak een klasse Gsm.
2. Een gsm heeft volgende eigenschappen: gsmnummer (String bv. 0474343434: de voorloopnul moet ook opgenomen worden), bouwjaar, merk, naamEigenaar, voornaamEigenaar, krediet (int), meterbatterij (int) (dit stelt de lading van de batterij voor uitgedrukt in minuten).
3. Het gsmnummer moet bestaan uit 10 cijfers waarvan het eerste een 0 is. Je mag er vanuit gaan dat er 10 cijfers worden ingegeven maar je moet controleren of het eerste cijfer een 0 is en dat er niet meer of minder cijfers worden ingegeven. Indien hier niet aan voldaan is, wordt aan het gsmnummer de waarde “ongeldig” toegekend. Programmeer deze code 1 keer en roep ze op wanneer nodig!
4. Zorg ervoor dat een GSM-object kan gemaakt worden op volgende 3 manieren.

-) door het meegeven van het merk en gsmnummer

-) door het meegeven van het merk, gsmnummer en meterbatterij

-) door geen enkel argument mee te geven (gsmnummer krijgt de waarde onbekend)

1. De gegevens van een GSM-object kunnen afgedrukt worden via de methode toonGSM(). Deze methode geeft de volgende output:

*Het gsmnummer ……… van ……..(naam en voornaam eigenaar) heeft de volgende kenmerken :*

*Bouwjaar : ….*

*Merk : …..*

*Krediet : …..*

*Meterbatterij : ….*

Voorzie in de afdruk dat voor een object aangemaakt via de defaultconstructor (en waarvan de velden nog niet gewijzigd zijn) een eigen boodschap komt :

*Dit object is een leeg object en heeft geen eigen waarden voor zijn datamembers.*

Opmerking: bij het afdrukken van de naam en voornaam van de eigenaar wordt de 1steletter van de naam in hoofdletters weergegeven en de rest in kleine letters. Hetzelfde geldt voor de voornaam.

1. Voorzie voor alle eigenschappen methoden om nieuwe waarden toe te kennen aan de eigenschappen en de waarden van de eigenschappen op te vragen.

Opmerking: wanneer het gsmnummer van een gsm-object ongeldig of onbekend is, moet eerst het gsmnummer gewijzigd worden. Zorg ervoor dat je een mededeling afdrukt, zodra de gebruiker eerst een ander kenmerk van dit gsm-object probeert te wijzigen.

1. Maak volgende methodes

* toonBeltijd() 🡪 de beltijd = krediet x 60 (seconden)
* tick() zonder parameters. Hierdoor neemt de meterbatterij met 2 eenheden af. Zorg ervoor dat je batterij niet negatief kan gaan.
* laadop() zonder parameters. Hierdoor wordt de meter van de batterij onmiddellijk op 200 gezet.

1. Maak een methode getAantalGsm() welke je het aantal tot nu gemaakte GSM-objecten teruggeeft.
2. Maak als hoofdprogramma een klasse GsmApp.jav die een main bevat.

* Maak een array met 3 gsm-objecten aan. Test bovenstaande methoden uit.
* Doorloop deze array en druk de gegevens van elk gsm-object in de array af.
* Druk het aantal gsm-objecten af. Maak hierbij gebruik van de methode getAantalGsm().

**Oefeningen op de klasse Math**

1. Schrijf een programma om  te berekenen. Het getal 41 dient als gegeven ingegeven te worden. Doe een foutencontrole op dit getal dat een oneven getal moet zijn.
2. Schrijf een programma om  te berekenen. x is een geheel getal dat via het toetsenbord als gegeven wordt ingegeven, y is een willekeurig geheel getal (>-10 en <10).  
   Controleer of . Is dit niet het geval, druk dan een gepaste foutmelding af, anders druk je het resultaat af.