



Access fun Grade 8–12 quizzes, matric past papers, K53 learner mock tests, and NBT prep!

*All in one easy-to-use app.*

**DOWNLOAD GO STUDY NOW**



Tap on the buttons above to download the app

 [www.gostudy.club](http://www.gostudy.club)



# basic education

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

## **NASIONALE SENIOR SERTIFIKAAT**

**GRAAD 12**

**INLIGTINGSTEKNOLOGIE V1**

**NOVEMBER 2012**

**MEMORANDUM**

**PUNTE: 120**

Hierdie memorandum bestaan uit 32 bladsye.

## ALGEMENE INLIGTING:

- Hierdie merkriglyne moet as basis gebruik word tydens die merksessies. Hulle is geskep om deur die merkers gebruik te word. Daar word van alle merkers vereis om 'n standaardiseringsvergadering by te woon, om te verseker dat die interpretasie van die riglyne korrek geïmplementeer word tydens die merk van die leerders se antwoordboeke.
- Dit word erken dat daar alternatiewe uitgangspunte bestaan insake die beklemtoning van inligting of detail van die riglyne en dat daar alternatiewe interpretasies of toepassing daarvan is.
- Let daarop dat leerders wat alternatiewe korrekte oplossings as dié gegewe in hierdie riglyne verskaf, volle krediet vir die relevante vraag sal ontvang.
- **BYLAE A, B en C** (bladsye 3–6) bevat die merkersrubriek vir elke vraag ongeag watter een van die programmeringstale gebruik is.
- **BYLAE D, E en F** (bladsye 7–15) bevat die programmeringskode se oplossings vir Delphi vir VRAAG 1 tot 3.
- **BYLAE G, H, I en J** (bladsye 16–28) bevat die programmeringskode se oplossings vir Java vir VRAAG 1 tot 3.
- Kopieë van BYLAE A, B en C (bladsye 3–6) moet gemaak word vir elke leerder om tydens die merksessie voltooi te word.

## BYLAAG A:

### VRAAG 1: MERKRUBRIEK – PROGRAMMERING EN DATABASIS

#### ALGEMENE NOTAS:

- Penaliseer vir verkeerde aantal aanhalings tekens slegs een keer. Herhaalde foutering in volgende antwoorde word nie weer gepenaliseer nie.
- By die gebruik van = strings, die LIKE mag as alternatief gebruik word.

SENTRUMNUMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER PUNT
1.1	<b>Navraag: Korrekte velde (or *) ✓; korrekte tabel ✓; korrekte ORDER BY beide velde ✓</b>	3	
	SQL: SELECT * FROM tblKarnivore ORDER BY FamilieNaam, WetenskaplikeNaam		
1.2	<b>Navraag: Korrekte velde &amp; tabel ✓; korrekte WHERE stelling wat korrekte familie vertoon en gebruik toevoer veranderlike ✓ AND ✓ KampNr begin met ZE ✓ gebruik LIKE ✓</b>	5	
	SQL(D): SELECT WetenskaplikeNaam, AlgemeneNaam, KampNr, KampGrootte FROM tblKarnivore WHERE KampNr LIKE "ZE%" AND FamilieNaam = " + sX + " "		
	Alternatief: ...FamilieNaam LIKE "%' + sX + '%" '		
	In Delphi aanvaar Parameters met SQL.		
	SQL(J): SELECT WetenskaplikeNaam, AlgemeneNaam, KampNr, KampGrootte FROM tblKarnivore WHERE KampNr LIKE 'ZE%' AND FamilieNaam = " + sX + ""  Alternatief: ...FamilieNaam = '%" + sX + "%" ... KampNr like '%ZE%' Left(KampNr,2) = 'ZE'		
1.3	<b>Navraag: Korrekte veld &amp; tabel ✓; COUNT(*) ✓ AS TelDiere ✓; GROUP BY bedreigd ✓</b>	4	
	SQL: SELECT Bedreigd, Count(*) AS TelDiere FROM tblKarnivore GROUP BY Bedreigd  Alternatief ... COUNT(Bedreigd) Moenie penaliseer indien DISTINCT gebruik is nie		



1.4	<b>Navraag: Korrekte veld &amp; tabel ✓; AS SpasiePerDier✓</b> <b>korrek bereken met hakkies✓;</b> <b>ROUND of FORMAT 1 OF 2 des.✓; korrekte WHERE</b> <b>stelling toets AlgemeneNaam vir muishond✓ met</b> <b>LIKE✓</b>	6	
	<p>SQL(D): SELECT KampNr, Format(KampGrootte / (TelVolwassenes+TelKleintjies), "#.0#") AS SpasiePerDier FROM tblKarnivore WHERE AlgemeneNaam LIKE "%muishond"</p> <p>Alternatief: Format(KampGrootte / (TelVolwassenes+TelKleintjies), "#.00") Format(KampGrootte / (TelVolwassenes+TelKleintjies), "0.00") Format(KampGrootte / (TelVolwassenes+TelKleintjies), ".00") Round(KampGrootte / (TelVolwassenes+TelKleintjies), 2)</p> <p>Aanvaar ook: ... ScientificName = "Herpestidae"</p> <p>SQL(J): SELECT KampNr, Format(KampGrootte / (TelVolwassenes+TelKleintjies), "#.0#") AS SpasiePerDier FROM tblKarnivore WHERE AlgemeneNaam LIKE '%muishond%'</p> <p>Alternatief: Format(KampGrootte / (TelVolwassenes+TelKleintjies), '#.00') Format(KampGrootte / (TelVolwassenes+TelKleintjies), '0.00') Format(KampGrootte / (TelVolwassenes+TelKleintjies), '.00') Round(KampGrootte / (TelVolwassenes+TelKleintjies), 2)</p> <p>Aanvaar ook: ... ScientificName = 'Herpestidae'</p>		
1.5	<b>Navraag: UPDATE korrekte tabel ✓; SET korrekte veld✓ met 'n</b> <b>formule wat die waarde met 3 vermeerder ✓; WHERE</b> <b>korrekte KampNr ✓</b>	4	
	<b>NOTA: die gebruik van dieselfde veld aan beide kante van</b> <b>die = teken in die formule.</b>		
	<p>SQL(D): UPDATE tblKarnivore SET TelKleintjies = TelKleintjies + 3 WHERE KampNr = "ZF1"</p> <p>SQL(J): UPDATE tblKarnivore SET TelKleintjies = TelKleintjies + 3 WHERE KampNr = 'ZF1'</p>		

**VRAAG 1: MERKRUBRIEK – PROGRAMMERING EN DATABASIS (vervolg)**

1.6	<p><b>Navraag:</b> <b>SELECT</b> korrekte velde✓; <b>FROM</b> beide tabelle✓; <b>WHERE</b> stelling koppel beide tabelle op KampNr ✓(linkerkant=) ✓(regterkant=); gebruik DAY✓ funksie op BesoekDatum✓; met veranderlike✓</p>		
	<p>SQL(D): 'SELECT tblVeeartsBesoeke.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore, tblVeeartsBesoeke WHERE Day(BesoekDatum)='+sX+' AND tblKarnivore.KampNr = tblVeeartsBesoeke.KampNr'</p> <p>Alternatief: <b>Gebruik 'aliases' vir tabel name:</b> 'SELECT K.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore K, tblVeeartsBesoeke V WHERE Day(BesoekDatum)='+sX+' AND K.KampNr = V.KampNr'</p> <p>Alternatief: <b>Gebruik JOIN notasie:</b> 'SELECT K.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore FROM tblKarnivore INNER JOIN tblVeeartsBesoeke ON tblKarnivore.KampNr = tblVeeartsBesoeke.KampNr WHERE Day(BesoekDatum)='+sX'</p> <p>NOTE: INNER JOIN mag vervang word deur LEFT of RIGHT JOIN</p>	7	
	<p>SQL(J): "SELECT tblVeeartsBesoeke.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore, tblVeeartsBesoeke WHERE Day(BesoekDatum)="+sX+" AND tblKarnivore.KampNr = tblVeeartsBesoeke.KampNr"</p> <p>Alternatief: <b>Gebruik 'aliases' vir tabel name:</b> "SELECT K.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore K, tblVeeartsBesoeke V WHERE Day(BesoekDatum)="+sX+" AND K.KampNr = V.KampNr"</p> <p>Alternatief: <b>Gebruik 'aliases' vir tabel name en korrekte datatipe:</b> "SELECT K.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore K, tblVeeartsBesoeke V WHERE Day(BesoekDatum)= '"+sX+"' AND K.KampNr = V.KampNr"</p> <p>Alternatief: <b>Gebruik JOIN notasie:</b> "SELECT K.KampNr, AlgemeneNaam, BesoekDatum, RedeVirBesoek, Dier_ID FROM tblKarnivore FROM tblKarnivore INNER JOIN tblVeeartsBesoeke ON tblKarnivore.KampNr = tblVeeartsBesoeke.KampNr WHERE Day(BesoekDatum)="+sX'</p> <p>NOTE: INNER JOIN mag vervang word deur LEFT of RIGHT JOIN</p>		

1.7	<b>Navraag:</b> INSERT INTO korrekte tabel✓; lys 5 velde (nie [BesoeKID] <i>autonumber field</i> )✓; VALUES in korrekte volgorde soos gelys in velde✓; datum gebruik #2012/10/25#✓; al die teks velde✓; boolean veld waarde✓  <b>NOTE:</b> indien geen velde gelys word nie maar ses waardes word gegee (1 punt )	6	
	<b>SQL(D):</b> INSERT INTO tblVeeartsBesoeke (BesoeKDatum, KampNr, RedeVirBesoeK, Opvolg, Dier_ID) VALUES (#2012/10/25#, "ZD5", "Oorinfeksie", True, "ZD5_3")  <b>Aanvaar:</b> yes/on/1 in plaas van true Die gebruik van " " vir die datum in die korrekte formaat (short date)		
	<b>SQL(J):</b> INSERT INTO tblVeeartsBesoeke (BesoeKDatum, KampNr, RedeVirBesoeK, Opvolg, Dier_ID) VALUES (#2012/10/25#, 'ZD5', 'Oorinfeksie', True, 'ZD5_3')  <b>Aanvaar:</b> yes/on/1 in plaas van true Die gebruik van ' ' vir die datum in die korrekte formaat (short date)		
	<b>TOTAAL:</b>	<b>35</b>	

**BYLAAG B:**

**VRAAG 2: MERKRUBRIEK – OBJEK-GEORIËNTEERDE PROGRAMMERING**

**ALGEMENE NOTAS:**

- Indien die leerder die datatipe van 'n gegewe veranderlike verander (bv karakter na string) penaliseer met EEN punt
- Penaliseer slegs EEN keer vir 'n spesifieke sintaksfout (bv ;)
- In JAVA word die gebruik van enkel = in plaas van == slegs een keer gepeenaliseer.

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER PUNT
2.1.1	<b>KONSTRUKTOR MET PARAMETERS</b> Vier korrekte parameters✓ met korrekte ooreenstemmende datatipes✓ Toeken van/'set' vier parameters ✓✓  NOTA: indien al die toekennings-stellings in verkeerde volgorde (maks 1 punt)	4	
2.1.2	<b>isGesik METODE:</b> Parameters: aantal diere en grootte kategorie✓ Stuur 'n boolean waarde terug ✓ Toets vir leë kamp ✓; Bereken grootte✓ Toets of gesik vir dier✓ (groot/medium/klein) Maak gebruik van nested IF/case/switch metode met intervale 7, 12 en 18 in logiese formaat✓ (Indien gebruik gemaak word van IF's moet 'n return veranderlike gebruik word) Ken return waarde/antwoord toe✓	7	
2.1.3	<b>toString METODE:</b> Vertoon tipe en kategorie✓; Vertoon twee byskrif ✓; Vertoon twee numeriese waardes: real/double kampgrootte en aantal diere; ✓ Vertoon oor die lyne (Delphi: ...#13;...#10) (JAVA: ...\n...\n...\n)✓  NOTA: Metodes kan geroep word  NOTA – Penaliseer met EEN punt indien: <ul style="list-style-type: none"> <li>• Geen waarde return word</li> <li>• Enige inkorrekte veranderlikes gebruik</li> </ul>	4	
2.2.1	Verklaar <b>objek skikking</b> – grootte 30 of meer✓ teller vir skikking✓	2	

2.2.2	<p><b>INISIALISEERING VAN SKIKKING:</b>  Toets of tekslêer bestaan✓ gebruik 'n IF..THEN..ELSE/  try...catch(FileNotFoundException e)  (Alle stellings moet in die korrekte plek wees)</p> <p><i>Lêer bestaan nie:</i>  Gepaste boodskap✓;  Beeëindig program/event ✓;  <i>Lêer bestaan:</i>  {DELPHI: AssignFile, Reset  JAVA: Skep objek om vanuit lêer te lees} ✓✓;  Inisialiseer skikking teller ✓; (aanvaar ook bt verklaring)  <b>Lus:</b> while not eof/hasNext()✓;  Lees reël uit tekslêer✓;  Onttrek die vier veldwaardes/veranderlikes waardes korrek  (1 punt per veld dmv copy/pos/split/indexOf...) ✓✓✓✓  Inkrementeer skikking teller ✓;  Instansieer objek met konstruktork met parameters✓✓</p> <p>(een punt linkerkant = een punt regterkant. Kontroleer korrekte parameter volgorde van konstruktork.)</p>	15	
2.2.3	<p><b>KEUSELYS OPSIE A:</b>  Opskrif✓; Lus deur skikking✓; Vertoon genommerde kampnr.✓;  Gebruik toString metode en vertoon kampinligting✓</p> <p><b>KEUSELYS OPSIE B:</b>  Sleutel in tipe, aantal, kategorie van dier✓;  Inisialiseer veranderlike (teller) ✓;  Voorwaardelike lus ✓(binne skikking grense AND vlag) ✓;  IF stelling roep <b>isGeskik</b> metode✓ met korrekte volgorde en  ooreenstemmende parameters ✓  Indien gevind:  Stel 3 attribute na korrekte tellerposisie in skikking deur  die set-metodes te gebruik ✓;  Verander vlag✓</p> <p>Inkrementeer teller buite IF (binne die while)✓</p> <p><i>Buite die lus:</i>  Vertoon boodskap met kampnr.✓  (aanvaar ook indien binne die IF...Then binne-in die While-lus)  Vertoon boodskap indien geen geskikte kamp gevind is nie✓</p> <p>NOTA  Die gebruik van die vlag/break/exit in die regte posisie is  aanvaarbaar.  Boodsappe in verkeerde posisie (maksimum 1 punt)</p>	4	
	<b>TOTAAL:</b>	<b>47</b>	

**BYLAAG C:**

**VRAAG 3: MERKRUBRIEK– PROBLEEMOPLOSSINGSPROGRAMMERING**

**ALGEMENE NOTAS:**

- Indien die leerder die datatipe van 'n gegewe veranderlike verander (bv karakter na string) penaliseer met EEN punt
- Penaliseer slegs EEN keer vir 'n spesifieke sintaksfout (bv ;)
- In JAVA word die gebruik van enkel = in plaas van == slegs een keer gepenaliseer.
- In JAVA mag gebruik gemaak word van die Scanner-klas ipv die BufferedReader

**SENTRUMNUMMER:**

**EKSAMENNUMMER:**

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER PUNT
3.1	<p><b>KEUSELYS OPSIE A:</b></p> <p>Lus deur skikking✓;</p> <p>Valideer eerste karakter: A of R ✓</p> <p>Valideer tweede karakter: A, B, C of D ✓</p> <p>Valideer derde karakter: M, V, m of v✓;</p> <p>Korrekte gebruik van logiese operator (AND/OR) in validering✓✓ (of deur gebruik te maak van nested if...then...else of afsonderlike IF stellings of vervang die voorkomste met Z en exit/breakof deur gebruik te maak van vlag/veranderlike)</p> <p>Vertoon ongeldige inskrywings✓;</p> <p>Vervang ongeldige inskrywings met Z in skikking by korrekte indeks✓</p>	8	
3.2	<p><b>KEUSELYS OPSIE B:</b></p> <p>Genereer random-getal✓ Getal in die interval van die aantal skikkings- elemente (1-32 vir Delphi of 0-31 vir Java); ✓</p> <p>Lus✓; Korrekte gebruik van lus-voorwaardes✓; Kontroleer of geldige kaartjie ✓;</p> <p><i>IF geldig:</i> stop die lus ✓;</p> <p><i>IF not geldig:</i> genereer nuwe random getal✓; Vertoon boodskap vir ongeldige inskrywing✓</p> <p><i>Buite lus:</i></p> <p>Vertoon boodskap✓ en wennommer ✓;</p> <p>(aanvaar ook indien binne die IF...Then binne-in die While-lus)</p> <p>Vertoon wenkaartjiestring uit skikking✓</p>	11	

3.3	<p><b>KIESLYS OPSIE C:</b></p> <p>Verklaar datastruktuur/datastrukture✓ vir kategorie elemente✓ van enige datatipes (bv. skikkings vir stoor van uitstallings (AA, AB ...) en punte vir elke uitstalling, nie 32 elemente)</p> <p><i>Berekening van punte:</i></p> <p>Nested loops: Lus deur punte skikking(8 elemente)✓; lus deur arrTic (32 punte)✓; in any order.</p> <p>IF geldige kaartjie✓; kontroleer eerste twee karakters ✓ van kaartjie vs. Kategorie (AA,BB,...)✓; kontroleer vir volwassene/kind✓; Inkrementeer punte korrek in skikking vir korrekte kategorie (AA,BB,...)✓</p> <p><i>Sorteer beide skikkings:</i></p> <p>Nested Lusse✓ met korrekte indekse vir datastruktuur verklaar✓; IF met korrekte voorwaarde✓; Ruil skikkings waardes om✓; herhaal vir tweede omruil gelyktydig✓ korrekte datatipes vir tydelike veranderlike✓</p> <p><i>Vertoon uitslae:</i></p> <p>Opskrifte✓; Vertoon afvoer in kolomme/tabs✓; Vertoon medalje byskrif (Goud/Silwer/Brons) ✓, korrekte kategorie, en ooreenstemmende puntwaarde vir daardie kategorie✓</p>	19	
	<b>TOTAAL:</b>	<b>38</b>	

**OPSOMMING VAN LEERDER SE PUNTE:**

	VRAAG 1	VRAAG 2	VRAAG 3	GROOTTOTAAL
<b>MAKS. PUNTE</b>	<b>35</b>	<b>47</b>	<b>38</b>	<b>120</b>
<b>LEERDER PUNT</b>				



## BYLAAG D: OPLOSSING VIR VRAAG 1: DELPHI

```
unit Vraag1UMemo;
//Oplossing vir Vraag 1 ...
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, DB, ADODB, Grids, DBGrids, ExtCtrls, Buttons, Menus;
type
  TfrmVraagEEN = class(TForm)
    qryRec: TADOQuery;
    dsrQry: TDataSource;
    grdRec: TDBGrid;
    mnuMain: TMainMenu;
    mnuOpsieA: TMenuItem;
    mnuOpsieB: TMenuItem;
    mnuOpsieC: TMenuItem;
    mnuOpsieD: TMenuItem;
    mnuOpsieE: TMenuItem;
    mnuOpsieF: TMenuItem;
    mnuOpsieG: TMenuItem;
    mnuVerlaat: TMenuItem;
    procedure mnuOpsieAClick(Sender: TObject);
    procedure mnuOpsieBClick(Sender: TObject);
    procedure mnuOpsieCClick(Sender: TObject);
    procedure mnuOpsieDClick(Sender: TObject);
    procedure mnuOpsieEClick(Sender: TObject);
    procedure mnuOpsieFClick(Sender: TObject);
    procedure mnuOpsieGClick(Sender: TObject);
    procedure mnuVerlaatClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmVraagEEN: TfrmVraagEEN;
implementation

{$R *.dfm}
//=====
procedure TfrmVraagEEN.mnuOpsieAClick(Sender: TObject);
begin
  qryRec.Close;
  qryRec.SQL.Text := 'SELECT * FROM tblKarnivore ORDER BY FamilieNaam,
WetenskaplikeNaam';
  qryRec.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieBClick(Sender: TObject);
var
  sX : String;
begin
  sX := INPUTBOX('Vraag 1', 'Sleutel Familiennaam in.', 'Canidae');
  qryRec.Close;
  qryRec.SQL.Text := 'SELECT WetenskaplikeNaam, AlgemeneNaam, KampNr,
KampGrootte ' +
                    'FROM tblKarnivore ' +
                    'WHERE (FamilieNaam LIKE "%" +sX + "%") AND (KampNr LIKE "ZE%")';
  qryRec.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieCClick(Sender: TObject);
begin
```



```
gryRec.Close;
gryRec.SQL.Text := 'SELECT Bedreigd, COUNT(TelVolwassenes+TelKleintjies) AS
TelDiere ' +
                  'FROM tblKarnivore ' +
                  'GROUP BY Bedreigd';

gryRec.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieDClick(Sender: TObject);
begin
  gryRec.Close;
  gryRec.SQL.Text := 'SELECT KampNr,
format((KampGrootte/(TelVolwassenes+TelKleintjies)), "#.00") AS SpasiePerDier '
+
                  'FROM tblKarnivore ' +
                  'WHERE AlgemeneNaam LIKE "%muishond%" ';

  gryRec.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieEClick(Sender: TObject);
begin
  gryRec.Close;
  gryRec.SQL.Text := 'UPDATE tblKarnivore ' +
                  'SET TelKleintjies = TelKleintjies + 3 ' +
                  'WHERE KampNr="ZF1"';

  gryRec.ExecSQL;
  MessageDlg('Rekords Suksesvol Verwerk', mtInformation, [mbok], 0);
end;
//=====
procedure TfrmVraagEEN.mnuOpsieFClick(Sender: TObject);
var
  sX : String;
begin
  sX := INPUTBOX('Vraag 1', 'Sleutel die dag van die maand waarin besoek
plaasvind in, bv.', '23');
  gryRec.Close;
  gryRec.SQL.Text := 'SELECT K.KampNr, AlgemeneNaam, BesoekDatum,
RedeVirBesoek, Dier_ID ' +
                  'FROM tblKarnivore K, tblVeeartsBesoeke V ' +
                  'WHERE Day(BesoekDatum)='+sX+' AND K.KampNr = V.KampNr';

  gryRec.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieGClick(Sender: TObject);
begin
  gryRec.Close;
  gryRec.SQL.Text := 'INSERT INTO tblVeeartsBesoeke ' +
                  '(BesoekDatum, KampNr, RedeVirBesoek, Opvolg, Dier_ID) '+
                  'VALUES (#2012/10/25#, "ZD5", "Oorinfeksie", True,
"ZD5_3")';
  gryRec.ExecSQL;
  MessageDlg('Rekords Suksesvol Verwerk', mtInformation, [mbok], 0);
end;
//=====
procedure TfrmVraagEEN.mnuVerlaatClick(Sender: TObject);
begin
  Application.Terminate;
end;
end.
```

## BYLAAG E: OPLOSSING VIR VRAAG 2: DELPHI

### VRAAG 2 – KLASSEENHEID

```
unit uVr2MEMO;
{*** Oplossing vir Vraag 2 - class ***}
interface

TYPE
    TVr2 = class(TObject)
    private
        fDTipe      : String;
        fAantal     : Integer;
        fGrootte    : Real;
        fKat        : Char;

    public
        constructor Create(sDTipe: String; iAantal : integer; rGrootte : Real;
            cKat : Char);
        function isGesik(cKat : char; iAantal : Integer) : Boolean;
        function toString : String;

        procedure setDTipe(sDTipe : String);
        procedure setAantal(iAantal : Integer);
        procedure setGrootte(rGrootte : Real);
        procedure setKat(cKat : Char);
        function getDTipe:String;
        function getAantal:integer;
        function getGrootte:real;
        function getKat:Char;
    end;

implementation

uses SysUtils;

{ TQuest2 }

constructor TVr2.Create(sDTipe: String; iAantal: integer; rGrootte: Real;
    cKat: Char);
begin
    fDTipe      := sDTipe;
    fAantal     := iAantal;
    fGrootte    := rGrootte;
    fKat        := cKat;
end;

function TVr2.isGesik(cKat: char; iAantal: integer): Boolean;
var
    rSpasie : real;
begin
    Result := false;
    if fDTipe = 'XXX'
    then
        begin
            rSpasie := fGrootte / iAantal;
            case cKat of
                'G' : Result := rSpasie >= 18;
                'M' : Result := (rSpasie >= 12) AND (rSpasie < 18);
                'K' : Result := (rSpasie >= 7)  AND (rSpasie < 12);
            end;
        end;
end;
end;
```

```
function TVr2.toString: String;
begin
    Result := fDTipe + '...' + fKat + #13 +
        'Kampgrootte: ' + FloatToStrF(fGrootte, ffFixed,8,1) + #13 +
        'Aantal diere: ' + IntToStr(fAantal) + #13 + #13;
end;

procedure TVr2.setDTipe(sDTipe: String);
begin
    fDTipe := sDTipe;
end;

procedure TVr2.setGrootte(rGrootte: Real);
begin
    fGrootte := rGrootte;
end;

procedure TVr2.setKat(cKat: Char);
begin
    fKat := cKat;
end;

procedure TVr2.setAantal(iAantal: Integer);
begin
    fAantal := iAantal;
end;

function TVr2.getDTipe:String;
begin
    Result := fDTipe;
end;

function TVr2.getAantal:integer;
begin
    Result := fAantal;
end;

function TVr2.getGrootte:real;
begin
    Result := fGrootte;
end;

function TVr2.getKat:Char;
begin
    Result := fKat;
end;

end.
```

## HOOFVORMEENHEID

```
unit Vraag2MEMO;
{*** Oplossing vraag 2 - vorm unit ***}
interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, StdCtrls, ComCtrls, Menus,
    uVr2MEMO;

type
    TfrmQ2 = class(TForm)
        mnuMain: TMainMenu;
```

```
mnuOpsieA: TMenuItem;
mnuVerlaat: TMenuItem;
redQ2: TRichEdit;
mnuOpsieB: TMenuItem;
procedure mnuVerlaatClick(Sender: TObject);
procedure mnuOpsieAClick(Sender: TObject);
procedure mnuOpsieBClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmQ2: TfrmQ2;
  KampArr  : Array[1..30] of TVr2;
  iTeller  : Integer;

implementation

{$R *.dfm}
{$R+}

procedure TfrmQ2.FormCreate(Sender: TObject);
var
  TLeer      : Textfile;
  iPos, iAantal : integer;
  rGrootte   : real;
  cKat       : char;
  sReel, sDier : string;
begin
  if FileExists ('DataV2.txt') <> true then
  begin
    ShowMessage('Die leer bestaan nie.');
```



```
    Exit;
  end;
  AssignFile(TLeer, 'DataV2.txt');
  Reset(TLeer);

  iTeller := 0;
  while (not EOF(TLeer)) do
  begin
    inc(iTeller);
    readln(TLeer, sReel);
    iPos := pos(';', sReel);
    sDier := copy(sReel, 1, iPos - 1);
    delete(sReel, 1, iPos);

    iPos := pos('#', sReel);
    iAantal := StrToInt(copy(sReel, 1, iPos - 1));
    delete(sReel, 1, iPos);

    iPos := pos(':', sReel);
    rGrootte := StrToFloat(copy(sReel, 1, iPos - 1));
    delete(sReel, 1, iPos);

    cKat := sReel[1];

    KampArr[iTeller] := TVr2.Create(sDier, iAantal, rGrootte, cKat);
  end;
  closeFile(TLeer);
end;
```

```
procedure TfrmQ2.mnuOpsieAClick(Sender: TObject);
var
  K :integer;
begin
  redQ2.Lines.Add('Lys van al die kampe');
  redQ2.Lines.Add('=====');
  For K := 1 to iTeller do
    begin
      redQ2.Lines.Add('Kampnommer: ' + IntToStr(K) + #13 +
        KampArr[K].toString);
    end;
  end;

procedure TfrmQ2.mnuOpsieBClick(Sender: TObject);
var
  K,iNom :integer;
  bGevind :boolean;
  cKat    :char;
  sDTipe  :String;
begin
  sDTipe := InputBox('Tipe dier', 'Sleutel die tipe dier in, bv.','Tier');
  iNum := StrToInt(InputBox('Aantal diere', 'Sleutel die aantal diere
in.','2'));
  cKat := InputBox('Kategorie', 'Sleutel die kategorie in (G/M/K)','G')[1];
  bGevind := false;
  K := 1;
  While (bGevind <> true) and (K <= iTeller) do
    begin
      if KampArr[K].isGesik(cKat, iNom) then
        begin
          KampArr[K].setDTipe(sDTipe);
          KampArr[K].setKat(cKat);
          KampArr[K].setAantal(iNom);
          bGevind := true;
        end
      else
        inc(K);
      end;
    redQ2.Lines.Clear;
    if NOT(bGevind) then
      redQ2.Lines.Add('Geen geskikte kamp is gevind nie.')
    else
      begin
        redQ2.Lines.Clear;
        redQ2.Lines.Add('Hierdie diere is in kampnommer ' + IntToStr(K) +
          ' geplaas.');
```



```
        redQ2.Lines.Add(' ');
        mnuOpsieA.Click;
      end;end;

procedure TfrmQ2.mnuVerlaatClick(Sender: TObject);
begin
  Application.Terminate;
end;

end.
```

## BYLAAG F: OPLOSSING VIR VRAAG 3: DELPHI

```
unit Vraag3UMemo;
    //Oplossing vir vraag 3
interface

uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
    Dialogs, Buttons, StdCtrls, ComCtrls, Menus;

type
    TfrmVR3 = class(TForm)
        mnuMain: TMainMenu;
        mnuOpsieA: TMenuItem;
        mnuOpsieB: TMenuItem;
        mnuOpsieC: TMenuItem;
        mnuVerlaat: TMenuItem;
        redV3: TRichEdit;
        procedure mnuOpsieAClick(Sender: TObject);
        procedure mnuVerlaatClick(Sender: TObject);
        procedure mnuOpsieBClick(Sender: TObject);
        procedure mnuOpsieCClick(Sender: TObject);
        procedure FormCreate(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    frmVR3: TfrmVR3;

    arrTic : Array[1..32] of string = ('RCm158', 'ADM33', 'RCv250', 'RAv5',
    'BRM32', 'Adm236', 'RCm23', 'RDM54', 'RCv17', 'RAm12', 'Adm9', 'RCV43',
    'RDm140', 'RDm23', 'ACV113', 'ABv30', 'RDm22', 'ARv38', 'RCV8', 'RAv53',
    'RCv12', 'ABV156', 'ADM31', 'ADM47', 'RAv48', 'ABV246', 'ABv59', 'RRM321',
    'ABm36', 'RCV31', 'RAm445', 'ACn26');

implementation

{$R *.dfm}
{$R+}

uses Math;

var
    //skikkings gebruik tydens oplossing vir Opsie C
    arrUitstal : Array[1..8] of string =
        ('AA', 'AB', 'AC', 'AD', 'RA', 'RB', 'RC', 'RD');
    arrPunte : Array[1..8] of Integer = (0,0,0,0,0,0,0,0);
    arrMedalje : Array[1..3] of string = ('Goud', 'Silwer', 'Brons');

procedure TfrmVR3.FormCreate(Sender: TObject);
begin
    Randomize;
end;

procedure TfrmVR3.mnuOpsieAClick(Sender: TObject);
var
    A : Integer;
begin
    redV3.Lines.Clear;
```

```
redV3.Lines.Add('Ongeldige inskrywings:');
For A := 1 to 32 do
  IF (arrTic[A][1] in ['A','R']) AND
    (arrTic[A][2] in ['A'..'D']) AND
    (Uppcase(arrTic[A][3]) in ['M','V'])
  then //valid ticket
  else
    begin //invalid ticket
      redV3.Lines.Add(arrTic[A]);
      arrTic[A] := 'Z';
    end;
end;

procedure TfrmVR3.mnuOpsieBClick(Sender: TObject);
var
  iKaartjie :integer;
  bGeldig   :boolean;
begin
  bGeldig := false;
  iKaartjie := random(32) + 1;
  while (bGeldig = false) do
    begin
      if arrTic[iKaartjie] = 'Z' then
        begin
          bGeldig := false ;
          redV3.Lines.Add('Ongeldig');
          iKaartjie := random(32) + 1;
        end
      else
        bGeldig := true;
      end;
    end;
  redV3.Lines.Add('Die posisie van die wenkaartjie in die skikking is: ' +
    intToStr(iKaartjie));
  redV3.Lines.Add('Die wenkaartjie: ' + arrTic[iKaartjie] );
end;

procedure TfrmVR3.mnuOpsieCClick(Sender: TObject);
var
  A, D, iPunt, iTemp : Integer;
  sTemp               : string;
begin
  //Bereken die punte vir elke uitstalling:
  for D := 1 to 8 do
    arrPunte[D] := 0;

  for D := 1 to 8 do
    begin
      For A := 1 to 32 do
        begin
          IF pos(arrUitstal[D], arrTic[A]) = 1 //only valid tickets <> Z
          then
            begin
              case arrTic[A][3] of
                'm', 'v' : iPunt := 5;
                'M', 'V' : iPunt := 12;
              end;
              inc(arrPunte[D], iPunt);
            end;
          end; //for A.
        end; //for D.

        //Sorteer die skikkings volgens punte >> ENIGE SORTEERMETODE
        For A := 1 to 8-1 do
```

```
For D := a+1 to 8 do
  IF arrPunte[A] < arrPunte[D]
  then
    begin //ruil elemente van beide skikkings om
      sTemp          := arrUitstal[D];
      arrUitstal[D]   := arrUitstal[A];
      arrUitstal[A]   := sTemp;

      iTemp          := arrPunte[D];
      arrPunte[D]     := arrPunte[A];
      arrPunte[A]     := iTemp;
    end;

//Vertoon uitslae
redV3.Lines.Clear;
redV3.Paragraph.TabCount := 2;
redV3.Paragraph.Tab[0]   := 80;
redV3.Paragraph.Tab[1]   := 200;
redV3.Lines.Add('Uitstallings wat medaljes wen:');
redV3.Lines.Add('Medalje' + #9 + 'Uitstalling' + #9 + 'Punte');
for A := 1 to 3 do
  redV3.Lines.Add(arrMedalje[A] + #9+ arrUitstal[A] + #9+
IntToStr(arrPunte[A]));
end;

procedure TfrmVR3.mnuVerlaatClick(Sender: TObject);
begin
  Application.Terminate;
end;

end.
```





## BYLAAG G: OPLOSSING VIR VRAAG 1: JAVA

```
// Oplossing vir Vraag 1
import java.io.*;
import java.sql.*;
import javax.swing.*;
import java.util.Scanner;

public class ToetsVraag1
{
    public static void main (String[] args) throws SQLException, IOException
    {
        BufferedReader inKb = new BufferedReader (new InputStreamReader
(System.in));
        Dieretuin DB = new Dieretuin();
        System.out.println();

        char keuse = ' ';
        do
        {
            System.out.println("\n\n          KEUSELYS");
            System.out.println();
            System.out.println("      Opsie A");
            System.out.println("      Opsie B");
            System.out.println("      Opsie C");
            System.out.println("      Opsie D");
            System.out.println("      Opsie E");
            System.out.println("      Opsie F");
            System.out.println("      Opsie G");
            System.out.println();
            System.out.println("      V - VERLAAT");
            System.out.println(" ");
            System.out.print("      Jou keuse? ");
            keuse = inKb.readLine().toUpperCase().charAt(0);
            System.out.println(" ");
            String sql = "";
            switch(keuse)
            {
                case 'A':                                     // Vraag 1.1
                {
                    sql = "SELECT * FROM tblKarnivore ORDER BY FamilieNaam,
WetenskaplikeNaam";

                    DB.query(sql);
                    break;
                }
                //=====
                case 'B':                                     // Vraag 1.2
                {
                    System.out.println("Vraag 1: Sleutel Familienaam in
bv.Canidae");

                    String sX = inKb.readLine();
                    sql = "SELECT WetenskaplikeNaam, AlgemeneNaam, KampNr,
KampGrootte FROM tblKarnivore WHERE (FamilieNaam LIKE '%" +sX + "%') AND
(KampNr LIKE 'ZE%')";

                    DB.query(sql);
                    break;
                }
                //=====
                case 'C':                                     // Vraag 1.3
                {
```

```
        sql = "SELECT Bedreigd,
COUNT(TelVolwassenes+TelKleintjies) AS TelDiere FROM tblKarnivore GROUP BY
Bedreigd";

        DB.query(sql);
        break;
    }
//=====
    case 'D': // Vraag 1.4
    {
        sql = "SELECT KampNr,
FORMAT((KampGrootte/(TelVolwassenes+TelKleintjies)), '#.00') AS SpasiePerDier
FROM tblKarnivore WHERE AlgemeneNaam LIKE '%muishond%' ";

        DB.query(sql);
        break;
    }
//=====
    case 'E': // Vraag 1.5
    {
        sql = "UPDATE tblKarnivore SET TelKleintjies =
TelKleintjies + 3 WHERE KampNr = 'ZF1' ";

        DB.query(sql);
        break;
    }
//=====
    case 'F': // Vraag 1.6
    {
        System.out.println("Vraag 1: Sleutel die dag van die maand
waarin besoek plaasvind in, bv. 23");
        String sX = inKb.readLine();
        sql = "SELECT K.KampNr, AlgemeneNaam, BesoekDatum,
RedeVirBesoek, Dier_ID FROM tblKarnivore K, tblVeeartsBesoeke V WHERE
(DAY(BesoekDatum)=" + sX + ") AND K.KampNr = V.KampNr";

        DB.query(sql);
        break;
    }
//=====
    case 'G': // Vraag 1.7
    {
        sql = "INSERT INTO tblVeeartsBesoeke (BesoekDatum, KampNr,
RedeVirBesoek, Opvolg, Dier_ID) VALUES (#2012/10/25#, 'ZD5', 'Oorinfeksie',
True, 'ZD5_3') ";

        DB.query(sql);
        break;
    }
}
}while (keuse != 'V');

DB.disconnect();
System.out.println("Klaar");
}
}
```

## BYLAAG H: OPLOSSING VIR VRAAG 2: JAVA

### VRAAG 2 – KLASSEENHEID

//oplossing vir Vraag 2 - objekklas.

```
public class Vr2Memo
{
    private String tipe;
    private int aantal;
    private double grootte;
    private char kat;

    public Vr2Memo(String tipe, int aantal, double grootte, char kat)
    {
        this.tipe = tipe;
        this.aantal = aantal;
        this.grootte = grootte;
        this.kat = kat;
    }

    public boolean isGeskik(char kat, int aantal)
    {
        boolean geskik = false;
        double spasie = 0;
        if (tipe.equalsIgnoreCase("XXX"))
        {
            spasie = grootte / aantal;

            if (kat == 'G' && spasie >= 18)
                geskik = true;

            if (kat == 'M' && (spasie >= 12 && spasie < 18))
                geskik = true;

            if (kat == 'K' && (spasie >= 7 && spasie < 12))
                geskik = true;
        }
        return geskik;
    }

    public String toString()
    {
        return tipe + "... " + kat + "\nKampgrootte: " + grootte +
        "\nAantal diere: " + aantal + "\n\n";
    }

    public void setTipe(String tipe) {
        this.tipe = tipe;
    }

    public void setAantal(int aantal) {
        this.aantal = aantal;
    }

    public void setGrootte(double grootte) {
        this.grootte = grootte;
    }

    public void setKat(char kat) {
        this.kat = kat;
    }

    public String getTipe() {
        return tipe;
    }
}
```

```
}

public int getAantal() {
    return aantal;
}

public double getGrootte() {
    return grootte;
}

public char getKat() {
    return kat;
}

}
```

## TOETSKLAS (DRYWERKLAS)

```
//Oplossing vir Vraag 2
import java.io.*;
import java.util.Scanner;

public class ToetsVraag2Memo {

    static Vr2Memo[] kampe = new Vr2Memo[30];
    static int tel;

    public static void main(String[] args) throws Exception
    {
        BufferedReader kb = new BufferedReader(new
InputStreamReader(System.in));
        readFile();
        char keuse = ' ';
        do {
            System.out.println("    KEUSELYS");
            System.out.println();
            System.out.println("  Opsie A");
            System.out.println("  Opsie B");
            System.out.println();
            System.out.println("  V - VERLAAT");
            System.out.println("\n  Jou keuse? ");
            keuse=kb.readLine().toUpperCase().charAt(0);
            switch (keuse) {
                case 'A':
                    vertoon();
                    break;
                case 'B':
                    opsieB();
                    break;
                case 'V':
                    System.out.println("Verlaat");
            }
        } while (keuse!= 'V');
    }

    public static void readFile() {
        try
        {
            tel = 0;
            Scanner sc = new Scanner (new FileReader("DataV2.txt"));
            while (sc.hasNext())
            {
                String reel = sc.nextLine();
                int pos1 = reel.indexOf(';',0);
```

```
String aTipe = reel.substring(0,pos1);

int posHash = reel.indexOf('#',pos1);
int aantal = Integer.parseInt(reel.substring(pos1 + 1,posHash));

int pos2 = reel.indexOf(';',posHash);
double grootte = Double.parseDouble(reel.substring(posHash +
1,pos2));

int posHash2 = reel.indexOf('#',pos2);
char kat = reel.charAt(posHash2-1);

kampe[tel] = new Vr2Memo(aTipe, aantal, grootte, kat);
tel++;
}
sc.close();
}

catch (FileNotFoundException e) {
    System.out.println("Leer bestaan nie.");
    System.exit(0);
}

catch (Exception f) {
    System.out.println(f);
}

}

public static void vertoon() {
    System.out.println("Lys van al die kampe");
    System.out.println("=====");

    for (int k = 0; k < tel; k++) {
        System.out.println("Kampnommer: " + (k+1)+"\n" +
kampe[k].toString());
    }
}

private static void opsieB() throws Exception {
    BufferedReader kb = new BufferedReader(new
InputStreamReader(System.in));

    boolean found = false;
    int count = 0;

    System.out.println("Sleutel die tipe dier in, bv. Tier");
    String dier = kb.readLine();
    System.out.println("Sleutel die aantal diere in, bv. 2");
    int aantal = Integer.parseInt(kb.readLine());
    System.out.println("Sleutel die kategorie in (G/M/K)");
    char kat = kb.readLine().charAt(0);

    while (found == false && count < tel)
    {
        if (kampe [count].isGesik(kat, aantal))
        {
            found = true;
            kampe [count].setTipe(dier);
            kampe [count].setAantal(aantal);
            kampe [count].setKat(kat);
        }
        else
            count++;
    }
    if (found == false)
```

```
        {
            System.out.println("Geen geskikte kamp is gevind nie.");
        }
        else
        {
            System.out.println("\n\nHierdie diere is in kampnommer " + (count
+1 ) + " geplaas.");
            System.out.println("\n");
            vertoon();
        }
    }
}
```



## BYLAAG I: OPLOSSING VIR VRAAG 3 MET OOP: JAVA

```
import java.io.IOException;

public class ToetsVraag3Memo
{
    public static void main(String[] args) throws IOException
    {
        Vraag3_Memo toets = new Vraag3_Memo();
        toets.vertoonKeuselys();
    }
}
//Objek klas wat 'n kaartjie beskryf
public class Kaartjie
{
    private String afdeling;
    private String uitstalling;
    private String geslag;
    private int nommer;

    public Kaartjie(String kaartjie)
    {
        afdeling = kaartjie.substring(0,1);
        uitstalling = kaartjie.substring(1,2);
        geslag = kaartjie.substring(2,3);
        nommer = Integer.parseInt(kaartjie.substring(3,4));
    }

    public boolean isGeldig()
    {
        boolean geldig = true;

        if ((("ABCD".indexOf(uitstalling.charAt(0))< 0) || (!(afdeling.equals("A")))
        && (!(afdeling.equals("R")))) || ("MmVv".indexOf(geslag.charAt(0))< 0 ))
            geldig = false;
        return geldig;
    }

    public int getPuntwaarde()
    {
        if (geslag.equals("M") || geslag.equals("V") )
            return 12;
        else
            return 5;
    }

    public String getAfdeling()
    {
        return afdeling;
    }

    public String getUitstalling()
    {
        return uitstalling;
    }

    public String getGeslag()
    {
        return geslag;
    }
}
```

```
public int getNommer()
{
    return nommer;
}
}
//=====
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Vraag3_Memo
{
    String[] arrTic = {"RCm158", "ADM33", "RCv250", "RAv5",
        "BRM32", "ADm236", "RCm23", "RDM54",
        "RCv17", "RAm12", "ADm9", "RCV43",
        "RDm140", "RDm23", "ACV113", "ABv30",
        "RDm22", "ARv38", "RCV8", "RAv53",
        "RCv12", "ABV156", "ADM31", "ADM47",
        "RAv48", "ABV246", "ABv59", "RRM321",
        "ABm36", "RCV31", "RAM445", "ACn26"}; // grootte 32

    String[] arrUitstalling = {"AA", "AB", "AC", "AD", "RA", "RB", "RC", "RD"};
    int[] arrPunte = {0,0,0,0,0,0,0,0};
    String[] arrMedalje = {"Goud", "Silwer", "Brons"};

    BufferedReader inKb;
//=====
// Opsie A
    public void valideer()
    {
        System.out.println("Ongeldige inskrywings");
        for (int tel = 0; tel < 32; tel++) {

            Kaartjie kaartjie = new Kaartjie(arrTic[tel]);
            if (kaartjie.isGeldig() == false) {

                System.out.println(arrTic[tel]);
                arrTic[tel] = "Z";
            }
        }
    }

//=====
//Opsie B
    public void getWenNommer()
    {
        boolean geldig = false;
        int wen = (int)(Math.random() * 32);
        while (geldig == false)
        {
            if(arrTic[wen].equals("Z"))
            {
                wen = (int)(Math.random() * 32);
                System.out.println("Ongeldig");
            }
            else
                geldig = true;
        }
        System.out.println("Die posisie van die wenkaartjie in die skikking
is: " + (wen+1));
    }
}
```



```
        System.out.println("Die wenkaartjie: " + arrTic[wen]);
    }

//=====
// Opsie C
// Identifiseer Medalje Wenners
public void getMedalWenners() throws IOException
{
    //For each display add points:
    for (int d = 0; d < 8; d++) {
        for (int t = 0; t < 32; t++) {
            if( !(arrTic[t].equalsIgnoreCase("Z"))){

                Kaartjie kaartjie = new Kaartjie(arrTic[t]);

                String uitstalKeuse = kaartjie.getAfdeling() +
kaartjie.getUitstalling();

                if (uitstalKeuse.equalsIgnoreCase(arrUitstalling[d])) {
                    arrPunte[d]= arrPunte[d] + kaartjie.getPuntwaarde();
                } //if
            } // if not Z
        } // for kaartjie
    } // for d

//Sorteer die twee skikkings
    for (int a = 0; a < 8 -1; a++) {
        for (int d = (a+1); d < 8; d++) {
            if ( arrPunte[a] < arrPunte[d]){

                String tempUit = arrUitstalling[a];
                arrUitstalling[a]= arrUitstalling[d];
                arrUitstalling[d] = tempUit;

                int tempP = arrPunte[a];
                arrPunte[a] = arrPunte[d];
                arrPunte[d] = tempP;
            } // if
        } // for d
    } // for a
//vertoon medaljes
    System.out.println("Uitstallings wat medaljes wen:");
    System.out.printf("%s%18s%9s\n", "Medalje", "Uitstalling", "Punte");
    for (int a = 0; a < 3; a++) {
        System.out.printf("%-8s%8s%15d\n", arrMedalje[a],
arrUitstalling[a],arrPunte[a]);
    }
} // getMedalWenners

public void vertoonKeuselys() throws IOException
{
    inKb = new BufferedReader (new InputStreamReader (System.in));

    System.out.println();

    char keuse = ' ';
    do
    {
        System.out.println("\n\n          KEUSELYS");
        System.out.println();
        System.out.println("          Opsie A");
        System.out.println("          Opsie B");
        System.out.println("          Opsie C");
    }
```

```
System.out.println();
System.out.println("      V - VERLAAT");
System.out.println(" ");
System.out.print("      Jou keuse? ");
keuse = inKb.readLine().toUpperCase().charAt(0);
System.out.println(" ");
String sql = "";
switch(keuse)
{
    case 'A':
        valideer();
        break;
    case 'B':
        getWenNommer();
        break;
    case 'C':
        getMedalWenners();
        break;
    case 'Q':
        System.out.println("VERLAAT");
        break;
}
}while(keuse != 'V');
}
```



## BYLAAG J: OPLOSSING VIR VRAAG 3 SONDER OOP: JAVA

```
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Vraag3_Memo
{
    String[] arrTic = {"RCm158", "ADM33", "RCv250", "RAv5",
                      "BRM32", "ADm236", "RCm23", "RDM54",
                      "RCv17", "RAm12", "ADm9", "RCV43",
                      "RDm140", "RDm23", "ACV113", "ABv30",
                      "RDm22", "ARv38", "RCV8", "RAv53",
                      "RCv12", "ABV156", "ADM31", "ADM47",
                      "RAv48", "ABV246", "ABv59", "RRM321",
                      "ABm36", "RCV31", "RAm445", "ACn26"};

    String[] arrUitstalling = {"AA", "AB", "AC", "AD", "RA", "RB", "RC", "RD"};
    int[] arrPunte = {0,0,0,0,0,0,0,0};
    String[] arrMedalje = {"Goud", "Silver", "Brons"};

    BufferedReader inKb;

    //=====
    // OPSIE A
    public void valideer()
    {
        System.out.println("Ongeldige inskrywings");
        for (int tel = 0; tel < 32; tel++)
        {
            char eersteKarak = arrTic[tel].charAt(0);
            char tweedeKarak = arrTic[tel].charAt(1); // String tweedeKarak =
arrTic[tel].substring(1,2);
            char derdeKarak = arrTic[tel].charAt(2);

            if ("ABCD".indexOf(tweedeKarak) < 0 || (eersteKarak != 'A' &&
eersteKarak != 'R' ) || ("MmVv".indexOf(derdeKarak) < 0 ))
            {
                System.out.println(arrTic[tel]);
                arrTic[tel] = "Z";
            }
        }
    }

    //=====
    //OPSIE B
    public void getWenNommer()
    {
        boolean geldig = false;
        int wen = (int)(Math.random() * 32);
        while (geldig == false)
        {
            if(arrTic[wen].equals("Z"))
            {
                wen = (int)(Math.random() * 32);
                System.out.println("Ongeldige");
            }
            else
                geldig = true;
        }
        System.out.println("Die posisie van die wenkaartjie in die skikking
is: " + (wen+1));
    }
}
```

```
        System.out.println("Die wenkaartjie: " + arrTic[wen]);
    }

//=====
// OPSIE C
// Identifiseer Medalje Wenners
public void getMedalWenners() throws IOException
{
    // Skryf kode vir Opsie C
    //Vir elke uitstalling bepaal punte:
    for (int d = 0; d < 8; d++) {
        for (int kaartjie = 0; kaartjie < 32; kaartjie++) {
            if( !(arrTic[kaartjie].equalsIgnoreCase("Z"))){
                String uitstalKeuse = arrTic[kaartjie].substring(0,2);
                if (uitstalKeuse.equalsIgnoreCase(arrUitstalling[d])) {
                    char geslag = arrTic[kaartjie].charAt(2);
                    if (geslag == 'v' || geslag == 'm')
                        arrPunte[d]= arrPunte[d] + 5;
                    else
                        arrPunte[d] = arrPunte[d]+ 12;
                } //if
            } // if not Z
        } // for kaartjie
    } // for d

    //Sorteer die twee skikkings
    for (int a = 0; a < (8-1); a++) {
        for (int d = (a+1); d < 8; d++) {
            if ( arrPunte[a] < arrPunte[d]){

                String tempUit = arrUitstalling[a];
                arrUitstalling[a]= arrUitstalling[d];
                arrUitstalling[d] = tempUit;

                int tempP = arrPunte[a];
                arrPunte[a] = arrPunte[d];
                arrPunte[d] = tempP;
            } // if
        } // for d
    } // for a
    //vertoon medaljes
    System.out.println("Uitstallings wat medaljes wen:");
    System.out.printf("%s%18s%9s\n", "Medalje", "Uitstalling", "Punte");
    for (int a = 0; a < 3; a++) {
        System.out.printf("%-8s%8s%15d\n", arrMedalje[a], arrUitstalling[a],
arrPunte[a]);
    }
} // getMedalWenners

public Vraag3_Memo() throws IOException
{
    inKb = new BufferedReader (new InputStreamReader (System.in));

    System.out.println();

    char keuse = ' ';
    do
    {
        System.out.println("\n\n          KEUSELYS");
        System.out.println();
        System.out.println("          Opsie A");
        System.out.println("          Opsie B");
        System.out.println("          Opsie C");
    }
```

```
System.out.println();
System.out.println("      V - VERLAAT");
System.out.println(" ");
System.out.print("      Jou keuse? ");
keuse = inKb.readLine().toUpperCase().charAt(0);
System.out.println(" ");
String sql = "";
switch(keuse)
{
    case 'A':
        valideer();
        break;
    case 'B':
        getWenNommer();
        break;
    case 'C':
        getMedalWenners();
        break;
    case 'V':
        System.out.println("VERLAAT");
        break;
}
}while(keuse != 'V');
}

public static void main(String[] args) throws IOException
{
    new Vraag3_Memo();
}
}
```