



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

FEBRUARIE/MAART 2014

MEMORANDUM

PUNTE: 120

Hierdie memorandum bestaan uit 33 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as basis gebruik word tydens die nasiensessies. Hulle is geskep om deur die nasieners gebruik te word.
- Daar word van alle nasieners vereis om 'n standaardiseringsvergadering by te woon om te verseker dat die interpretasie van die riglyne korrek geïmplementeer word tydens die nasien van die kandidate se praktiese werk.
- Daar word erken dat daar alternatiewe uitgangspunte bestaan in sake die beklemtoning van inligting of detail van die riglyne en dat daar alternatiewe interpretasies of toepassing daarvan is.
- Let daarop dat kandidate wat alternatiewe korrekte oplossings as dié gegee in hierdie riglyne verskaf, volle krediet vir die relevante vraag sal ontvang.
- **Bylae A, B en C** (bladsye 3-7) bevat die nasienrubriek vir elke vraag ongeag watter een van die programmeringstale gebruik is.
- **Bylae D, E, F en G** (bladsye 8-20) bevat die programmeringskode se oplossings vir Delphi vir VRAAG 1 tot 3.
- **Bylae H, I, J en K** (bladsye 21-33) bevat die programmeringskode se oplossings vir Java vir VRAAG 1 tot 3.
- Kopieë van bylaes A, B en C (bladsye 3-7) moet gemaak word vir elke kandidaat om tydens die nasiensessie voltooi te word.

BYLAE A

VRAAG 1: NASIENRUBRIEK – PROGRAMMERING EN DATABASIS

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT SE PUNTE
1.1	Navraag: Korrekte velde (of *) ✓; korrekte tabel ✓; ORDER BY DESC ✓	3	
	SQL: SELECT * FROM tblRespondente ORDER BY VraelysID DESC		
1.2	Navraag: Korrekte veld & tabel ✓; WHERE DatumIngegee ✓; > #korrekte datum# ✓	3	
	SQL: SELECT VraelysID, DatumIngegee, StudentID FROM tblRespondente WHERE DatumIngegee > #2013/08/07# Alternatief: SELECT VraelysID, DatumIngegee, StudentID FROM tblRespondente WHERE Day(DatumIngegee) > 7		
1.3	Navraag: Korrekte veld & tabel ✓; WHERE Stad ✓; LIKE ✓; veranderlike ✓; wildcards ✓; Aantal toestelle >=2 ✓; Internet kontrak ✓;	7	
	SQL (D): SELECT Stad, GetalMobieleToestelle, KonneksieTipe FROM tblRespondente WHERE Stad LIKE '%' + sX + '%' AND GetalMobieleToestelle >= 2 AND InternetKontrak = TRUE SQL (J): SELECT Stad, GetalMobieleToestelle, KonneksieTipe FROM tblRespondente WHERE Stad LIKE '%' + sX + '%' AND GetalMobieleToestelle >= 2 AND InternetKontrak = TRUE Alternatief: yes/on/1 in plaas van true		
1.4	Navraag: Korrekte velde & tabel ✓; FORMAT & #0.00 ✓; Gemiddeld (avg) ✓; AS GemMobielePerStad ✓; GROUP BY Stad ✓	5	
	SQL (D): SELECT Stad, FORMAT(AVG(GetalMobieleToestelle), "#0.00") AS GemMobielePerStad FROM tblRespondente GROUP BY Stad Alternatief: FORMAT(AVG(GetalMobieleToestelle), "#.00") Alternatief: FORMAT(AVG(GetalMobieleToestelle), "0.00") Alternatief: FORMAT(AVG(GetalMobieleToestelle), ".00") Alternatief: ROUND(AVG(GetalMobieleToestelle), 2) SQL (J): SELECT Stad, FORMAT(AVG(GetalMobieleToestelle), '#0.00') AS GemMobielePerStad FROM tblRespondente GROUP BY Stad Alternatief: FORMAT(AVG(GetalMobieleToestelle), '#.00') Alternatief: FORMAT(AVG(GetalMobieleToestelle), '0.00') Alternatief: FORMAT(AVG(GetalMobieleToestelle), '.00') Alternatief: ROUND(AVG(GetalMobieleToestelle), 2)		

VRAAG 1: NASIENRUBRIEK – PROGRAMMERING EN DATABASIS – vervolg

1.5	Navraag: Korrekte velde✓; beide tabelle✓; Count(enige veld)✓; AS GetalVraelyste✓; WHERE klousule wat beide tabelle koppel met StudentID✓; GROUP BY al die velde ✓	6	
	SQL: SELECT Naam, Van, JaarVanStudie, COUNT(*) AS GetalVraelyste FROM tblRespondente, tblStudente WHERE tblRespondente.StudentID = tblStudente.StudentID GROUP BY Naam, Van, JaarVanStudie		
	Alternatief: Gebruik aliases vir tabelname: SELECT Naam, Van, JaarVanStudie, COUNT(*) AS GetalVraelyste FROM tblRespondente R, tblStudente S WHERE R.StudentID = S.StudentID GROUP BY Naam, Van, JaarVanStudie		
	Alternatief: Gebruik JOIN-notasie: SELECT Naam, Van, JaarVanStudie, COUNT(*) AS GetalVraelyste FROM tblRespondente INNER JOIN tblStudente on tblRespondente.StudentID = tblStudente.StudentID GROUP BY Naam, Van, JaarVanStudie		
	NOTA: Mag ook LEFT JOIN of RIGHT JOIN gebruik as alternatief vir INNER JOIN		
1.6	Navraag: UPDATE beide tabelle✓; SET ✓; korrekte berekening✓; WHERE klousule wat beide tabelle koppel met StudentID✓; AND korrekte naam en van✓	5	
	SQL (D): UPDATE tblRespondente, tblStudente SET GetalMobieleToestelle = GetalMobieleToestelle + 1 WHERE tblRespondente.StudentID= tblStudente.StudentID AND Naam = "Kabelo" AND Van = "Mkosi" SQL (J): UPDATE tblRespondente, tblStudente SET GetalMobieleToestelle = GetalMobieleToestelle + 1 WHERE tblRespondente.StudentID= tblStudente.StudentID AND Naam = 'Kabelo' AND Van = 'Mkosi'		
1.7	Navraag: DELETE ✓; korrekte tabel✓; WHERE geen Internet kontrak✓ AND konneksietipe✓; IS✓ NOT NULL✓	6	
	SQL: DELETE FROM tblRespondente WHERE InternetKontrak = False AND KonneksieTipe IS NOT NULL		
	TOTAAL:	35	

BYLAE B**VRAAG 2: NASIENRUBRIEK – OBJEK-GEORIËNTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT SE PUNTE
2.1.1	GEPARAMETERISEERDE KONSTRUKTOR: Korrekte volgorde✓ en datatipe van parameters✓; Ken vier parameters toe✓	3	
2.1.2	berekenGem METODE: Deel voltooide getal vraelyste deur ure ✓; Terugstuur van reële getal✓	2	
2.1.3	toString METODE: Byskrifte ✓; <eoln> of #13-karakter ✓; Vertoon alle attribute korrek (karakter ✓) (numeries ✓) NOTA: Mag privaat attribute/get-metodes gebruik	4	
2.2.1	INISIALISERING VAN SKIKKING: {DELPHI: AssignFile (1 punt), Reset (1 punt) JAVA: Skep objek om uit lêer te lees (1 punt); instansieer objek (1 punt)} ✓✓; Inisialiseer lus teller ✓ Lus om deur lêer te lees✓; Lees 'n reël uit tekslêer ✓; Lees volgende DRIE reëls uit tekslêer✓; Onttrek 'n karakter van die tweede teksreël ✓; Skakel derde teksreël om na 'n heelgetalwaarde ✓; Skakel vierde teksreël om na 'n wisselpuntwaarde ✓; Instansieer objek deur konstruktorkonstruktor met parameters te gebruik: {objek links = ✓; klas regs van =✓; paramaters: tipe en volgorde✓} Verander skikking teller✓; Maak die tekslêer toe✓	14	
2.2.2	KEUSELYSOPSIE A: Opskrif ✓ Lus om deur skikking te lees✓; Vertoon objekte in skikking se data deur van die toString-metode gebruik te maak✓	3	
2.2.3	KEUSELYSOPSIE B: Inisialiseer veranderlikes (gem van beste manlik en vroulik)✓ Lus om deur skikking te stap✓ Gebruik berekenGem-metode✓ IF manlik (M) ✓ AND berekenGem > hoogste manlike gem✓ stoor naam/posisie van hoogste✓ en vervang hoogste manlike gem met nuwe hoogste waarde✓ Herhaal vir vroulik ✓ Afvoer vir beste manlik en beste vroulik (naam✓ en gemid✓); Formateer gemiddeld tot 2 desimale plekke✓	11	

VRAAG 2: NASIENRUBRIEK – OBJEK-GEORIËNTEERDE PROGRAMMERING - vervolg

2.2.4	<p>KEUSELYSOPSIE C: Toevoer van naam, getal voltooid vraelyste en ure✓ Inisialiseer vlag✓; Inisialiseer teller✓ Voorwaardelike lus (toets skikking interval✓ AND vlag) ✓ IF naam gevind ✓ verander vlag✓ stel attribute by korrekte posisie in skikking deur van die set metodes gebruik te maak✓; tel ingesleutelde waardes by✓; maak van get-metodes gebruik om vorige waardes te herwin✓ Inkrementeer lus teller✓</p> <p><i>Buite die lus:</i> Boodskap indien student nie op die lys nie✓</p>	12	
	TOTAAL:	49	

BYLAE C**VRAAG 3: NASIENRUBRIEK – PROBLEEMOPLOSSINGSPROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT SE PUNTE
3.1	<p>Verklaar geskikte datastruktuur (bv. skikking) vir unieke name van speletjies en 'n teller veranderlike ✓; Inisialiseer skikking teller ✓</p> <p><i>Genereer skikking met speletjies se name:</i> Lus om deur skikking te stap (arrData) ✓ Onttrek die naam van die speletjie ✓ (toeken) ✓ (copy/indexOf) ✓ (gebruik posisie van #) Inisialiseer Boolese vlag ✓ Lus om deur speletjies-skikking te stap ✓ Toets speletjiename ✓ vir gelyk aan die skikkingelement ✓ Indien gelyk verander Boolese vlag ✓</p> <p>Buite lus: Indien nie gevind ✓ Inkrementeer die speletjie-skikking teller ✓ Ken die naam van die speletjie aan die speletjie-skikking toe ✓</p> <p><i>Vertoon die name van die speletjies:</i> Lus om deur die speletjies skikking te stap ✓ Vertoon keuselys herhalend ✓ Vertoon die speletjiename ✓ uit skikking ✓</p>	18	
3.2	<p>Aanvaar gebruiker toevoer (lees vanaf sleutelbord) ✓ Herhaal ✓ toevoer van gebruiker totdat geldige waarde ingesleutel is ✓</p> <p><i>Bereken statistiek:</i> Verklaar en inisialiseer tellers vir elke toestel/totaal ✓ Identifiseer korrekte speletjie uit skikking (arrSpeletjies) afhangende van gebruikerstoevoer ✓ Lus deur skikking (arrData) ✓ Toets of die speletjie wat gekies is in arrData-element ✓ Inkrementeer totaal teller ✓; Toets of toestel ✓ deel is van die skikkingselement ✓; Inkrementeer korrekte toestel teller ✓; Herhaal vir al drie die toestelle ✓</p> <p><i>Vertoon statistiek:</i> Opskrif met die naam van die speletjie ✓ wat ingelees is en die totale aantal keer genoem ✓ <i>Bereken persentasie gebruik van elke toestel met formule:</i> Toestel teller / (totale aantal keer genoem) ✓ * 100 ✓; Vertoon gebruik van elke toestel afgerond tot een desimaal ✓ Vertoon saamgestelde inligting in een reël ✓</p>	18	
TOTAAL:		36	

OPSOMMING VAN KANDIDAAT SE PUNTE:

	VRAAG 1	VRAAG 2	VRAAG 3	TOTAAL
MAKS. PUNTE	35	49	36	120
KANDIDAAT SE PUNTE				

BYLAE D: OPLOSSING VIR VRAAG 1: DELPHI

```

unit Vraag1U_MEMO;
  //'n 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)
    qryVR1: TADOQuery;
    dsrQry: TDataSource;
    grdVR1: 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
  qryVR1.Close;
  qryVR1.SQL.Text := 'SELECT * FROM tblRespondente ORDER BY VraelysID DESC';
  qryVR1.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieBClick(Sender: TObject);
begin
  qryVR1.Close;
  qryVR1.SQL.Text := 'SELECT VraelysID, DatumIngegee, StudentID ' +
    'FROM tblRespondente ' +
    'WHERE DatumIngegee > #2013/08/07#';
  qryVR1.Open;
end;
//=====

```


NSS – Memorandum

```

procedure TfrmVraagEEN.mnuOpsieCClick(Sender: TObject);
var
  sX : String;
begin
  sX := INPUTBOX('Vraag 1', 'Sleutel die stad se naam of gedeelte van die naam in',
                'stad');

  qryVR1.Close;
  qryVR1.SQL.Text := 'SELECT Stad, GetalMobieleToestelle, KonneksieTipe ' +
                    'FROM tblRespondente ' +
                    'WHERE Stad LIKE "%" + sX + "%" AND ' +
                    'GetalMobieleToestelle >= 2 AND InternetKontrak = TRUE';

  qryVR1.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieDClick(Sender: TObject);
begin
  qryVR1.Close;
  qryVR1.SQL.Text := 'SELECT Stad, FORMAT(AVG(GetalMobieleToestelle),"0.00") '+
                    'AS GemMobielePerStad ' +
                    'FROM tblRespondente GROUP BY Stad';

  qryVR1.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieEClick(Sender: TObject);
begin
  qryVR1.Close;
  qryVR1.SQL.Text := 'SELECT Naam, Van, JaarVanStudie, COUNT(*) AS GetalVraelyste '+
                    'FROM tblRespondente R, tblStudente S ' +
                    'WHERE (R.StudentID = S.StudentID) '+
                    'GROUP BY Naam, Van, JaarVanStudie';

  qryVR1.Open;
end;
//=====
procedure TfrmVraagEEN.mnuOpsieFClick(Sender: TObject);
begin
  qryVR1.Close;
  qryVR1.SQL.Text := 'UPDATE tblRespondente,tblStudente ' +
                    'SET GetalMobieleToestelle = GetalMobieleToestelle + 1 ' +
                    'WHERE tblRespondente.StudentID = tblStudente.StudentID '+
                    'AND Naam = "Kabelo" AND Van = "Mkosi"';

  qryVR1.ExecSQL;
  MessageDlg('Rekords suksesvol verwerk',mtInformation,[mbOk],0);
end;
//=====
procedure TfrmVraagEEN.mnuOpsieGClick(Sender: TObject);
begin
  qryVR1.Close;
  qryVR1.SQL.Text := 'DELETE FROM tblRespondente '+
                    'WHERE InternetKontrak = False AND ' +
                    'KonneksieTipe IS NOT NULL';

  qryVR1.ExecSQL;
  MessageDlg('Rekords suksesvol verwerk',mtInformation,[mbOk],0);
end;
//=====
procedure TfrmVraagEEN.mnuVerlaatClick(Sender: TObject);
begin
  Application.Terminate;
end;

end.

```

BYLAE E: OPLOSSING VIR VRAAG 2: DELPHI**2.1. KLASSEENHEID**

```
unit uStudent_Memo;
  //'n Oplossing vir Vraag 2 - Klaseenheid.
interface

TYPE
  TStudent = class(TObject)
  private
    fNaam      : String;
    fGeslag    : Char;
    fVraelyste : Integer;
    fUre       : Real;
  public
    constructor Create(sNaam:String; cGeslag:Char; iVraelyste:Integer; rUre:Real);
    function berekenGem : Real;
    function toString   : String;

    function GetNaam      : String;
    function GetGeslag    : Char;
    function GetVraelyste : Integer;
    procedure SetVraelyste(iVraelyste : Integer);
    function GetUre       : Real;
    procedure SetUre(rUre : Real);
  end;

implementation

uses SysUtils;

{ TStudent}

constructor TStudent.Create(sNaam: String; cGeslag: Char;
  iVraelyste: Integer; rUre: Real);
begin
  fNaam      := sNaam;
  fGeslag    := cGeslag;
  fVraelyste := iVraelyste;
  fUre       := rUre;
end;

function TStudent.berekenGem: Real;
begin
  Result := fVraelyste / fUre;
end;

function TStudent.toString: String;
begin
  Result := 'Student: ' + fNaam + ' ('+fGeslag+')' + #13 +
    'Ingesamelde vraelyste: ' + IntToStr(fVraelyste) + #13 +
    'Totale getal ure: ' + FloatToStr(fUre) + #13 + #13;
end;

function TStudent.GetNaam: String;
begin
  Result := fNaam;
end;
```

```
function TStudent.GetGeslag: Char;
begin
  Result := fGeslag;
end;

function TStudent.GetVraelyste: Integer;
begin
  Result := fVraelyste;
end;

procedure TStudent.SetVraelyste(iVraelyste: Integer);
begin
  fVraelyste := iVraelyste;
end;

function TStudent.GetUre: Real;
begin
  Result := fUre;
end;

procedure TStudent.SetUre(rUre: Real);
begin
  fUre := rUre;
end;

end.
```

2.2. HOOFVORMEENHEID – VRAAG 2

```
unit Vraag2U_Memo;
  //'n Oplossing vir Vraag 2 - Hoofvormeenheid.
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ComCtrls, Menus,
  uStudent_Memo;

type
  TfrmV2 = class(TForm)
    mnuMain: TMainMenu;
    mnuOpsieA: TMenuItem;
    mnuVerlaat: TMenuItem;
    redQ2: TRichEdit;
    mnuOpsieB: TMenuItem;
    mnuOpsieC: TMenuItem;
    procedure mnuVerlaatClick(Sender: TObject);
    procedure mnuOpsieAClick(Sender: TObject);
    procedure mnuOpsieBClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure mnuOpsieCClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmV2: TfrmV2;
  arrData : array[1..20] of TStudent;
  iTeller : Integer;
```

implementation

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

procedure TfrmV2.FormCreate(Sender: TObject);
var
  TLeer          : TextFile;
  iVraelyste     : Integer;
  sNaam, sGeslag : String;
  cGeslag        : Char;
  rUre           : Real;
begin
  IF NOT FileExists('Datav2.txt') then
  begin
    MessageDlg('Leer bestaan nie.', mtInformation, [mbOK], 0);
    mnuOpsieA.Visible := False;
    mnuOpsieB.Visible := False;
    mnuOpsieC.Visible := False;
    Exit;
  end;
  AssignFile(TLeer, 'Datav2.txt');
  Reset(TLeer);
  iTeller := 0;
  while NOT EOF(TLeer) Do
  begin
    Readln(TLeer, sNaam);
    Readln(TLeer, sGeslag);
    Readln(TLeer, iVraelyste);
    Readln(TLeer, rUre);
    cGeslag := sGeslag[1];
    Inc(iTeller, 1);
    arrData[iTeller] := TStudent.Create(sNaam, cGeslag, iVraelyste, rUre);
  end;
  CloseFile(TLeer);
end;

procedure TfrmV2.mnuOpsieAClick(Sender: TObject);
var
  A : Integer;
begin
  // Keuselysopsie A
  redQ2.Lines.Clear;
  redQ2.Lines.Add('Lys van studente' + #13);
  for A := 1 to iTeller do
    redQ2.Lines.Add(arrData[A].toString);
end;

procedure TfrmV2.mnuOpsieBClick(Sender: TObject);
var
  A          : Integer;
  rHoogM, rHoogV : Real;
  sNameM, sNameV : String;
begin
  // Keuselysopsie B
  redQ2.Lines.Clear;
  rHoogM := 0;
  rHoogV := 0;
  for A := 1 to iTeller do
  begin
    case arrData[A].GetGeslag of
      'M' : begin
```

```

        IF arrData[a].berekenGem > rHoogM then
        begin
            rHoogM := arrData[a].berekenGem;
            sNameM := arrData[a].GetNaam;
        end;
    end; //m.
    'V' : begin
        IF arrData[a].berekenGem > rHoogV then
        begin
            rHoogV := arrData[a].berekenGem;
            sNameV := arrData[a].GetNaam;
        end;
    end; //if
end; //case
end; //for
redQ2.Lines.Add('Studente met die hoogste gemiddelde waardes:'+#13);
redQ2.Lines.Add('Manlik: ' + sNameM + ' met ' + #13 + 'n gemiddeld van ' +
                FloatToStrF(rHoogM, ffFixed, 8,2));
redQ2.Lines.Add(' ');
redQ2.Lines.Add('Vroulik: ' + sNameV + ' met ' + #13 + 'n gemiddeld van ' +
                FloatToStrF(rHoogV, ffFixed, 8,2));
end;

procedure TfrmV2.mnuOpsieCClick(Sender: TObject);
var
    bGevind      : Boolean;
    A, iVraelyste : Integer;
    rUre         : Real;
    sNaam        : String;
begin
    // Keuselysopsie C
    sNaam := InputBox('Vraag 2', 'Naam van student', 'Eliana');
    iVraelyste := StrToInt(InputBox('Vraag 2', 'Getal voltooide vraelyste
ingesamel', '17'));
    rUre := StrToFloat(InputBox('Vraag 2', 'Getal ure', '1.5'));
    A := 1;
    bGevind := False;
    while (A <= iTeller) AND NOT bGevind do
    begin
        IF arrData[A].GetNaam = sNaam then
        begin
            bGevind := True;
            arrData[A].SetVraelyste(arrData[A].GetVraelyste + iVraelyste);
            arrData[A].SetUre(arrData[A].GetUre + rUre);
        end
        else
            inc(A, 1);
    end; //while
    IF NOT bGevind then
    begin
        redQ2.Lines.Clear;
        redQ2.Lines.Add('Die student is nie op die lys nie');
    end
    else
        mnuOpsieA.Click;
end;

procedure TfrmV2.mnuVerlaatClick(Sender: TObject);
begin
    Application.Terminate;
end;

end.

```

BYLAE F: OPLOSSING VIR VRAAG 3: DELPHI (OOP)**3.1. SPELER KLASEENHEID**

```
// 'n OOP Oplossing vir Vraag 3
unit uSpeler;
interface

type
  TSpeler = class(TObject)
  private
    fSpeletjieNaam : String;
    fToestel       : String;
  public
    constructor Create(sSpeletjieNaam, sToestel : String);
    procedure SetSpeletjieNaam(sSpeletjieNaam : String);
    function  GetSpeletjieNaam : String;
    procedure SetToestel(sToestel : String);
    function  GetToestel : String;
    function  toString      : String;
  end;

implementation

{ TSpeler }
uses SysUtils;

constructor TSpeler.Create(sSpeletjieNaam, sToestel: String);
begin
  fSpeletjieNaam := sSpeletjieNaam;
  fToestel       := sToestel;
end;

function TSpeler.GetSpeletjieNaam: String;
begin
  Result := fSpeletjieNaam;
end;

function TSpeler.GetToestel: String;
begin
  Result := fToestel;
end;

procedure TSpeler.SetSpeletjieNaam(sSpeletjieNaam: String);
begin
  fSpeletjieNaam := sSpeletjieNaam;
end;

procedure TSpeler.SetToestel(sToestel: String);
begin
  fToestel := sToestel;
end;

function TSpeler.toString: String;
begin
  Result := 'Speler {Speletjie = ' + fSpeletjieNaam + ', toestel = ' + fToestel +
  '}' ;
end;

end.
```

3.2. HOOFVORMEENHEID – VRAAG 3

```

unit Vraag3OOP_U;
  //'n OOP Oplossing vir Vraag 3
interface

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

type
  TfrmV3memo = class(TForm)
    mnuMain: TMainMenu;
    mnuOpsieA: TMenuItem;
    mnuVerlaat: TMenuItem;
    redV3: TRichEdit;
    procedure mnuOpsieAClick(Sender: TObject);
    procedure mnuVerlaatClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmV3memo: TfrmV3memo;

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

uses uSpeler;

var
  arrData : array[1..35] of String =
    ('Civilisation#PS3', 'Command & Conquer#PR', 'Solitaire#Xbox',
    'Chess#PR', 'Tetris#PR', 'Chess#PR', 'Command & Conquer#PR',
    'Civilisation#PR', 'SimCity#PR', 'Tetris#PR', 'SimCity#PR',
    'Civilisation#PS3', 'Tetris#PS3', 'Command & Conquer#PS3',
    'SimCity#PR', 'Solitaire#PR', 'Sims#Xbox', 'SimCity#Xbox',
    'Command & Conquer#PR', 'Chess#PS3', 'Tetris#Xbox',
    'Civilisation#Xbox', 'SimCity#PS3', 'Solitaire#PR',
    'Sims#Xbox', 'Command & Conquer#PS3', 'Command & Conquer#PS3',
    'Civilisation#PS3', 'Civilisation#PS3', 'Command & Conquer#Xbox',
    'SimCity#PS3', 'Solitaire#PS3', 'Civilisation#Xbox',
    'Command & Conquer#PR', 'SimCity#PR');

  arrSpeletjies : array[1..20] of String;
  iSTeller      : Integer;

  arrSpelers    : array[1..35] of TSpeler;

procedure SkepSpelerObjekte;
var
  A : Integer;
  sSNaam, sToestel : String;
begin
  for A := 1 to length(arrData) do
    begin
      sSNaam := Copy(arrData[A], 1, Pos('#', arrData[A])-1);
      sToestel := Copy(arrData[A], Pos('#', arrData[A])+ 1, 5);

```

```
        arrSpelers[A] := TSpeler.Create(sSNaam, sToestel);
    end;
end;

procedure SkepSpeletjieSkikking;
var
    A, B      : Integer;
    sSNaam   : String;
    bGevind  : Boolean;
begin
    for A := 1 to length(arrSpeletjies) do
        arrSpeletjies[A] := '';

        iTeller := 0;
        for A := 1 to length(arrData) do
            begin
                sSNaam := arrSpelers[A].GetSpeletjieNaam;
                bGevind := False;
                B := 1;
                while (B < length(arrSpeletjies)) AND NOT bGevind do
                    begin
                        IF sSNaam = arrSpeletjies[B] then
                            bGevind := True
                        else Inc(B, 1);
                    end; //while
                IF NOT bGevind then
                    begin
                        Inc(iTeller, 1);
                        arrSpeletjies[iTeller] := sSNaam;
                    end; //if
                end; //for A
            end;
        end;
    procedure VerwerkData(iSNo : Integer; var iPR, iXbox, iPS3, iTel : Integer);
    var
        A      : Integer;
        sSNaam : String;
    begin
        iPR := 0;
        iPS3 := 0;
        iXbox := 0;
        iTel := 0;
        sSNaam := arrSpeletjies[iSNo];
        for A := 1 to length(arrSpelers) do
            begin
                IF UpperCase(sSNaam) = UpperCase(arrSpelers[A].GetSpeletjieNaam) then
                    begin
                        Inc(iTel, 1);
                        IF UpperCase(arrSpelers[A].getToestel) = 'PR' then inc(iPR, 1);
                        IF UpperCase(arrSpelers[A].getToestel) = 'PS3' then inc(iPS3, 1);
                        IF UpperCase(arrSpelers[A].getToestel) = 'Xbox' then inc(iXbox, 1);
                    end; //if.
                end; //for A.
            end;
        end;

    procedure TfrmV3memo.mnuOpsieAClick(Sender: TObject);
    var
        sSNaam           : String;
        A, iSNom, iPR, iXbox, iPS3, iTel : Integer;
    begin
        redV3.Lines.Clear;
        redV3.Lines.Add('Lys van speletjies');
        redV3.Lines.Add(' ');
    end;
```



```
for A := 1 to iSTeller do
  redV3.Lines.Add(IntToStr(A) + '. ' + #9 + arrSpeletjies[A]);

repeat
  iSNom := StrToInt(InputBox('Vraag 3','Sleutel die nommer van die speletjie
                              in','1'));

  IF NOT(iSNom IN [1..iSTeller]) then
    ShowMessage('Ongeldige toevoer'); // Or MessageDialog ...
until iSNom in [1..iSTeller];

sSNaam := arrSpeletjies[iSNom];
VerwerkData(iSNom, iPR, iXbox, iPS3, iTel);
redV3.Lines.Clear;
redV3.Paragraph.TabCount := 2;
redV3.Paragraph.Tab[0] := 50;
redV3.Paragraph.Tab[1] := 100;
redV3.Lines.Add(sSNaam + ' is ' + IntToStr(iTel) + ' keer genoem. ');
redV3.Lines.Add(' ');
redV3.Lines.Add('Persentasie gebruik vir toestelle: ');
redV3.Lines.Add('PS3' + #9 + 'Xbox' + #9 + 'PR');
redV3.Lines.Add(FloatToStrF(iPS3/iTel*100, ffFixed, 5,1) + '%' + #9 +
                 FloatToStrF(iXbox/iTel*100, ffFixed, 5,1) + '%' + #9 +
                 FloatToStrF(iPR/iTel*100, ffFixed, 5,1) + '%');

end;

procedure TfrmV3memo.mnuVerlaatClick(Sender: TObject);
begin
  Application.Terminate;
end;

procedure TfrmV3memo.FormCreate(Sender: TObject);
begin
  SkepSpelerObjekte;
  SkepSpeletjieSkikking;
end;

end.
```

BYLAE G: OPLOSSING VIR VRAAG 3: DELPHI (Sonder OOP)

```

unit Vraag3U_Memo;
// 'n Oplossing vir Vraag 3 (sonder OOP)
interface

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

type
  TfrmVr3 = class(TForm)
    redV3: TRichEdit;
    mnuMain: TMainMenu;
    mnuOpsieA: TMenuItem;
    mnuVerlaat: TMenuItem;
    procedure mnuOpsieAClick(Sender: TObject);
    procedure mnuVerlaatClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

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

var
  frmVr3: TfrmVr3;
  arrData : array[1..35] of String =
    ('Civilisation#PS3', 'Command & Conquer#PR', 'Solitaire#Xbox',
    'Chess#PR', 'Tetris#PR', 'Chess#PR', 'Command & Conquer#PR',
    'Civilisation#PR', 'SimCity#PR', 'Tetris#PR', 'SimCity#PR',
    'Civilisation#PS3', 'Tetris#PS3', 'Command & Conquer#PS3',
    'SimCity#PR', 'Solitaire#PR', 'Sims#Xbox', 'SimCity#Xbox',
    'Command & Conquer#PR', 'Chess#PS3', 'Tetris#Xbox',
    'Civilisation#Xbox', 'SimCity#PS3', 'Solitaire#PR',
    'Sims#XBox', 'Command & Conquer#PS3', 'Command & Conquer#PS3',
    'Civilisation#PS3', 'Civilisation#PS3', 'Command & Conquer#Xbox',
    'SimCity#PS3', 'Solitaire#PS3', 'Civilisation#Xbox',
    'Command & Conquer#PR', 'SimCity#PR');

  arrSpeletjies : array[1..20] of String;
  iSTeller      : Integer;

procedure SkepSpeletjieSkikking;
var
  A, B : Integer;
  bGevind : Boolean;
  sSNaam : String;
begin
  for A := 1 to length(arrSpeletjies) do
    arrSpeletjies[A] := '';

  iSTeller := 0;
  for A := 1 to length(arrData) do
    begin
      sSNaam := copy(arrData[A], 1, pos('#', arrData[A])-1);
      bGevind := False;
    end;
  end;

```

```

    B := 1;
    while (B < length(arrGames)) AND NOT bGevind do
    begin
        IF sSNaam = arrSpeletjies[B] then
            bGevind := True
        else Inc(B, 1);
    end; //while
    IF NOT bGevind then
    begin
        Ic(iSTeller, 1);
        arrSpeletjies[iSTeller] := sSNaam;
    end; //if
end; //for A
end;

procedure VerwerkData(iSNo : Integer; var iPR, iXbox, iPS3, iTel : Integer);
var
    sSNaam : String;
    A      : Integer;
begin
    iPR := 0;
    iPS3 := 0;
    iXbox := 0;
    iTel := 0;
    sSNaam := arrSpeletjies[iSNo];
    for A := 1 to length(arrData) do
    begin
        IF pos(sSNaam, arrData[A]) > 0 then
        begin
            Inc(iTel, 1);
            IF pos('#PR', arrData[A]) > 0 then inc(iPR, 1);
            IF pos('#PS3', arrData[A]) > 0 then inc(iPS3, 1);
            IF pos('#Xbox', arrData[A]) > 0 then inc(iXbox, 1);
        end; //if
    end; //for A
end;

procedure TfrmVr3.mnuOpsieAClick(Sender: TObject);
var
    sSNaam : String;
    A, iSNom, iPR, iXbox, iPS3, iTel : Integer;
begin
    SkepSpeletjieSkikking;
    redV3.Lines.Clear;
    redV3.Lines.Add('Lys van speletjies');
    redV3.Lines.Add(' ');
    for A := 1 to iSTeller do
        redV3.Lines.Add(IntToStr(A) + ' . ' + #9 + arrSpeletjies[A]);
    repeat
        iSNom := StrToInt(InputBox('Vraag 3', 'Sleutel die nommer van die speletjie op
            die lys in', '1'));

        IF NOT(iSNom IN [1..iSTeller]) then
            ShowMessage('Ongeldige toevoer');
    until iSNom in [1..iSTeller];

    sSNaam := arrSpeletjies[iSNom];

    VerwerkData(iSNom, iPR, iXbox, iPS3, iTel);
    redV3.Lines.Clear;

```

```
redV3.Paragraph.TabCount := 2;
redV3.Paragraph.Tab[0] := 50;
redV3.Paragraph.Tab[1] := 100;
redV3.Lines.Add(sSNaam + ' is ' + IntToStr(iTel) + ' keer genoem.');
```

```
redV3.Lines.Add(' ');
redV3.Lines.Add('Persentasie gebruik vir toestelle:');
```

```
redV3.Lines.Add('PS3' + #9 + 'Xbox' + #9 + 'PR');
```

```
redV3.Lines.Add(FloatToStrF(iPS3/iTel*100, ffFixed, 5,1) + '%' + #9 +
    FloatToStrF(iXbox/iTel*100, ffFixed, 5,1) + '%' + #9 +
    FloatToStrF(iPR/iTel*100, ffFixed, 5,1) + '%');
```

```
end;
```

```
procedure TfrmVr3.mnuVerlaatClick(Sender: TObject);
begin
    Application.Terminate;
end;
```

```
end.
```

BYLAE H: OPLOSSING VIR VRAAG 1: JAVA

```
//'n 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
    {
        Scanner sc = new Scanner(System.in);
        //OF BufferedReader inKb = new BufferedReader (new InputStreamReader
            (System.in));

        Vraag1 DB = new Vraag1();
        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 = sc.nextLine().toUpperCase().charAt(0);
            //OF keuse = inKb.readLine().toUpperCase().charAt(0);
            System.out.println(" ");
            String sql = "";

            switch(keuse)
            {
                case 'A': // Vraag 1.1
                {
                    sql = "SELECT * FROM tblRespondente ORDER BY VraelysID DESC";
                    DB.query(sql);
                    break;
                }
            }
            //=====
            case 'B': // Vraag 1.2
            {
                sql = "SELECT VraelysID, DatumIngegee, StudentID FROM tblRespondente
                    WHERE DatumIngegee > #2013/08/07#";
                DB.query(sql);
                break;
            }
            //=====

```

```

case 'C': // Vraag 1.3
{
    System.out.println("Vraag 1: Sleutel die stad se naam of gedeelte van die
                        naam in. bv. stad ");
    String sX = sc.nextLine();
    //OF String sX = inKb.readLine();

    sql = "SELECT Stad, GetalMobieleToestelle, KonneksieTipe FROM
            tblRespondente WHERE Stad LIKE '%" + sX + "%' AND
            GetalMobieleToestelle >= 2 AND InternetKontrak = TRUE";
    DB.query(sql);
    break;
}
//=====
case 'D': // Vraag 1.4
{
    sql = "SELECT Stad, FORMAT(AVG(GetalMobieleToestelle), '0.00') AS
            GemMobielePerStad FROM tblRespondente GROUP BY Stad";
    DB.query(sql);
    break;
}
//=====
case 'E': // Vraag 1.5
{
    sql = "SELECT Naam, Van, JaarVanStudie, COUNT(*) AS AantOpnames FROM
            tblRespondente, tblStudente WHERE tblRespondente.StudentID =
            tblStudente.StudentID GROUP BY Naam, Van, JaarVanStudie";
    DB.query(sql);
    break;
}
//=====
case 'F': // Vraag 1.6
{
    sql = "UPDATE tblRespondente,tblStudente SET GetalMobieleToestelle =
            GetalMobieleToestelle + 1 WHERE tblRespondente.studentID =
            tblStudente.studentID AND naam = 'Kabelo' AND van = 'Mkosi'";
    DB.query(sql);
    break;
}
//=====
case 'G': // Vraag 1.7
{
    sql = "DELETE FROM tblRespondente WHERE InternetKontrak = False AND
            KonneksieTipe IS NOT NULL";
    DB.query(sql);
    break;
}
}
}while (keuse != 'V');

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

```

BYLAE I: OPLOSSING VIR VRAAG 2: JAVA

2.1. OBJEKLAS

```
// 'n Oplossing vir Vraag 2 - OOP
public class StudentMemo {

    private String naam;
    private char geslag;
    private int vraelyste;
    private double ure;

    public StudentMemo(String naam, char geslag, int vraelyste, double ure) {
        this.naam = naam;
        this.geslag = geslag;
        this.vraelyste = vraelyste;
        this.ure = ure;
    }

    public double berekenGem() {
        return vraelyste/ ure;
    }

    public String toString() {
        return "Student: " + getNaam() + " (" + geslag + ")\nIngesamelde vraelyste:
" + getVraelyste() + "\nTotale getal ure: " + getUre() + "\n";
    }

    public String getNaam() {
        return naam;
    }

    public char getGeslag() {
        return geslag;
    }

    public int getVraelyste() {
        return vraelyste;
    }

    public void setVraelyste(int vraelyste) {
        this.vraelyste = vraelyste;
    }

    public double getUre() {
        return ure;
    }

    public void setUre(double ure) {
        this.ure = ure;
    }
}
```

2.2. TOETS-/DRYWERKLAS – VRAAG 2

```
// 'n Oplossing vir Vraag 2 - OOP
import java.io.*;
import java.text.DecimalFormat;
import java.util.Scanner;

public class ToetsVraag2_Memo {

public static void main(String[] args) throws IOException {

Scanner sc = new Scanner(System.in);
Scanner sf;
// OF BufferedReader bf;
// BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));

int teller = 0;
StudentMemo[] arrData = new StudentMemo[20];

// try {
    sf = new Scanner(new FileReader("DataV2.txt"));

    // OF bf = new BufferedReader(new FileReader("DataV2.txt"));
    // String naam = bf.readLine();
    // while (name != null)
    while (sf.hasNext())
    {
        String naam = sf.nextLine();
        // OF naam = bf.readLine();
        char geslag = sf.nextLine().charAt(0);
        // OF char geslag = bf.readLine().charAt(0);
        int vraelyste = Integer.parseInt(sf.nextLine());
        // OF int vraelyste = Integer.parseInt(bf.readLine());
        double ure = Double.parseDouble(sf.nextLine());
        // OF double ure = Double.parseDouble( bf.readLine());
        arrData[teller] = new StudentMemo(naam, geslag, vraelyste, ure);
        teller++;
    }
    sf.close();
    /* Word benodig as die BufferedReader gebruik word
    bf.close();
    } catch (FileNotFoundException e) {
    System.out.println("Leer bestaan nie.");
    System.exit(0);
    } catch (Exception f) {
        System.out.println(f);
    } */

    char keuse;
    do {
        System.out.println("    KEUSELYS\n");
        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("\nJou keuse? ");
```



```
keuse = sc.nextLine().toUpperCase().charAt(0);
//OF keuse = input.nextLine().toUpperCase().charAt(0);

switch (keuse) {
    case 'A':
        // vertoon skikking deur gebruik te maak van toString metode
        System.out.println("Lys van studente\n");
        for (int tel = 0; tel < teller; tel++) {
            System.out.println(arrData[tel]);
        }
        break;

    case 'B':
        DecimalFormat df = new DecimalFormat("0.00");
        String naamM = "";
        String naamV = "";
        double hoogM = 0;
        double hoogV = 0;
        for (int tel = 0; tel < teller; tel++)
        {
            double gemid = arrData[tel].berekenGem();
            if (arrData[tel].getGeslag() == 'M' && gemid > hoogM)
            {
                hoogM = gemid;
                naamM = arrData[tel].getNaam();
            }
            if (arrData[tel].getGeslag() == 'V' && gemid > hoogV)
            {
                hoogV = gemid;
                naamV = arrData[tel].getNaam();
            }
        }
        System.out.println("Studente met die hoogste gemiddelde waardes:\n");
        System.out.println("Manlik: " + naamM + " met 'n gemiddeld van " +
            df.format(hoogM));
        System.out.println("\nVroulik: " + naamV + " met 'n gemiddeld van " +
            df.format(hoogV) + "\n");
        break;

    case 'C':
        System.out.println("Naam van student: ");
        String naam = sc.nextLine();
        //OF String naam = input.nextLine();

        System.out.println("Getal voltooide vraelyste ingesamel: ");
        String voltooideV = sc.nextLine();
        //OF int voltooideV = input.nextInt();

        System.out.println("Getal ure: ");
        String nuweUre = sc.nextLine();
        //OF int nuweUre = input.nextInt();

        boolean gevind = false;
        int tel = 0;
        while (gevind == false && tel < teller) {
            if (arrData[tel].getNaam().equalsIgnoreCase(naam)) {
                gevind = true;
                int vraelyste = arrData[tel].getVraelyste();
                double ure = arrData[tel].getUre();
                arrData[tel].setVraelyste(vraelyste + Integer.parseInt(voltooideV));
                arrData[tel].setUre(ure + Double.parseDouble(nuweUre));
            }
        }
    }
}
```

```
        else
            tel++;
    } //while
    if (gevind == false) {
        System.out.println("Die student is nie op die lys nie.");
    }
    else {
        System.out.println(arrData[tel].toString());
    }
    break;

    case 'V':
        System.out.println("Verlaat");
        break;
    }
} while (keuse != 'V');
}
```

BYLAE J: OPLOSSING VIR VRAAG 3: JAVA (OOP)**3.1. SPELER-OBJEKKLAS:**

```
// 'n OOP oplossing vir Vraag 3
public class Speler {
    private String speletjie;
    private String toestel;

    public Speler(String speletjie, String toestel) {
        this.speletjie = speletjie;
        this.toestel = toestel;
    }

    public String getToestel() {
        return toestel;
    }

    public void setToestel(String toestel) {
        this.toestel = toestel;
    }

    public String getSpeletjieNaam() {
        return speletjie;
    }

    public void setSpeletjieNaam(String speletjie) {
        this.speletjie = speletjie;
    }

    public String toString() {
        return "Speler {" + "speletjie = " + speletjie + ", toestel = " + toestel +
        "}" ;
    }
}
```

3.2. OPNAMESTATS-OBJEKKLAS

```
//OOP oplossing vir Vraag 3
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.text.DecimalFormat;
import java.util.Scanner;

public class OpnameStats {

    Scanner sc = new Scanner(System.in);
    // OF BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));
    String[] arrSpeletjies = new String [20];
    int sTeller = 0;
    Speler [] arrSpelers = new Speler[35];

    public void skepSpelersSkik(String [] arrData) {
        for (int tel = 0; tel < arrData.length; tel++)
        {
            String[] arrItems = arrData[tel].split("#");
            arrSpelers[tel] = new Speler(arrItems[0], arrItems[1]);
        }
    }
}
```

```

public void skepSpeletjiesSkik()
{
    String speletjieNaam = "";
    int b = 0;
    for (int tel = 0; tel < arrSpelers.length; tel++)
    {
        speletjieNaam = arrSpelers[tel].getSpeletjieNaam();
        boolean gevind = false;
        while (!(gevind) && (b < arrSpeletjies.length))
        {
            if ( speletjieNaam.equalsIgnoreCase(arrSpeletjies[b]))
                gevind = true;
            else
                b++;
        }
        if (!(gevind))
        {
            arrSpeletjies[sTeller] = speletjieNaam;
            sTeller++;
        }
        b = 0;
    }
    sTeller--;
}

public int getSpeletjieTeller()
{
    return sTeller;
}

public int vertoonKeuselys() throws IOException {
    System.out.println("Lys van speletjies:\n");
    int tel;
    for (tel = 0; tel <= sTeller; tel++) {
        System.out.println((tel + 1) + "\t" + arrSpeletjies[tel]);
    }
    int sNom = 0;
    do
    {
        try
        {
            System.out.print("Sleutel die nommer van 'n speletjie op die lys in > ");
            sNom = Integer.parseInt(sc.nextLine());
            // OF int sNom = Integer.parseInt(kb.readLine());
        } catch (NumberFormatException E)
        {
            System.out.println("Ongeldige toevoer");
        }
    }while (!((sNom > 0) && (sNom <= sTeller+1)));
    return sNom;
}

public void getStatistiek(int iKeuse)
{
    String speletjieKeuse = arrSpeletjies[iKeuse];
    int telPS3 = 0;
    int telXbox = 0;
    int telPR = 0;
    double aantal = 0;
}

```

```

    for (int tel = 0; tel < arrSpelers.length; tel++) {
        if (arrSpelers[tel].getSpeletjieNaam().equals(speletjieKeuse)) {
            aantal++;
            String device = arrSpelers[tel].getToestel();
            if (device.indexOf("PS3") >= 0) {
                telPS3++;
            }
            if (device.indexOf("Xbox") >= 0) {
                telXbox++;
            }
            if (device.indexOf("PR") >= 0) {
                telPR++;
            }
        }
    }
    System.out.println(speletjieKeuse + " is " + aantal + " keer genoem." );
    System.out.println("");
    System.out.println("Persentasie gebruik van toestelle:");
    String headings = String.format("%-20s%-20s%-20s", "PS3", "Xbox", "PC");
    String outString = String.format("%-3.1f%-16s%-3.1f%-16s%-3.1f%-16s",
        (telPS3 / aantal * 100), "%", (telXbox / aantal * 100), "%", (telPR /
        aantal * 100), "%");
    System.out.println(headings);
    System.out.println(outString);
    System.out.println();
}
}

```

3.3. TOETSVRAAG3-DRYWERKLAS

```

//\n OOP oplossing vir Vraag 3
import java.io.*;
import java.util.Scanner;

public class toetsVraag3 {

    public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);
        //OF BufferedReader kb = new BufferedReader(new InputStreamReader(System.in));

        String[] arrData = {"Civilisation#PS3", "Command & Conquer#PR",
"Solitaire#Xbox",
        "Chess#PR", "Tetris#PR", "Chess#PR", "Command & Conquer#PR",
        "Civilisation#PR", "SimCity#PR", "Tetris#PR", "SimCity#PR",
        "Civilisation#PS3", "Tetris#PS3", "Command & Conquer#PS3",
        "SimCity#PR", "Solitaire#PR", "Sims#Xbox", "SimCity#Xbox",
        "Command & Conquer#PR", "Chess#PS3", "Tetris#Xbox",
        "Civilisation#Xbox", "SimCity#PS3", "Solitaire#PR", "Sims#Xbox",
        "Command & Conquer#PS3", "Command & Conquer#PS3", "Civilisation#PS3",
        "Civilisation#PS3", "Command & Conquer#Xbox", "SimCity#PS3",
        "Solitaire#PS3", "Civilisation#Xbox", "Command & Conquer#PR",
        "SimCity#PR"};

        OpnameStats obj = new OpnameStats();

        char cKeuse = ' ';
        int sKeuse = 0;

        obj.skepSpelersSkik(arrData);
        obj.skepSpeletjiesSkik();
    }
}

```

```
int speletjieTeller = obj.getSpeletjieTeller();
do {
    System.out.println("=====");
    System.out.println("  KEUSELYS\n");
    System.out.println("Opsie A");
    System.out.println("");
    System.out.println("V - Verlaat");
    System.out.println("\nJou keuse? ");
    cKeuse = sc.nextLine().toUpperCase().charAt(0);
    // OF cKeuse = kb.readLine().toUpperCase().charAt(0);
    switch (cKeuse) {
        case 'A':
            sKeuse = obj.vertoonKeuselys();
            obj.getStatistiek(sKeuse-1);
            break;
        case 'V':
            System.out.println("Verlaat");
            break;
    }
} while (cKeuse != 'V');
}
```

BYLAE K: OPLOSSING VIR VRAAG 3: JAVA (Sonder OOP)

```
//'\n Oplossing vir Vraag 3 sonder OOP

import java.util.Scanner;
import java.text.*;
import java.io.*;
public class ToetsVraag3
{
    public ToetsVraag3()
    {
        SkepSpeletjieSkikking();
        Keuselys();
    }

    public static void main(String[] args)
    {
        new ToetsVraag3();
    }

    Scanner sc = new Scanner(System.in);
    // OF BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));

    String[] arrData = {"Civilisation#PS3", "Command & Conquer#PR", "Solitaire#Xbox",
        "Chess#PR", "Tetris#PR", "Chess#PR", "Command & Conquer#PR",
        "Civilisation#PR", "SimCity#PR", "Tetris#PR", "SimCity#PR",
        "Civilisation#PS3", "Tetris#PS3", "Command & Conquer#PS3",
        "SimCity#PR", "Solitaire#PR", "Sims#Xbox", "SimCity#Xbox",
        "Command & Conquer#PR", "Chess#PS3", "Tetris#Xbox",
        "Civilisation#Xbox", "SimCity#PS3", "Solitaire#PR", "Sims#Xbox",
        "Command & Conquer#PS3", "Command & Conquer#PS3", "Civilisation#PS3",
        "Civilisation#PS3", "Command & Conquer#Xbox", "SimCity#PS3",
        "Solitaire#PS3", "Civilisation#Xbox", "Command & Conquer#PR",
        "SimCity#PR"};

    String[] arrSpeletjies = new String[20];
    int teller = 0;

    public void SkepSpeletjieSkikking()
    {
        for (int k = 0; k < arrData.length;k++)
        {
            boolean gevind = false;
            String[] temp = arrData[k].split("#");
            int toets = 0;
            do
            {
                if (temp[0].equals(arrSpeletjies[toets]))
                {
                    gevind = true;
                }
                else
                {
                    toets++;
                }
            } while ((toets < teller) && (gevind==false));
        }
    }
}
```

```

        if (gevind == false)
        {
            arrSpeletjies[teller] = temp[0];
            teller++;
        }
    }
}

public void Keuselys()
{
    char cKeuse;
    do {
        System.out.println("  KEUSELYS\n");
        System.out.println("Opsie A");
        System.out.println("");
        System.out.println("V - Verlaat");
        System.out.println("\nJou Keuse?  ");
        cKeuse = sc.nextLine().toUpperCase().charAt(0);
        // OF cKeuse = kb.readLine().toUpperCase().charAt(0);
        switch (cKeuse) {
            case 'A':
                System.out.println("Lys van speletjies\n");
                for (int tel = 0; tel < teller; tel++)
                {
                    System.out.println((tel+1) + ". " + arrSpeletjies[tel]);
                }
                int speletjieNom = 0;
                do
                {
                    try
                    {
                        System.out.print("Sleutel die nommer van 'n speletjie op die lys
                                                                                   in > ");

                        speletjieNom = Integer.parseInt(sc.nextLine());
                        // OF speletjieNom = Integer.parseInt(bf.readLine());
                    }
                    catch (NumberFormatException E)
                    {
                        System.out.println("Ongeldige toevoer");
                    }
                }while (!((speletjieNom > 0) && (speletjieNom < teller)));

                if (speletjieNom < teller + 1)
                {
                    String speletjie = arrSpeletjies[speletjieNom-1];
                    int pc = 0;
                    int ps3 = 0;
                    int xbox = 0;
                    for (int k =0; k < arrData.length;k++)
                    {
                        if (arrData[k].toUpperCase().indexOf(speletjie.
                                                                    toUpperCase())== 0)
                        {
                            if (arrData[k].indexOf("PR")>0)
                            {
                                pc++;
                            }
                            if (arrData[k].indexOf("PS3")>0)
                            {
                                ps3++;
                            }
                        }
                    }
                }
            }
        }
    }
}

```



```
        if (arrData[k].indexOf("Xbox")>0)
        {
            xbox++;
        }
    }
    // afvoer
    double tot = pc + ps3 + xbox;
    System.out.println("\n\n");
    System.out.println(speletjie+ " is " + tot + " keer genoem." );
    System.out.println("");
    System.out.println("Persentasie gebruik van toestelle:");
    String headings = String.format("%-20s%-20s%-20s", "PS3",
        "Xbox", "PR");
    String outString = String.format("%-3.1f%-16s%-3.1f%-16s%-
3.1f%-16s", (ps3/tot* 100), "%", (xbox /tot * 100), "%", (pc / tot
* 100), "%");
    System.out.println(headings);
    System.out.println(outString);
    System.out.println("\n");
    }
else
{
    System.out.println("Verlaat");
}

break;
case 'V':
    System.out.println("Verlaat");
break;
}
} while (cKeuse != 'V');
}
}
```