



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIORSERTIFIKAAT-EKSAMEN/ NASIONALE SENIORSERTIFIKAAT-EKSAMEN

INLIGTINGSTEGNOLOGIE V1

2021

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 23 bladsye.

ALGEMENE INLIGTING:


- Hierdie nasienriglyne moet as die basis vir die nasiensessie gebruik word. Dit is voorberei om deur nasieners gebruik te word. Daar word na alle nasieners verwag om 'n deeglike standaardiseringsvergadering by te woon om seker te maak dat die riglyne konsekwent geïnterpreteer en tydens die nasien van die kandidate se werk toegepas word.
- Let op dat leerders wat 'n alternatiewe korrekte oplossing as wat as voorbeeld van 'n oplossing in die nasienriglyne gegee word verskaf, volle krediet vir die relevante oplossing moet kry tensy die spesifieke instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom is nie.
- **Bylaag A, B, C en D** (bladsy 3 tot 10) sluit die nasienrubriek vir elke om te gebruik vir enigeen van die twee programmeringstale in.
- **Bylaag E, F, G en H** (bladsy 11 tot 23) bevat voorbeelde in programmeringskode van oplossings vir VRAAG 1 tot VRAAG 4.
- Kopieë van **Bylaag A, B, C, D en die opsomming van die leerder se punte** (bladsy 3 tot 10) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.



BYLAAG A

VRAAG 1: NASIENRUBRIEK – ALGEMENE PROGRAMMERINGSVAARDIGHEDE

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1	<p>Knoppie [1.1 – Bereken dikte van snye]</p> <p>Verklaar 'n heelgetalveranderlike en 'n reele veranderlike ✓ Stel die 'Stretch' van beeld na 'true' met kode ✓ Verkry getal snye uit 'spin edit' ✓ Dikte van snye = (242 / ✓ getal snye) ✓ Vertoon dikte van snye in lblV1_1 ✓ geformateer tot 2 desimale plekke ✓</p>	7	
1.2	<p>Knoppie [1.2 – Bereken kleingeld]</p> <p>Stel konstante BROOD_PRYE = 12.90 ✓ Verklaar twee reele veranderlikes ✓ Verkry bedrag wat aangebied is uit 'edit box' ✓ omgeskakel na real/float ✓ Kleingeld = Bedrag aangebied ✓ - BROOD_PRYE ✓ As Bedrag aangebied >= BROOD_PRYE ✓ Vertoon Kleingeld ✓ as geldeenheid ('Currency') ✓ anders ✓ Vertoon boodskap OP PANEL ✓ bv. 'Onvoldoende bedrag'</p> <p>Alternatief vir As (If): As Kleingeld >= 0 Alternatief vir anders (else): As Bedrag aangebied < BROOD_PRYE</p>	11	
1.3	<p>Knoppie [1.3 – Veelvoude van 10]</p> <p>Stel Teller vir veelvoude van 10 op 0 ✓ Lus ✓ 10 keer ✓ Genereer ewekansige ✓ getal In die reeks van 50 tot 100 ✓ Vertoon ewekansige getal ✓ Toets of ewekansige getal ✓ veelvoud van 10 is ✓ Inkrementeer Teller ✓ Vertoon Teller met boodskap ✓</p>	10	

1.4	<p>Knoppie [1.4 – Versteekte sekuriteitskode]</p> <p>Kry die posisie van 't' ✓ Terwyl ✓ (lengte van sekuriteitskode < 8) EN ✓ (Pos 't' > 0) ✓ Toets of karakter in posisie -1 ✓ nie ' ' is nie ✓ Voeg karakter in by posisie -1 ✓ by sekuriteitskode ✓ Verwyder karakters uit paragraaf tot by posisie van 't' ✓ Kry posisie van 't' ✓ Vertoon sekuriteitskode ✓ in hoofletters ✓</p> <p>KONSEPTE:</p> <p>Lus(1) vanaf 2 (1) beweeg deur gegewe paragraaf (1) Toets die lengte van die sekuriteitskode < 8 (1) Toets dat die karakter by die indeks(1) = 't' (1) Toets of die karakter by indeks -1 (1) nie 'n spasie is nie (1) Voeg die karakter by indeks -1 (1) by die kode (1)</p> <p>Vertoon die sekuriteitskode (1) in hoofletters (1)</p> <p>NOTA: Die lus kan by 1 begin tot lengte -1 met 'n ander toetsmetode.</p>	12	
	 TOTAAL AFDELING A:	40	

BYLAAG B

VRAAG 2: NASIENRUBRIEK – DATABASISPROGRAMMERING

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	SQL-stellings		
2.1.1	Knoppie [2.1.1 – Lys van winkels] SELECT * ✓ OF noem al die velde FROM tblWinkels ✓ ORDER BY WinkelGrootte DESC ✓	3	
2.1.2	Knoppie [2.1.2 - Bestuurdernommer] SELECT WinkelNaam ✓ FROM tblWinkels ✓ WHERE BestuurderNommer = ' + sBestuurderNom ✓	3	
2.1.3	Knoppie [2.1.3 – Huur > R50 000] SELECT WinkelNaam, WinkelGrootte, Format(65 * WinkelGrootte ✓, "Currency" ✓) AS Rental ✓ FROM tblShops WHERE 65 * WinkelGrootte ✓ > 50000 ✓ Alternatief: GROUP BY WinkelNaam, WinkelGrootte HAVING (WinkelGrootte * 65 > 50000)	5	
2.1.4	Knoppie [2.1.4 – Omset per bestuurder] SELECT BestuurderNaam, BestuurderVan, ✓ FORMAT(SUM(Omset) ✓, "Currency") AS TotaleOmset ✓ FROM tblWinkels W, tblBestuurders B ✓ WHERE W.BestuurderNommer = B.BestuurderNommer ✓ Korrekte aliasse of tabelname GROUP BY BestuurderNaam, BestuurderVan ✓	6	
2.1.5	Knoppie [2.1.5 – Dateer winkelnaam op] UPDATE tblWinkels ✓ SET WinkelNaam = "TeenDream" ✓ WHERE WinkelNaam = "Jeans 4U" ✓	3	
	Subtotaal:	20	

VRAAG 2: NASIENRUBRIEK (VERVOLG)

2.2	DATABASISMANIPULASIE		
2.2.1	Knoppie [2.2.1 – Getal klein winkeltjies] Beweeg na die eerste rekord in tblWinkels ✓ Stel teller as 0 ✓ Lus (Loop) tot einde van tblWinkels ✓ Toets of Winkelgrootte kleiner is as 300 dan ✓ Tel 1 by die teller ✓ Beweeg na volgende rekord in tblWinkels tabel ✓ Vertoon die totale aantal klein winkeltjies (teller) ✓	7	
2.2.2	Knoppie [2.2.2 – Vertoon bestuurder] Stel 'n vlag bGevind op false ✓ Begin by die eerste rekord in tblWinkels ✓ Lus (Loop) tot einde van tblWinkels. ✓ Toets of WinkelNaam in die table dieselfde is as die naam van winkel wat van die toevoer komponent verkry is ✓ Stel die vlag bGevind na True ✓ Ken die bestuurder se nommer in die tblWinkels tabel aan 'n veranderlike (iBestuurder) toe ✓ Beweeg na volgende rekord in tblWinkels ✓ einde (lus)	13	
	Subtotaal:	20	
	TOTAAL AFDELING B:	40	

BYLAAG C

VRAAG 3: NASIENRUBRIEK - OBJEK-GEÛRIENTEERDE PROGRAMMERING

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	<p>Konstruktor-metode:</p> <p>Opskrif (metode verklaring) met drie parameterwaardes ✓ met regte data tipes ✓ Ken naam-parameterwaarde toe aan regte attribuut ✓ Ken balans- en geskenkbewysnommer-parameterwaardes toe aan die regte attribute ✓</p>	4	
3.1.2	<p>getBalance funksie:</p> <p>Funksie-opskrif met reele getalwaarde as terugstuurtipe ✓ fBalans toegekan aan result of funksienaam ✓</p> <p>getGeskenkBNommer-funksie:</p> <p>Funksie-opskrif met heelgetalwaarde as terugstuurtipe ✓ fGeskenkBNommer toegekan aan result ✓ of ken die antwoord toe aan die funksienaam: <i>getVoucherNumber := fVoucherNumber;</i></p>	4	
3.1.3	<p>isVoldoende funksie:</p> <p>Funksie-opskrif/verklaring en booleanwaarde as terugstuurtipe ✓ met parameter van real/float data tipe ✓</p> <p>Toets of parameterwaarde \leq balans-attribuut ✓ result = true ✓ Anders result = false ✓</p> <p>Alternatiewe kode vir toets Stel result op fBalans \geq parameterwaarde (3)</p>	5	
3.1.4	<p>dateerBalansOp-prosedure:</p> <p>Prosedure-opskrif/verklaring wat parameter van real/float data tipe ontvang ✓ Balans = Balans ✓ – parameterwaarde ✓</p>	3	
3.1.5	<p>toString-metode:</p> <p>Drie string-opskrifte (Geskenkbewysnommer, Naam van klant, Beskikbare balans) ✓ en drie attribute waardes ✓ in regte formaat ✓ Stuur string terug ✓</p>	4	
Subtotaal: Objekklas		20	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	<p>Button [Koop]</p> <p>Kry geskenkbewysnommer en naam van redigeerblokkies (edit boxes) ✓ Kry bedrag van kombinasieblokkie (combobox) ✓ <i>Instansieer die objGeskenkBewys objek:</i> objGeskenkBewys:= ✓ TGeskenkBewys.create ✓ Gebruik dire argumente met regte data tipes en regte volgorde ✓ Gebruik toString-metode om inligting van geskenkbewys in richedit-komponent te vertoon ✓</p>	6	
3.2.2 (a)	<p>Knoppie [Vertoon balans]</p> <p>Kry geskenkbewysnommer uit redigeerblokkie (edit box) ✓ Toets of dit die regte geskenkbewys is deur die getGeskenkBNommer te gebruik ✓ Vertoon balans op paneel deur getBalans-metode te gebruik ✓ in geldeenheid (currency) formaat ✓ Stel btnV3_2_2_b op enabled ✓</p>	5	
3.2.2 (b)	<p>Knoppie [Gebruik geskenkbewys]</p> <p>Kry aankoopbedrag uit redigeerblok (edit box) en verander formaat ✓</p> <p>As objGeskenkbewys.isVoldoende(rAankope) ✓ Dateer balans op deur dateerBalansOp-metode te gebruik ✓ Vertoon 'Geskenkbewys is suksesvol gebruik' op etiket (label) ✓ anders Bereken uitstaande bedrag ✓ Vertoon boodskap en uitstaande bedrag in geldeenheidformaat op etiket (label) ✓ Dateer balans op deur die dateerBalansOp-metode✓ te gebruik met getBalans-metode as argument ✓</p> <p>Vertoon die opgedateerde balans op die paneel✓</p>	9	
	Subtotaal: Vormeenheidklas	20	
	TOTAAL AFDELING C:	40	

BYLAAG D

VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
4.1	<p>Knoppie [4.1 – Vul en vertoon skikking]</p> <p>Vertoon opskrifte✓ Verkry naam van inkopiesentrum uit radioknoppie✓ Assign tekslêer met regte inkopiesentrum naam.txt✓ Reset tekslêer✓ Stel ry-teller op 0 ✓ Lus tot by einde van tekslêer ✓ Vermeerder ry-teller ✓ Lees reël uit tekslêer✓ Stoor reël in skikking in regte posisie✓ Stel afvoerstring saam: Ry-teller en punt en reël✓ Voeg spasies in op regte plekke✓ Tot lengte van reël 25 is (of geskikte) karakters✓ Vertoon winkelreël✓ Vertoon afvoerstring✓ en lengte van reël✓ Closefile</p> <p>KONSEPTE: Kry naam van tekslêer (1) Assign en reset (2) Lees inhoud van tekslêer (2) Hantering van die indeks van die skikking (2) Stoor in die skikking (1) Bou afvoerstring op</p> <ul style="list-style-type: none"> • Begin met die lynnommer (1) • Voeg korrekte aantal spasies by (2) • Voeg winkelreël by (1) Vertoon opskrifte en winkels vir elke vloer (3)	15	

VRAAG 4: NASIENRUBRIEK (VERVOLG)

4.2	<p>Knoppie [4.2 – Tel tipe winkel per vlak]</p> <p>Sleutel winkeltipe in deur invoerblok (inputbox) te gebruik ✓ Struktuur om te toets vir geldige winkel letters ✓ Toets of letter ✓ nie geldig is nie ✓ Vertoon foutboodskap ✓ en toon regte letters Anders (indien letter geldig is) Vertoon opskrif wat beskrywing van tipe winkel uit die skikking insluit ✓ Lus (Loop) deur die vlakke ✓ Inisialiseer teller vir winkels per vlak na 0 ✓ Stel afvoerstring op ry-teller ✓ + punt Lus deur die lengte van die stringwaarde in arrWinkels ✓ - geneste lus/ binne lus ✓ Toets of letter gelyk is aan winkeltipe wat ingesleutel is ✓ Vermeerder teller ✓ Voeg winkelteller by afvoerlyn ✓ Vertoon resultate vir die vlak ✓</p> <p>KONSEPTE: Sleutel winkeltipe in (1) Toets of invoer winkeltipe geldig is (1) deur 'n geskikte struktuur/metode te gebruik (2) vertoon boodskap vir verkeerde tipe winkel (1) Vertoon die opskrifte met die winkel tipe (1) Tel die aantal winkels vir elke vlak deur geneste lusse te gebruik (6) Inisialiseer die afvoerstring met vlakteller (1) Voeg winkelteller(1) by en vertoon in buitenste lus (1)</p>	15	
TOTAAL AFDELING D:		30	
GROOTTOTAAL:		150	

OPSOMMING VAN LEERDER SE PUNTE:

SENTRUMNOMMER:		LEERDER SE EKSAMENNOMMER:			
	AFDELING A	AFDELING B	AFDELING C	AFDELING D	
	VRAAG 1	VRAAG 2	VRAAG 3	VRAAG 4	GROOT-TOTAAL
MAKS. PUNTE	40	40	40	30	150
LEERDER SE PUNTE					

BYLAAG E: OPLOSSING VIR VRAAG 1

```
//=====
// Vraag 1.1 7 punte
//=====
procedure TfrmVraag1.btnV1_1Click(Sender: TObject);
var
    iGetalSnye: integer;
    rDikte: real;
begin
    // Vraag 1.1
    imgV1_1.Stretch := true;
    iGetalSnye := spnV1_1.Value;
    rDikte := (242 / iGetalSnye);
    lblV1_1.Caption := FloatToStrF(rDikte, ffFixed, 6, 2);
end;
//=====
// Vraag 1.2 11 punte
//=====
procedure TfrmVraag1.btnV1_2Click(Sender: TObject);
const
    BROOD_PRYS = 12.90;
var
    rBedrag, rKleingeld: real;
begin
    // Vraag 1.2
    rBedrag := StrToFloat(edtV1_2.Text);
    rKleingeld := rBedrag - BROOD_PRYS;
    if rBedrag >= BROOD_PRYS then // Alternatief: if rKleingeld >= 0 ...
        pnlV1_2.Caption := 'Kleingeld: ' + FloatToStrF(rKleingeld,
            ffCurrency, 8, 2)
    else
        pnlV1_2.Caption := 'Onvoldoende bedrag aangebied';
    // Alternatief: if in plaas van else
end;
//=====
// Vraag 1.3 10 punte
//=====
procedure TfrmVraag1.btnV1_3Click(Sender: TObject);
var
    iLusTeller, iTelVeelv10: integer;
    iRandomGetal: integer;
begin
    // Vraag 1.3
    redV1_3.Clear;
    iTelVeelv10 := 0;
    for iLusTeller := 1 to 10 do
    begin
        iRandomGetal := RandomRange(50, 101); // of alternatiewe maniere
        redV1_3.Lines.Add(IntToStr(iRandomGetal));
        if iRandomGetal MOD 10 = 0 then
        begin
            inc(iTelVeelv10);
        end;
    end;

    redV1_3.Lines.Add(#13 + 'Getal veelvoude van 10: ' +
        IntToStr(iTelVeelv10));
end;
```

```
//=====
// Vraag 1.4 12 punte
//=====
procedure TfrmVraag1.btnV1_4Click(Sender: TObject);
var
  sParagraaf: String;
  sSekuriteitsKode: String;
  iPos: integer;
  // I: integer; // nodig vir alternatiewe oplossing

begin
  // Vraag 1.4

  // Kode wat voorsien is
  sParagraaf :=
    sParagraaf := 'I am not lazy, I am just very relaxed. He who laughs
      last did not get the joke. When nothing is going right, go left. '
    + 'I love school when it is vacation. I put the "Pro" in
      procrastinate.';

  // sParagraaf := 'My Grade 12 year is always going to be my greatest as
  it provides me with the most opportunities.';

  sSekuriteitsKode := '';
  iPos := pos('t', sParagraaf);
  while (length(sSekuriteitsKode) < 8) AND ( iPos <> 0) do
  begin
    if sParagraaf[iPos - 1] <> ' ' then
      sSekuriteitsKode := sSekuriteitsKode + sParagraaf[iPos - 1];
    Delete(sParagraaf, 1, iPos);
    iPos := pos('t', sParagraaf);
  end;
  edtV1_4.Text := uppercase(sSekuriteitsKode);

  // Alternatiewe oplossing
  // for I := 2 to length(sParagraaf) do
  // begin
  //   if (sParagraaf[I] = 't') AND (sParagraaf[I - 1] <> ' ') AND
  //     (length(sSekuriteitsKode) < 8) then
  //     sSekuriteitsKode := sSekuriteitsKode + sParagraaf[I - 1];
  // end;
  // edtV1_4.Text := uppercase(sSekuriteitsKode);
end;

end.
```

BYLAAG F: OPLOSSING VIR VRAAG 2

```
unit Vraag2_U;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  
Forms, Dialogs, StdCtrls, Buttons, ExtCtrls, ConnectDB_U, DB, ADODB,  
Grids, DBGrids, ComCtrls, DateUtils, DBCtrls;
```

```
type
```

```
TfrmDBVraag2 = class(TForm)
  pnlBtns: TPanel;
  bmbClose: TBitBtn;
  bmbHerstelDB: TBitBtn;
  pgcDBAdmin: TPageControl;
  tabsQ2SQL: TTabSheet;
  btnV2_1_1: TBitBtn;
  btnV2_1_3: TBitBtn;
  btnV2_1_2: TBitBtn;
  btnV2_1_4: TBitBtn;
  bmbV2_1_5: TBitBtn;
  grpV2_2_1: TGroupBox;
  grpResultate: TGroupBox;
  dbgrdSQL: TDBGrid;
  grpV2_1_3: TGroupBox;
  pnlVDB: TPanel;
  cmbV2_1_2: TComboBox;
  redV2_2_1: TRichEdit;
  Label1: TLabel;
  btnV2_2_1: TButton;
  grpV2_2_2: TGroupBox;
  btnV2_2_2: TButton;
  redV2_2_2: TRichEdit;
  b: TTabSheet;
  grpBestuurders: TGroupBox;
  grpWinkels: TGroupBox;
  dbgrdONE: TDBGrid;
  dbgrdMany: TDBGrid;
  procedure bmbHerstelDBClick(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure FormClose(Sender: TObject; var Action: TCloseAction);
  procedure btnV2_1_1Click(Sender: TObject);
  procedure btnV2_1_3Click(Sender: TObject);
  procedure btnV2_1_2Click(Sender: TObject);
  procedure btnV2_1_4Click(Sender: TObject);
  procedure bmbV2_1_5Click(Sender: TObject);
  procedure btnV2_2_2Click(Sender: TObject);
  procedure btnV2_2_1Click(Sender: TObject);
private
public
end;
```

```
var
```

```
frmDBVraag2: TfrmDBVraag2;  
dbCONN: TConnection;
```



```
// --- Globale veranderlikes om te gebruik ---
tblWinkels, tblBestuurders: TADOTable;

implementation
{$R *.dfm}
{$R+}
//=====
// Vraag 2.1 - SQL afdeling
//=====

//=====
// Vraag 2.1.1          3 punte
//=====
    sSQL1 := 'SELECT * FROM tblWinkels ORDER BY WinkelGrootte DESC';

//=====
// Vraag 2.1.2          3 punte
//=====
    sSQL2 := 'SELECT WinkelNaam FROM tblWinkels WHERE BestuurderNommer
            = ' + sBestuurderNom;

//=====
// Vraag 2.1.3          5 punte
//=====
    sSQL3 := 'SELECT WinkelNaam, WinkelGrootte,
            Format(65 * WinkelGrootte,"Currency") AS Huur FROM tblWinkels
            WHERE 65 * WinkelGrootte > 50000';

//=====
// Vraag 2.1.4          6 punte
//=====
    sSQL4 := 'SELECT BestuurderNaam, BestuurderVan,
            Format(SUM(Omset),"Currency")
            AS [TotaleOmset] FROM tblWinkels W, tblBestuurders B
            WHERE W.BestuurderNommer = B.BestuurderNommer
            GROUP BY BestuurderNaam, BestuurderVan';

//=====
// Vraag 2.1.5          3 punte
//=====
    sSQL5 :=
        'UPDATE tblWinkels
        SET WinkelNaam = "TeenDream"
        WHERE WinkelNaam = "Jeans 4U"';
```

```
//=====
// Vraag 2.2 - Delphi-kode afdeling
//=====

//=====
// Vraag 2.2.1          7 punte
//=====

procedure TfrmDBVraag2.btnV2_2_1Click(Sender: TObject);
var
  iTelKlein: integer;
begin // Vraag 2.2.1
  // Sleutel jou kode hier in
  tblWinkels.First;
  iTelKlein := 0;
  while not tblWinkels.Eof do
  begin
    if tblWinkels['WinkelGrootte'] < 300 then
      inc(iTelKlein);
    tblWinkels.Next;
  end;
  redV2_2_1.Lines.Add('Klein winkeltjies: ' + IntToStr(iTelKlein));

  // Kode wat voorsien is
  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;
// =====
// Vraag 2.2.2          13 punte
// =====

procedure TfrmDBVraag2.btnV2_2_2Click(Sender: TObject);
var
  sWinkelNaam: String;
  bGevind: Boolean;
  iBestuurderNom: integer;
begin
  // Kode wat voorsien is
  redV2_2_2.Clear;
  redV2_2_2.Paragraph.TabCount := 2;
  redV2_2_2.Paragraph.Tab[0] := 70;
  redV2_2_2.Paragraph.Tab[1] := 150;

  redV2_2_2.Lines.Add('Naam' + #9 + 'Van' + #9 + 'Kontaknommer');
  sWinkelNaam := inputBox('Soek winkel', 'Sleutel naam van winkel
  in',
  'Little Kitchen Grocery Store');

  // Vraag 2.2.2
  // Sleutel jou kode hier in
  bGevind := false;
  tblWinkels.First;
  while (NOT bGevind) and (NOT tblWinkels.Eof) do
  begin
    if tblWinkels['WinkelNaam'] = sWinkelNaam then
    begin
      bGevind := true;
      iBestuurderNom := tblWinkels['BestuurderNommer']
    end;
    tblWinkels.Next;
  end;
end;
```

```
if bGevind then
begin
tblBestuurders.First;
while NOT tblBestuurders.Eof do
begin
if tblBestuurders['BestuurderNommer'] = iBestuurderNom then
begin

redV2_2_2.Lines.Add(tblBestuurders['BestuurderNaam']
+ #9 + tblBestuurders['BestuurderVan'] + #9 +
tblBestuurders['KontakNommer']);
end;
tblBestuurders.Next;
end;
end
else
begin
ShowMessage('Die winkel is nie in hierdie inkopiesentrum nie');
end;
end;

end.
```



BYLAAG G: OPLOSSING VIR VRAAG 3**Objekklas:**

```

unit GeskenkBewys_U;
interface
uses SysUtils;
type

    TGeskenkBewys = class(TObject)

    private
    var
        fGeskenkBNommer: integer;
        fNaam: String;
        fBalans: real;

    public
        constructor create(iGeskenkBNom: integer; sNaam: String; rBalans:
real);
        function getBalans: real;
        function getGeskenkBNommer: integer;
        function isVoldoende(rBedrag: real):boolean;
        procedure dateerBalansOp(rBedrag: real);
        function toString(): String;
    end;

```

```

implementation

```

```

{ TGeskenkBewys }

```



```

// =====

```

```

// Vraag 3.1.1                    4 punte

```

```

// =====

```

```

constructor TGeskenkBewys.create(iGeskenkBNom: integer; sNaam: String;
    rBalans: real);

```

```

begin
    fGeskenkBNommer := iGeskenkBNom;
    fNaam := sNaam;
    fBalans := rBalans;
end;

```

```

// =====

```

```

// Vraag 3.1.2                    2 punte

```

```

// =====

```

```

function TGeskenkBewys.getGeskenkBNommer: integer;

```

```

begin
    Result := fGeskenkBNommer;
end;

```

```

// =====

```

```

// Vraag 3.1.2                    2 punte

```

```

// =====

```

```

function TGeskenkBewys.getBalans: real;

```

```

begin
    Result := fBalans;
end;

```

```

// =====

```

```

// Vraag 3.1.3                    5 punte

```

```
// =====
function TGeskenkBewys.isVoldoende(rBedrag: real): boolean;
begin
  if rBedrag <= fBalans then
    begin
      Result := True;
    end
  else
    begin
      Result := False;
    end;
  // Alternatief: Result := fBalans >= rBedrag;
end;

// =====
// Vraag 3.1.4          3 punte
// =====
procedure TGeskenkBewys.dateerBalansOp(rBedrag: real);
begin
  fBalans := fBalans - rBedrag;
end;

// =====
// Vraag 3.1.5          4 punte
// =====
function TGeskenkBewys.toString: String;
begin
  Result := 'Geskenkbewysnommer: ' + IntToStr(fGeskenkBNommer)
    + #13 + 'Naam van klant: ' + fName + #13 + 'Beskikbare balans: ' +
    FloatToStrF(fBalans, ffCurrency, 8, 2);
end;

end.
```

Hoofvormeenheid:

```
unit Vraag3_U;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, ExtCtrls, GeskenkBewys_U, ComCtrls, pngimage,
Math;
```

```
type
```

```
TfrmVraag3 = class(TForm)
  pnlV3Opskrif: TPanel;
  grbV3_2_1: TGroupBox;
  grbV3_2_2: TGroupBox;
  Label2: TLabel;
  cmbV3_2_1: TComboBox;
  btnV3_2_1: TButton;
  Label5: TLabel;
  edtV3_2_2: TEdit;
  btnV3_2_2_b: TButton;
  Label6: TLabel;
  edtV3_2_1: TEdit;
```

```

Image1: TImage;
redV3_2_1: TRichEdit;
PageControl1: TPageControl;
tshV3_2_1: TTabSheet;
tshV3_2_2: TTabSheet;
lblV3_2_2: TLabel;
pnlV3_2_2: TPanel;
Label7: TLabel;
btnV3_2_2_a: TButton;
Panel1: TPanel;
Label1: TLabel;
edtV3_GeskenkBNom: TEdit;
procedure btnV3_2_1Click(Sender: TObject);
procedure btnV3_2_2_bClick(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure btnV3_2_2_aClick(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmVraag3: TfrmVraag3;
  objGeskenkBewys: TGeskenkBewys;

```

```
implementation
```

```

{$R *.dfm}
// =====
// Vraag 3.2.1 6 punte
// =====
procedure TfrmVraag3.btnV3_2_1Click(Sender: TObject);
var
  sNaam: String;
  rBedrag: real;
  iGeskenkBNom: integer;
begin
  // Kode wat voorsien is
  redV3_2_1.Clear;
  // Vraag 3.2.1
  iGeskenkBNom := StrToInt(edtV3_GeskenkBNom.Text);
  sNaam := edtV3_2_1.Text;
  rBedrag := StrToFloat(cmbV3_2_1.Text);
  objGeskenkBewys := TGeskenkBewys.create(iGeskenkBNom, sNaam, rBedrag);
  redV3_2_1.Lines.Add(objGeskenkBewys.toString);
end;

```



```
// =====
// Vraag 3.2.2 (a)          5 punte
// =====
procedure TfrmVraag3.btnV3_2_2_aClick(Sender: TObject);
var
  iGBNom: integer;
begin
  // Vraag 3.2.2
  iGBNom := StrToInt(edtV3_GeskenkBNom.Text);
  if iGBNom = objGeskenkBewys.getGeskenkBNommer then
  begin
    pnlV3_2_2.Caption := FloatToStrF(objGeskenkBewys.getBalans,
ffCurrency, 8,
    2);
    btnV3_2_2_b.Enabled := true;
  end;
end;

// =====
// Vraag 3.2.2 (b)          9 punte
// =====
procedure TfrmVraag3.btnV3_2_2_bClick(Sender: TObject);
var
  rAankoopBedrag, rBalansOpKaart, rBedragUitstaande: real;
begin
  // Kode wat voorsien is
  edtV3_2_2.SetFocus;

  // Vraag 3.2.1 - Sleutel jou kode hier in

  rAankoopBedrag := StrToFloat(edtV3_2_2.Text);
  if objGeskenkBewys.isVoldoende(rAankoopBedrag) = true then
  begin
    lblV3_2_2.Caption := ('Geskenkbewys suksesvol gebruik');
    objGeskenkBewys.dateerBalansOp(rAankoopBedrag);
  end
  else
  begin
    rBedragUitstaande := rAankoopBedrag - objGeskenkBewys.getBalans;
    lblV3_2_2.Caption := ('Bedrag deur jou verskuldig om die aankoop te
    voltooi - ' + FloatToStrF(rBedragUitstaande, ffCurrency, 8, 2));
    objGeskenkBewys.dateerBalansOp(objGeskenkBewys.getBalans);
  end;
  rBalansOpKaart := objGeskenkBewys.getBalans;
  pnlV3_2_2.Caption := FloatToStrF(rBalansOpKaart, ffCurrency, 8, 2);
end;

// Kode wat voorsien is
procedure TfrmVraag3.FormCreate(Sender: TObject);
begin
  btnV3_2_2_b.Enabled := false;
  PageContröll.TabIndex := 0;
end;

end.
```

BYLAAG H: OPLOSSING VIR VRAAG 4

```

unit Vraag4_U;

interface

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

type
  TfrmVraag4 = class(TForm)
    btnV4_1: TButton;
    redV4: TRichEdit;
    rgpV4: TRadioGroup;
    btnV4_2: TButton;
    procedure FormActivate(Sender: TObject);
    procedure btnV4_1Click(Sender: TObject);
    procedure btnV4_2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag4: TfrmVraag4;

  arrWinkelTipes: array [1 .. 8] of String = (
    'Elektronika',
    'Jeweliersware',
    'Meubels',
    'Klere',
    'Restaurant',
    'Speelgoed',
    'Wassery',
    'Geskenke'
  );
  arrWinkels: array [1 .. 10] of String;
  iVlak: integer;

implementation
  {$R *.dfm}
  // =====
  // Vraag 4.1          15 punte
  // =====
  procedure TfrmVraag4.btnV4_1Click(Sender: TObject);
  var
    iKol, iRy, iAantal: integer;
    tLeer: TextFile;
    sReel: String;
  begin
    redV4.Clear;
    redV4.Lines.Add('Vlak: ' + #9 + ' Winkels:' + #9 +
      'Getal winkels per vlak:');
    redV4.Lines.Add('=====');
    AssignFile(tLeer, rgpV4.Items[rgpV4.ItemIndex] + '.txt');
  end;

```

```

Reset(tLeer);
iVlak := 0;

while not eof(tLeer) do
begin
  inc(iVlak);
  Readln(tLeer, sReel);
  arrWinkels[iVlak] := sReel;
  sReel := intToStr(iVlak) + '.' + sReel;

  iAantal := length(sReel);
  while length(sReel) < 25 do
    insert(' ', sReel, 3);

  redV4.Lines.Add(sReel + #9 + intToStr(iAantal));
end;
end;
// =====
// Vraag 4.2          15 punte
// =====
procedure TfrmVraag4.btnV4_2Click(Sender: TObject);
const
  Winkels = 'EJMKRSWG';
var
  iKol, iRy, iTotaal, iIndeks: integer;
  sWinkelTipe, sWinkel, sResultate: String;

begin
  // Vraag 4.2
  redV4.Lines.Add(' ');
  sWinkelTipe := uppercase(InputBox('Sleutel tipe winkel in, bv. M',
    'Geldige tipe winkels: E G J K M R S W', 'M'));
  iIndeks := pos(sWinkelTipe, Winkels);
  if iIndeks = 0 then
    Showmessage('Die geldige tipe winkels is : '+#13 +'E G J K M R S W')
  else
    begin
      redV4.Lines.Add('Tipe winkel: ' + arrWinkelTipes[iIndeks] + #13 +
        'Getal per vlak ');

      for iRy := 1 to iVlak do
        begin
          iTotaal := 0;
          sResultate := intToStr(iRy) + '. ';
          for iKol := 1 to length(arrWinkels[iRy]) do
            begin
              sWinkel := arrWinkels[iRy][iKol];
              if (sWinkel = sWinkelTipe) then
                inc(iTotaal);
            end;
          sResultate := sResultate + #9 + intToStr(iTotaal);
          redV4.Lines.Add(sResultate); // ry-totaal
        end; // gaan na volgende ry
      end; // else
    end;
end;

```



```
// Kode wat voorsien is - moenie verander nie
procedure TfrmVraag4.FormActivate(Sender: TObject);
begin
  redV4.Paragraph.TabCount := 10;
  redV4.Paragraph.Tab[0] := 20;
  redV4.Paragraph.Tab[1] := 70;
  redV4.Paragraph.Tab[2] := 130;
  redV4.Paragraph.Tab[3] := 200;
  redV4.Paragraph.Tab[4] := 260;
  redV4.Paragraph.Tab[5] := 320;
  redV4.Paragraph.Tab[6] := 370;
  redV4.Paragraph.Tab[7] := 430;
  redV4.Paragraph.Tab[8] := 500;
  redV4.Paragraph.Tab[9] := 570;
end;

end.
```

