



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEKNOLOGIE V1

NOVEMBER 2025

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 27 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 van 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 leerders 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.
- **Bylae A, B, C en D** (bladsy 3 tot 12) sluit die nasienriglyne vir elke vraag in.
- **Bylae E, F, G en H** (bladsy 13 tot 27) bevat voorbeelde van oplossings vir Vrae 1 tot 4 in programmeringskode.
- Kopieë van **Bylae A, B, C, D en die opsomming van die leerder se punte** (bladsy 3 tot 12) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

BYLAE A**VRAAG 1: NASIENRUBRIEK – ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1	Knoppie [1.1 - Increase value] Onttrek die getal uit pnlQ1_1 ✓ Skakel om na integer ✓ Vermeerder die getal met 1 ✓ Vertoon die getal in pnlQ1_1 omgeskakel na string ✓	4	
1.2	Knoppie [1.2 - Volume] Onttrek die lengte van die een sy uit spnQ1_2 ✓ Bereken volume: $rVolume := 5 * \sqrt{(3 + \sqrt{5})} / 12 * \text{Power}(iSide, 3) \checkmark$ Vertoon volume in edtQ1_2 omgeskakel na 'n string en geformateer tot 2 desimale plekke ✓ LET WEL: Enige TWEE wiskundige funksies kan gebruik word.	6	
1.3	Knoppie [1.3 - Shopping aisle] Onttrek die kode uit edtQ1_3 ✓ Vind die posisie van die (#) (of ander meganisme om die gangnommer en die kategorie te skei) ✓ Onttrek die gangnommer ('aisle number') ✓ Onttrek/gebruik die kategorie as karakter ✓ Maak korrekte gebruik van 'n Case ✓ Vir alle moontlike karakters ✓ ('F', 'D', 'B', 'S', 'P') ✓ om kategoriebeskrywing te bepaal ✓ Stel 'n string met "Aisle", gangnommer, ✓ ":" en kategoriebeskrywing ✓ Vertoon die string in lblQ1_3 ✓	11	

1.4	Knoppie [1.4 - Populate and display] Lus ('loop') vanaf 1 tot lengte van skikking ✓ Genereer ewekansige getalle ✓ in korrekte reeks (1 – 15) ✓ Ken ewekansige getal aan elke element in die skikking toe ✓ Voeg die waardes in skikkings saam om string te bou, ✓ met waardes wat deur 'n streep (-) geskei is ✓ Meganisme om ekstra streep (-) te hanteer ✓ Vertoon saamgevoegde string in memQ1_4 ✓	8	
1.5	Knoppie [1.5 - Determine HCF] Toets IF Num1 < Num2 ✓ Ken Num1 toe aan LowerNum ✓ Anders Ken Num2 toe aan LowerNum ✓ <i>//Maak gebruik van While lus</i> Cnt = 1; ✓ Lus ('Loop') solank Cnt <= LowerNum ✓ Toets IF ✓ Num1 MOD Cnt = 0 ✓ Toets IF Num2 MOD Cnt = 0 ✓ HCF = Cnt ✓ Inc(Cnt) ✓ Vertoon HCF in edtQ1_5 ✓ <i>Alternatief deur FOR... loop te gebruik</i> Lus ('Loop') Cnt (1) van 1 tot (1) LowerNum (1) Toets IF (1) Num1 MOD Cnt = 0 (1) Toets IF Num2 MOD Cnt = 0 (1) HCF = Cnt (1) Vertoon HCF in edtQ1_5 (1)	11	
	TOTAAL AFDELING A:	40	

BYLAE B**VRAAG 2: NASIENRUBRIEK – DATABASISPROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	SQL-stellings		
2.1.1	Knoppie [2.1.1 - Farm details] SELECT FarmName, NearestTown, SizeInHectares ✓ FROM tblFarms ✓ ORDER BY SizeInHectares DESC ✓	3	
2.1.2	Knoppie [2.1.2 - Contact details] SELECT FullName, ContactNumber FROM tblFarmOwners ✓ WHERE Email IS NULL ✓ Alternatief: ISNULL(Email)		
2.1.3	Knoppie [2.1.3 - Average farm size] SELECT FarmType, FORMAT(AVG(SizeInHectares) ✓, "0.00" ✓) AS AverageSize ✓ FROM tblFarms WHERE FarmType = ✓ ' ' + sFarmType + ' ' ✓ GROUP BY FarmType ✓ Alternatief: FarmType = ' + QuotedStr(sFarmType) + ' FarmType LIKE ' ' + sFarmType + ' '	6	
2.1.4	Knoppie [2.1.4 - Young farm owners] SELECT Fullname, DateOfBirth, ✓ INT ✓ ((NOW ✓ - DateOfBirth) / 365 ✓) AS Age FROM tblFarmOwners ✓ WHERE INT((NOW - DateOfBirth) / 365) ✓ <= 25 ✓ Alternatief: (Date() - DateOfBirth)		
		7	

2.1.5	Knoppie [2.1.5 - Multiple farms in KZN]	7	
	<pre> SELECT FullName, COUNT(*) ✓ AS TotalFarms FROM tblFarmOwners, tblFarms ✓ WHERE tblFarmOwners.OwnerID = tblFarms.OwnerID ✓ AND ✓ (NearestTown = "Durban" OR NearestTown = "Pietermaritzburg") ✓ GROUP BY FullName ✓ HAVING COUNT(*) > 1 ✓ Alternatief: NearestTown IN ("Durban", "Pietermaritzburg) </pre>		
	Subtotaal:	25	

VRAAG 2: NASIENRUBRIEK (VERVOLG)

2.2	Databasismanipulasie		
	Knoppie [2.2 – Mixed-farm type] Toets IF file exists = False/try..except ✓ Vertoon 'n geskikte boodskap ✓ Exit of else-stelling ✓ AssignFile(tFile, 'MiixedFarms.txt') ✓ Reset(tFile) ✓ Lus ('Loop') deur tekslêer ✓ Readln(tFile, FarmID veranderlike) ✓ tblFarms.First ✓ Lus ('Loop') deur tblFarms ✓ Toets IF FarmID veranderlike = tblFarms['FarmID'] ✓ Toets IF tblFarms['SizeInHectares'] > 100 ✓ lblFarms.Edit ✓ tblFarms['FarmType'] = 'Mixed' ✓ tblFarms.Post Anders Vertoon tblFarms['FarmID' as string, tblFarms['FarmName'] en tblFarms['SizeInHectares'] as string in redQ2_2 ✓ tblFarms.Next ✓ Eindig lus (tblFarms) Eindig lus (Tekslêer)	15	
	Subtotaal:	15	
	TOTAAL AFDELING B:	40	

BYLAE C**VRAAG 3: NASIENRUBRIEK – OBJEK-GEÖRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	Konstruktor metode Konstruktor opskrif met String, Integer en Boolean parameters ✓ Ken die drie parameters toe aan die korrekte attribute ✓ Ken die volgende verstekwaardes toe aan die ander attribute: fNoOfHarvests = 2 ✓ fTotalHoneyHarvested = 80 ✓ fHarvestDates = '2025/01/05' + #13 + '2025/06/23' ✓	4	
3.1.2	Funksie getNoOfHarvests Funksie opskrif getNoOfHarvests met integer as terugstuur-tipe ✓ Stuur fNoOfHarvests terug ✓	2	
3.1.3	Funksie calcAverage Funksie opskrif calcAverage ✓ met real terugstuurtype ✓ Stuur terug ✓ fTotalHoneyHarvested/ fNoOfHarvests ✓ AANVAAR OOK: fTotalHoneyHarvested/ getNoOfHarvests	4	
3.1.4	Prosedure updateBeehiveDetails Prosedure opskrif met parameter van tipe real ✓ fTotalHoneyHarvested := fTotalHoneyHarvested ✓ + parameter ✓ fNoOfHarvests := fNoOfHarvests + 1 fHarvestDates := fHarvestDates ✓ + #13 + system date ✓	6	

3.1.5	Funksie checkHealthStatus Funksie opskrif met Boolean terugstuurtype ✓ Toets IF (fBeeCount > 7000) ✓ OR (fNoOfHarvests <= 3) ✓ AND (fPests = False) ✓ Result = True ✓ Anders Result = False ✓ <i>Alternatief 1:</i> Funksie opskrif met Boolean terugstuurtype (1) Result = False (1) Toets IF (fBeeCount > 7000) (1) OR (fNoOfHarvests <= 3) (1) Toets IF (fPests = False) (1) Result = True (1) <i>Alternatief 2:</i> Funksie opskrif met Boolean terugstuurtype (1) Result = (2) ((fBeeCount > 7000) (1) OR (fNoOfHarvests <=3)) (1) AND (fPests = FALSE) (1)	6	
	Subtotaal: Objekklas	22	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	Knoppie [3.2.1 - Instantiate beehive object] objBeehive := ✓ TBeehive.Create() ✓ Onttrekte parameters in volgorde ✓ Vertoon objek in redQ3_2 ✓ met toString metode ✓	5	
3.2.2	Knoppie [3.2.2 - Ready to harvest?] Toets IF objBeehive.checkHealthStatus = TRUE ✓ Vertoon 'Ready to harvest' op lblQ3_2_2 btnQ3_2_3.Enabled = True Else ✓ Vertoon 'Not ready to harvest' op lblQ3_2_2 ✓ //beide btnQ3_2_3.Enabled = False ✓ //beide Konsepte: Toets IF voorwaarde (1) met Else-stelling (1) Vertoon boodskap 'Ready to harvest' / 'Not ready to Harvest' op lblQ3_2_2 (1) btnQ3_2_3.Enabled = True/False (1)	4	
3.2.3	Knoppie [3.2.3 - Honey harvested] Onttrek Waarde vanaf edtQ3_2_3 as real datatipe ✓ Roep objBeehive.updateBeehiveDetails (Waarde) ✓ Vertoon in redQ3: Byskrif + objBeehive.getNoOfHarvests as string ✓ objBeehive.toString ✓	4	
3.2.4	Knoppie [3.2.4 - Average honey harvested] Roep objBeehive.calcAverage ✓ om gemiddeld te vertoon deur ShowMessage te gebruik ✓ geformateer tot 2 desimale plekke ✓	3	
3.2.5	Knoppie [3.2.5 - Change status of pests] Roep die objBeehive.setPests ✓ Deaktiveer btnQ3_2_3 ✓	2	
	Subtotaal Vormklas:	18	
	TOTAAL AFDELING C:	40	

BYLAE D**VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
4.1.1	Knoppie - [4.1.1 - Extract] Lus ('Loop') Ry van 1 tot 10 ✓ Lus ('Loop')Kol van 1 tot 2 ✓ Vind posisie van die plekhouer ✓ in arrSoil[Ry] ✓ Onttrek waarde as getal ✓ en stoor in arr2DSoil[Ry,Kol] ✓ Verwyder ('Delete') van indeks 1 tot plekhouer ✓ Meganisme om laaste waarde te hanteer ✓	8	
4.1.2	Knoppie - [4.1.2 - Validate] Lus ('Loop') Ry van 1 tot 10 ✓ Inisialiseer Totaal na 0 ✓ Lus ('Loop') Kol van 1 tot 3 ✓ Vermeerder Totaal met waarde van arr2DSoil[Ry,Kol] ✓ Toets IF Totaal < 100 ✓ Lus ('Loop') A van 1 tot 3 ✓ Bereken verhouding arr2DSoil[Ry, A] ✓ / Totaal * 100 ✓ Vertoon hektaar en Ry-nommer as string in redQ4_1 ✓	9	

4.2	Knoppie - [4.2 - Nutrient markers] Lus Loop1 van 1 tot 50000 ✓ Inisialiseer Som na 0 ✓ Lus ✓ Loop2 van 1 tot Lengte(Loop1) ✓ //aantal syfers Inisialiseer Faktoriaal na 1 ✓ Lus ✓ Loop3 van 1 tot skakel na integer-tipe om (skakel na String-tipe om(Loop1)✓ [Loop2]) ✓ //waarde van syfer Faktoriaal := Faktoriaal * Loop3 ✓ Som := Som + Faktoriaal ✓ Toets ✓ IF Som = Loop1 ✓ Vertoon Loop1/Som as string in memQ4_2 ✓ Konsepte: Buitenste Loop1 (1) Inisialiseer Som-veranderlike (1) Binneste Loop2 (1) – Geneste loop2 deur lengte van die string van buitenste lus se veranderlike te gebruik (1) Inisialiseer Faktoriaal-veranderlike (1) Onttrek van syfer – Geneste Loop3(1), Syfer-ontrekking (1) Omgeskakel na heelgetal (1) Bereken faktoriaal (1) Logika van som (1) Vergelyking logika – IF stelling (1) Som = Buitenste Loop1(1) Afvoer (1)	13	
	TOTAAL AFDELING D: GROOTTOTAAL:	30 150	

OPSOMMING VAN LEERDER SE PUNTE:

SENTRUMNOMMER:		LEERDER SE EKSAMENNUMMER:			
	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					

BYLAE E: OPLOSSING VIR VRAAG 1

```
unit Question1_U;

interface

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

type
  gpbQ1_1: TGroupBox;
  pnlQ1_1: TPanel;
  btnQ1_1: TButton;
  gpbQ1_2: TGroupBox;
  lblLength: TLabel;
  spnQ1_2: TSpinEdit;
  imgIcosahedron: TImage;
  edtQ1_2: TEdit;
  btnQ1_2: TButton;
  gpbQ1_3: TGroupBox;
  lblDate: TLabel;
  edtQ1_3: TEdit;
  lblQ1_3: TLabel;
  gpbQ1_4: TGroupBox;
  memQ1_4: TMemo;
  btnQ1_4: TButton;
  lblVolume: TLabel;
  btnQ1_3: TButton;
  GroupBox1: TGroupBox;
  edtQ1_5_Num1: TEdit;
  Label1: TLabel;
  Label2: TLabel;
  edtQ1_5_Num2: TEdit;
  btnQ1_5: TButton;
  edtQ1_5: TEdit;
  procedure btnQ1_1Click(Sender: TObject);
  procedure btnQ1_2Click(Sender: TObject);
  procedure btnQ1_3Click(Sender: TObject);
  procedure btnQ1_4Click(Sender: TObject);
  procedure btnQ1_5Click(Sender: TObject);
private
  { Private declarations }

public
  { Public declarations }

end;

var
  frmQuestion1: TfrmQuestion1;

implementation
{$R *.dfm}
```

```
// =====
// Vraag 1.1           4 punte
// =====
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
var
    iNum: Integer;
begin
    // Question 1.1
    iNum := StrToInt(pnlQ1_1.Caption);
    Inc(iNum);
    pnlQ1_1.Caption := IntToStr(iNum);
end;

// =====
// Vraag 1.2           6 punte
// =====
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
    iSide: Integer;
    rVolume: Real;
begin
    // Question 1.2
    iSide := spnQ1_2.Value;
    rVolume := 5 * (3 + Sqrt(5)) / 12 * Power(iSide,3);
    edtQ1_2.Text := FloatToStrF(rVolume, ffFixed, 8, 2);
end;

// =====
// Vraag 1.3           11 punte
// =====
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
    sCode, sOutput: string;
    sAisle: string;
    cCategory : Char;
    iPos : Integer;
begin
    // Question 1.3
    sCode := edtQ1_3.Text;
    iPos := Pos('#', sCode);
    cCategory := sCode[iPos+1];
    sAisle := Copy(sCode, 1, iPos-1);
    case cCategory of
        'F': sOutput := 'Fruit and vegetables';
        'D': sOutput := 'Dairy';
        'B': sOutput := 'Butchery';
        'S': sOutput := 'Sauces';
        'P': sOutput := 'Pasta and Rice';
    end;
    sOutput := 'Aisle ' + sAisle + ': ' + sOutput;
    lblQ1_3.Caption := sOutput;
end;
```

```
// =====  
// Vraag 1.4      8 punte  
// =====  
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);  
// Provided code  
var  
    arrNumbers: array [1 .. 9] of Integer;  
// End of provided code  
  
    iCnt: Integer;  
    sOut: String;  
begin  
    // Question 1.4  
  
    sOut := '';  
    for iCnt := 1 to length(arrNumbers) do  
    begin  
        arrNumbers[iCnt] := RandomRange(1, 16);  
        sOut := sOut + '-' + IntToStr(arrNumbers[iCnt]);  
    end;  
    Delete(sOut, length(sOut), 1);  
    memQ1_4.Text := sOut;  
end;  
  
// =====  
// Vraag 1.5      11 punte  
// =====  
procedure TfrmQuestion1.btnQ1_5Click(Sender: TObject);  
var  
    iCnt: Integer;  
    iHCF, iLowerNum, iNum1, iNum2: Integer;  
begin  
    // Provided code  
    iNum1 := StrToInt(edtQ1_5_Num1.Text);  
    iNum2 := StrToInt(edtQ1_5_Num2.Text);  
    iHCF := 0;  
  
    // Question 1.5  
    if iNum1 < iNum2 then  
        iLowerNum := iNum1  
    else  
        iLowerNum := iNum2;  
  
    // Code using a while loop  
    iCnt := 1;  
    While iCnt <= iLowerNum do  
    begin  
        if (iNum1 MOD iCnt = 0) then  
            if (iNum2 MOD iCnt = 0) then  
                iHCF := iCnt;  
            Inc(iCnt);  
        end;  
    end;
```

Code using a Repeat Until loop

```
{ iCnt := 1;
  Repeat
    if (iNum1 MOD iCnt = 0) then
      if (iNum2 MOD iCnt = 0) then
        iHCF := iCnt;
      Inc(iCnt);
  Until iCnt > iLowerNum;
end;}
```

//Code using a For loop

```
{for iCnt := 1 to iLowerNum do
begin
  if (iNum1 MOD iCnt = 0) then
    if (iNum2 MOD iCnt = 0) then
      begin
        iHCF := iCnt;
      end;
end;}
```

```
edtQ1_5.Text := 'HCF: ' + IntToStr(iHCF);
```

```
end;
```

```
end.
```

BYLAE F: OPLOSSING VIR VRAAG 2

```
//=====
// Vraag 2.1 - Afdeling: SQL statements
//=====

//=====
// Vraag 2.1.1          3 punte
//=====
procedure TfrmQuestion2.btnQ2_1_1Click(Sender: TObject);
var
    sSQL1: string;
begin
    // Question 2.1.1

    sSQL1 := 'SELECT FarmName, NearestTown, SizeInHectares FROM tblFarms ' +
        'ORDER BY SizeInHectares DESC';

    // Provided code - do not change
    dbCONN.runSQL(sSQL1);
end;

//=====
// Vraag 2.1.2          2 punte
//=====
procedure TfrmQuestion2.btnQ2_1_2Click(Sender: TObject);
var
    sSQL2: string;
begin
    // Question 2.1.2

    sSQL2 := 'SELECT FullName, ContactNumber ' +
        'FROM tblFarmOwners WHERE Email IS NULL';

    // Provided code - do not change
    dbCONN.runSQL(sSQL2);
end;

//=====
// Vraag 2.1.3          6 punte
//=====
procedure TfrmQuestion2.btnQ2_1_3Click(Sender: TObject);
var
    sSQL3, sFarmType: string;
begin
    // Provided code
    sFarmType := cmbQ2_1_3.Text;

    // Question 2.1.3
    sSQL3 :=
        'SELECT FarmType, FORMAT(AVG(SizeInHectares),"0.00") as AverageSize ' +
        'FROM tblFarms ' + 'WHERE FarmType = "' + sFarmType + '" GROUP BY
        FarmType';
    // Provided code - do not change
    dbCONN.runSQL(sSQL3);
end;
```

```
//=====
//  Vraag 2.1.4           7 punte
//=====
procedure TfrmQuestion2.btnQ2_1_4Click(Sender: TObject);
var
    sSQL4: string;
begin
    //Question 2.1.4

    sSQL4 := 'SELECT FullName, DateOfBirth, INT((NOW - DateOfBirth)/365)
              AS Age ' +
              'FROM tblFarmOwners ' +
              'WHERE INT((NOW - DateOfBirth)/365) <= 25 ';

    // Alternatief
    sSQL4:= 'SELECT FullName, INT(NOW - DateOfBirth)/365) AS Age
            ' +
            ' FROM tblFarmOwners' +
            ' WHERE INT((NOW- DateOfBirth)/365) <= 25 ';

    // Provided code - do not change
    dbCONN.runSQL(sSQL4)

end;

//=====
//  Vraag 2.1.5           7 punte
//=====
procedure TfrmQuestion2.btnQ2_1_5Click(Sender: TObject);
var
    sSQL5: string;
begin
    // Question 2.1.5
    sSQL5 := 'SELECT  FullName, COUNT(*) AS TotalFarms ' +
              'FROM tblFarmOwners, tblFarms ' +
              'WHERE tblFarmOwners.OwnerID = tblFarms.OwnerID AND
              (NearestTown = "Durban" OR NearestTown = "Pietermaritzburg")' +
              'GROUP BY FullName ' +
              'HAVING COUNT(*) > 1 ';

    // Provided code - do not change
    dbCONN.runSQL(sSQL5);
```

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

//=====
// Vraag 2.2           15 punte
// =====

procedure TfrmQuestion2.btnQ2_2Click(Sender: TObject);
var
    tFile: TextFile;
    iFarmID: Integer;
    sFarmType: string;
begin
    // Provided code - do not change
    redQ2_2.Clear;
    redQ2_2.Lines.Add('Farms on list that do not qualify: ');

    // Question 2.2
    Assignfile(tFile, 'MixedFarms.txt');
    try
        Reset(tFile);
    except
        ShowMessage('File does not exist');
        Exit;
    end;

    while not Eof(tFile) do
    begin
        Readln(tFile, iFarmID);
        tblFarms.First;
        while not tblFarms.Eof do
        begin
            if (tblFarms['FarmID'] = iFarmID) then
            begin
                if tblFarms['SizeInHectares'] > 100 then
                begin
                    tblFarms.Edit;
                    tblFarms['FarmType'] := 'Mixed';
                    tblFarms.Post;
                end
            else
                redQ2_2.Lines.Add(IntToStr(tblFarms['FarmID']) + ' - ' + tblFarms
                    ['FarmName'] + ' - ' + IntToStr(tblFarms['SizeInHectares']) +
                    ' hectares');
            end;
            tblFarms.Next;
        end;
    end;

    // Provided code
    tblFarms.Sort := 'FarmID ASC';
end;
```

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

```

unit Beehive_U;

interface

Uses SysUtils, DateUtils, Math;

Type
  TBeehive = class(TObject)
  Private
    fBeehiveID: String;
    fBeeCount: Integer;
    fPests: Boolean;
    fNoOfHarvests: Integer;
    fTotalHoneyHarvested: Real;
    fHarvestDates: String;

  Public
    Constructor Create(sBeehiveID: String; iBeeCount: Integer;
                      bPests : boolean);
    Function getNoOfHarvests: Integer;
    Function calcAverage: Real;
    Procedure updateBeehiveDetails(rHoneyKG: Real);
    Function checkHealthStatus: Boolean;

    // Provided code
    Procedure setPests;
    Function toString: String;
  End;
implementation

{ TBeehive }
// =====
// Vraag 3.1.1          4 punte
// =====
Constructor TBeehive.Create(sBeehiveID: String; iBeeCount: Integer; bPests
: boolean);
begin
  fBeehiveID := sBeehiveID;
  fBeeCount := iBeeCount;
  fPests := bPests;
  fNoOfHarvests := 2;
  fTotalHoneyHarvested := 80;
  fHarvestDates := '2025/01/05' + #13 + '2025/06/23';
end;
// =====
// Vraag 3.1.2          2 punte
// =====
function TBeehive.getNoOfHarvests: Integer;
begin
  Result := fNoOfHarvests;
end;

```

```
// =====
// Vraag 3.1.3           4 punte
// =====
function TBeehive.calcAverage: Real;
begin
    Result := fTotalHoneyHarvested / fNoOfHarvests;
end;
// =====
// Vraag 3.1.4           6 punte
// =====
procedure TBeehive.updateBeehiveDetails(rHoneyKG: Real);
begin
    fTotalHoneyHarvested := fTotalHoneyHarvested + rHoneyKG;
    Inc(fNoOfHarvests);
    fHarvestDates := fHarvestDates + #13 + DateToStr(Date);
end;
// =====
// Vraag 3.1.5           6 punte
// =====
function TBeehive.checkHealthStatus: Boolean;
var
    bHealthy : Boolean;
begin
    bHealthy := False;
    if (fPests = False) AND ((fBeeCount > 7000) OR
                             (fNoOfHarvests <= 3)) then
        bHealthy := True;
    Result := bHealthy;
//Alternative
// Result := (fPests = False) AND ((fBeeCount > 7000) OR
//                                   (fNoOfHarvests <= 3))

end;
// =====
// Metodes wat voorsien is
// =====
procedure TBeehive.setPest;
begin
    if fPests = True then
        fPests := False
    else
        fPests := True;
end;
function TBeehive.toString: String;
var
    sPestStatus: String;
begin
    if fPestStatus then
        sPestStatus := 'Pests in beehive.'
    else
        sPestStatus := 'No pests in beehive.';
    Result := 'Beehive ID: ' + #9 + fBeehiveID + #13 +
        'Number of bees in beehive: ' + #9 + IntToStr(fBeeCount) + #13 +
        'Honey harvested in kg: ' + #9 +
        FloatToStrF(fTotalHoneyHarvested, ffFixed, 8, 1) + #13 +
        'Number of harvests: ' + #9 + IntToStr(fNoOfHarvests) + #13 + #13 +
        'Harvest dates: ' + #13 + fHarvestDates + #13 + sPestStatus;
end;
end.
```

Hoofvormeenheid:

```

unit Question3_U;
interface
uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
    Forms, Dialogs, StdCtrls, pngimage, ExtCtrls, Spin, ComCtrls, Math,
    beehive_u;

type
    TfrmQuestion3_2 = class(TForm)
        pnlQ3_2: TPanel;
        imgBeeLeft: TImage;
        gpbQ3_2_1: TGroupBox;
        cmbQ3: TComboBox;
        lblCode: TLabel;
        btnQ3_2_1: TButton;
        lblNumBees: TLabel;
        gpbQ3_2_3: TGroupBox;
        lblHavest: TLabel;
        imgBeeRight: TImage;
        gpbQ3_2_4: TGroupBox;
        btnQ3_2_4: TButton;
        redQ3_2: TRichEdit;
        btnQ3_2_3: TButton;
        spnQ3: TSpinEdit;
        edtQ3_2_3: TEdit;
        btnQ3_2_2: TButton;
        gpbQ3_2_2: TGroupBox;
        GroupBox1: TGroupBox;
        lblQ3_2_2: TLabel;
        GroupBox2: TGroupBox;
        GroupBox3: TGroupBox;
        btnQ3_2_5: TButton;
        Edit1: TEdit;
        chbQ3: TCheckBox;
        lblPests: TLabel;
        procedure btnQ3_2_1Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure btnQ3_2_3Click(Sender: TObject);
        procedure btnQ3_2_4Click(Sender: TObject);
        procedure btnQ3_2_2Click(Sender: TObject);
        procedure btnQ3_2_5Click(Sender: TObject);
        procedure chbQ3Click(Sender: TObject);
        procedure GroupBox2Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;
var
    frmQuestion3_2: TfrmQuestion3_2;
    // Provided code
    objBeehive: TBeehive;

```

implementation

```
// =====  
// Vraag 3.2.1          5 punte  
// =====  
procedure TfrmQuestion3_2.btnQ3_2_1Click(Sender: TObject);  
var  
    // Provided code  
    sBeehiveID: string;  
    iNumBees: Integer;  
    bPests: Boolean;  
begin  
    // Provided code  
    redQ3_2.Clear;  
    sBeehiveID := cmbQ3.Text;  
    iNumBees := spnQ3.Value;  
    bPests := chbQ3.checked;  
  
    // Question 3.2.1  
    objBeehive := TBeehive.Create(sBeehiveID, iNumBees, bPests);  
    redQ3_2.Lines.Add(objBeehive.toString);  
end;;  
  
// =====  
// Vraag 3.2.2          4 punte  
// =====  
procedure TfrmQuestion3_2.btnQ3_2_2Click(Sender: TObject);  
begin  
    // Question 3.2.2  
    if objBeehive.checkHealthStatus then  
        lblQ3_2_2.Caption := 'Beehive is ready to harvest.'  
    else  
        lblQ3_2_2.Caption := 'Beehive is not ready to harvest.';  
    btnQ3_2_3.Enabled := objBeehive.checkHealthStatus;  
end;  
  
// =====  
// Vraag 3.2.3          4 punte  
// =====  
procedure TfrmQuestion3_2.btnQ3_2_3Click(Sender: TObject);  
var  
    rAmount: Real;  
begin  
    // Provided code  
    redQ3_2.Clear;  
  
    // Question 3.2.3  
    rAmount := StrToFloat(edtQ3_2_3.Text);  
    objBeehive.updateBeehiveDetails(rAmount);  
    redQ3_2.Lines.Add('Harvest number:' +  
        IntToStr(objBeehive.getNoOfHarvests));  
    redQ3_2.Lines.Add(objBeehive.toString);  
end;
```

```
// =====  
// Vraag 3.2.4          3 punte  
// =====  
procedure TfrmQuestion3_2.btnQ3_2_4Click(Sender: TObject);  
begin  
    // Question 3.2.4  
    ShowMessage('Average honey harvested = ' + FloatToStrF  
        (objBeehive.calcAverage, ffFixed, 6, 2));  
end;  
  
// =====  
// Vraag 3.2.5          2 punte  
// =====  
procedure TfrmQuestion3_2.btnQ3_2_5Click(Sender: TObject);  
begin  
    objBeehive.setPests;  
    btnQ3_2_3.Enabled := false;  
end;  
  
// =====  
// Kode wat voorsien is  
// =====  
  
{ $REGION 'Provided code - DO NOT MODIFY' }  
  
procedure TfrmQuestion3_2.FormCreate(Sender: TObject);  
begin  
    redQ3_2.clear;  
    redQ3_2.Paragraph.tabcount := 1;  
    redQ3_2.Paragraph.tab[0] := 150;  
  
    btnQ3_2_3.Enabled := False;  
end;  
{ $ENDREGION }  
  
end.
```

BYLAE H: OPLOSSING VIR VRAAG 4

```
unit Question4_U;

interface

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

type
  TfrmQuestion4 = class(TForm)
    gbxQ4_1: TGroupBox;
    gbxQ4_2: TGroupBox;
    btnQ4_1_1: TButton;
    btnQ4_1_2: TButton;
    btnQ4_2: TButton;
    memQ4_2: TMemo;
    redQ4_1: TRichEdit;
    procedure btnQ4_1_1Click(Sender: TObject);
    procedure btnQ4_1_2Click(Sender: TObject);
    procedure btnQ4_2Click(Sender: TObject);

  private
    { Private declarations }
  public
    procedure Display;
  end;

var
  frmQuestion4: TfrmQuestion4;
  // Provided code
  arrSoil: array [1 .. 10] of String = (
    '20:50:10',
    '40:30:30',
    '30:28:30',
    '60:20:20',
    '25:30:45',
    '50:40:10',
    '30:55:15',
    '45:35:20',
    '35:0:55',
    '55:25:20'
  );
  arr2DSoil: array [1 .. 10, 1 .. 3] of Real;

implementation

{$R *.dfm}
```

```
// =====
// Vraag 4.1.1           8 punte
// =====
procedure TfrmQuestion4.btnQ4_1_1Click(Sender: TObject);
var
    iRow, iCol, iPos: Integer;
    sTemp : String;
begin
    // Question 4.1.1

    for iRow := 1 to 10 do
    begin
        for iCol := 1 to 2 do
        begin
            iPos := pos(':', arrSoil[iRow]);
            arr2DSoil[iRow, iCol] := StrToInt(copy(arrSoil[iRow], 1, iPos - 1));
            delete(arrSoil[iRow], 1, iPos);
        end;
        arr2DSoil[iRow, iCol] := StrToInt(arrSoil[iRow]);
    end;

    // Provided code
    redQ4_1.Clear;
    Display;
end;

// =====
// Vraag 4.1.2           9 punte
// =====
procedure TfrmQuestion4.btnQ4_1_2Click(Sender: TObject);
var
    iRow, iCol, A: Integer;
    rClay, rSand, rSilt, rSum: Real;
begin
    // provided code
    redQ4_1.Clear;
    redQ4_1.lines.add('Hectares adjusted:');
    // Question 4.1.2
    for iRow := 1 to 10 do
    begin
        rSum := 0;
        for iCol := 1 to 3 do
            rSum := rSum + arr2DSoil[iRow, iCol];

        if (rSum < 100) then
        begin
            for A := 1 to 3 do
                arr2DSoil[iRow, A] := (arr2DSoil[iRow, A] / rSum * 100);
            redQ4_1.Lines.Add('Hectare ' + IntToStr(iRow));
        end;
    end;

    // Provided code
    redQ4_1.Lines.Add(#13 + 'Updated data: ');
    redQ4_1.Lines.Add('=====');
    Display;
end;
```

```
// =====
// Vraag 4.2           13 punte
// =====
procedure TfrmQuestion4.btnQ4_2Click(Sender: TObject);
var
  rTFactor, rTNum: Real;
  iValue: Integer;
  iLoop1, iLoop2, iLoop3: Integer;
begin
  // Provided code
  memQ4_2.Clear;
  memQ4_2.Lines.Add('Nutrient markers:');
  memQ4_2.Lines.Add('=====');

  // Question 4.2
  For iLoop1 := 1 to 50000 do
  begin
    rTNum := 0;
    For iLoop2 := 1 to Length(IntToStr(iLoop1)) do
    begin
      rTFactor := 1;
      For iLoop3 := 1 to StrToInt(IntToStr(iLoop1)[iLoop2]) do
        rTFactor := rTFactor * iLoop3;
        rTNum := rTNum + rTFactor;
      end;
      if rTNum = iLoop1 then
        memQ4_2.Lines.Add(IntToStr(iLoop1));
      end;
    end;
  end;

//=====
// Display Method
// =====
// Provided code
procedure TfrmQuestion4.Display;
var
  iRow, iCol: Integer;
  sTotal: String;
begin
  redQ4_1.Lines.Add('Hectare' + #9 + 'Clay' + #9 + 'Sand' + #9 +
'Silt');
  For iRow := 1 to 10 do
  begin
    sTotal := IntToStr(iRow) + #9#9;
    For iCol := 1 to 2 do
      sTotal := sTotal + FloatToStrF(arr2DSoil[iRow, iCol],
                                     ffFixed, 8, 1) + #9;
    sTotal := sTotal + FloatToStrF(arr2DSoil[iRow, 3], ffFixed, 8, 1);
    redQ4_1.Lines.Add(sTotal);
  end;
end;

end.
```