basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIORSERTIFIKAAT-EKSAMEN/
NASIONALE SENIORSERTIFIKAAT-EKSAMEN

INLIGTINGSTEGNOLOGIE V1

MEI/JUNIE 2026

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 27 bladsye.

Kopiereg voorbehou

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 10) sluit die nasienriglyne vir elke vraag in.

- **Bylae E, F, G** en **H** (bladsy 11 tot 27) bevat voorbeelde van oplossings vir VRAAG 1 tot 4 in programmeringskode.

- Kopieë van **Bylae 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.

BYLAE A

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					

VRAAG 1: NASIENRUBRIEK – ALGEMENE PROGRAMMERINGSVAARDIGHEDE

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
1.1	**Knoppie [1.1 - Properties]** Vir chbQ1_1 doen: • Verander opskrif ('caption') na 'Marine World' ✓ • TRUE ✓ gemerk • Font styl na [fsBold] ✓ • Enabled na FALSE✓	**4**	

1.2	**Knoppie [1.2 - Fudge]** Kry area uit cmbQ1_2 ✓ Omgeskakel na Integer ✓ iNumPieces = iArea ✓ DIV 4 ✓ iPackets = iNumPieces DIV 6 ✓ iExtra = iNumPieces MOD 6 ✓ Vertoon die afvoer in memQ1_2 in drie aparte reëls ✓ Met korrekte etiket ('label') ✓ Met korrekte veranderlikes omgeskakel na string ✓	**9**	

1.3	**Knoppie [1.3 - Hexadecimal]** Verkry sHex met gebruik van InputBox ✓ Vertoon sHex op lblQ1_3Hex ✓ Inisialiseer rDecimal en iExp na 0 ✓ Bepaal Length(sHex) ✓ Lus ('Loop') van lengte van (sHex) downto 1 ✓ Verkry karakter ✓ Case karakter of ✓ '0'..'9' skakel karakter om na heelgetal en ken toe aan iValue veranderlike ✓ 'A'..'F' ✓ ken ooreenstemmende heelgetal waarde toe aan iValue veranderlike ✓ rDecimal = rDecimal ✓ + iValue * Power(16, iExp) ✓ inkrementeer (iExp) ✓ Vertoon rDecimal omgeskakel na string in lblQ1_3Dec ✓	**14**	
1.4	**Knoppie [1.4 - Extract word]** Verkry sin wat deur gebruiker geselekteer is uit lstQ1_4 ✓ Vertoon sin in redQ1_4 ✓ Voorwaardelike lus om: ✓ Genereer ewekansige indeks waarde ✓ in reeks 1 tot lengte van sin ✓ Totdat karakter in posisie van ewekansige waarde 'n spasie is ✓ Vertoon genereerde waarde in redQ1_4 omgeskakel na string ✓ Vee teks in sin uit van 1 tot die posisie van die spasie ✓ Vind die eerste spasie in oorblywende deel van die sin ✓ Toets of nie laaste woord in die sin is nie ✓ Verkry woord van 1 tot by posisie van spasie - 1 ✓ Anders Verkry woord van 1 tot by lengte van sin -1 ✓ Vertoon die woord in redQ1_4 ✓ **Alternatief om volgende spasie te vind:** Lus ('Loop') deur die oorblywende deel van sin (1) Toets vir spasie (1)	**13**	
	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 - All creatures]** SELECT * ✓ FROM tblCreatures ✓ ORDER BY CreatureName ✓	**3**	
2.1.2	**Knoppie [2.1.2 - Shallow water]** SELECT HabitatID, Depth, MinTemp FROM tblHabitats ✓ WHERE Depth < 25 ✓ AND MinTemp >= 10 ✓	**3**	
2.1.3	**Redigeerblokkie [2.1.3 - Search habitat]** SELECT HabitatName, Description ✓ FROM tblHabitats ✓ WHERE Description LIKE ✓ "%' + sFind + '%" ✓ **AANVAAR ook:** LIKE ' + quotedstr('%' + sFind + '%')	**4**	
2.1.4	**Knoppie [2.1.4 - Calculate salinity]** SELECT HabitatID, SalinityConst, RIGHT(SalinityConst,2) ✓- LEFT(SalinityConst,2) ✓ AS Salinity ✓ FROM tblHabitats ✓	**4**	
2.1.5	**Knoppie [2.1.5 - Average length]** SELECT ~~H.~~HabitatName, ROUND(AVG(CreatureLength) ✓,2 ✓) AS AvgLength FROM tblHabitats H, tblCreatures C ✓ WHERE H.HabitatID = C.HabitatID ✓ GROUP BY HabitatName ✓ HAVING ✓ AVG(CreatureLength) > 1 ✓	**7**	
	Subtotaal:	**21**	

VRAAG 2: NASIENRUBRIEK (VERVOLG)

2.2	**Databasismanipulasie**		

2.2.1	**Knoppie [2.2.1 - Update]**
	
	Gaan na die eerste rekord in tblCreatures ✓
	Lus ('Loop') deur tblCreatures ✓
	  Toets of sCreature ✓ in
	  tblCreatures['CreatureName'] voorkom ✓
	  tblCreatures.Edit✓
	  tblCreatures[‘CreatureName'] =
	  tblCreatures[‘CreatureName'] + '*' ✓
	
	  tblCreatures.Post
	  Vertoon die CreatureName en HabitatID ✓
	  in redQ2_2 in IF-stelling ✓
	  tblCreatures.Next ✓
	Einde van lus ('loop') (tblCreatures)
		9	

2.2.2	**Knoppie [2.2.2 - Habitat summary]**
	
	Gaan na die eerste rekord in tblHabitats
	Lus ('loop') deur tblHabitats ✓
	  Stel Teller = 0 ✓
	  Gaan na die eerste rekord in tblCreatures
	  Lus ('loop') deur tblCreatures ✓
	  Toets of tblCreatures['HabitatID'] =
	  tblHabitats['HabitatID'] ✓
	  Inkrementeer Teller ✓
	  tblCreatures.Next ✓
	  Einde van lus (tblCreatures)
	  As Teller > 0 ✓
	  Vertoon HabitatName + #9 ✓ +
	  Teller omgeskakel na string ✓
	  tblHabitats.Next ✓
	Einde van lus (tblHabitats)
	
	LET WEL:
	Een punt word toegeken aan die korrekte gebruik van Tabelnaam.First en Tablenaam.Next vir elke tabel.
		10	

	Subtotaal:	**19**	
	TOTAAL AFDELING B:	**40**	

BYLAE C

VRAAG 3: NASIENRUBRIEK – OBJEK-GEÖRIENTEERDE PROGRAMMERING

SENTRUMNOMMER:		**EKSAMENNOMMER:**	
VRAAG	**BESKRYWING**	**MAKS. PUNTE**	**LEERDER-PUNT**
3.1.1	**Funksie generateTurtleID** Funksie-opskrif met string terugstuurtipe en resultaat ✓ Result := UpperCase ✓ (Kopie eerste 3 karakters van fSpecies ✓ + Kopie laaste 2 karakters van fName) ✓ + fAge omgeskakel as string ✓	**5**	
3.1.2	**Constructor Create** Die volgende moet onderkant gegewe kode getoets word ✓ Toets of TurtleID ✓ = 'None' ✓ fTurtleID := generateTurtleID ✓	**4**	
3.1.3	**Prosedure injuryObservation** Prosedure-opskrif met string parameter ✓ Toets of parameterwaarde = 'Improving' ✓ Verminder fInjurySeverity met 2 ✓ Toets of fInjurySeverity < 0 verander dit na 1 ✓ Toets of parameterwaarde = 'Unstable' ✓ fInjurySeverity = 10 ✓	**6**	
3.1.4	**Funksie getInjurySeverity** Funksie-opskrif met heelgetal-terugkeertipe✓ Result = fInjurySeverity ✓	**2**	
3.1.5	**Prosedure setRehabDays** Prosedure-opskrif ✓ met Heelgetalparameter ✓ fRehabDays = parameterwaarde ✓	**3**	
3.1.6	**Funksie estimateReleaseDate** Funksie-opskrif met datumparameter ✓ en datum terugstuurtipe ✓ Result = fRehabDays + parameterwaarde ✓	**3**	
	Subtotaal: Objekklas	**23**	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	**Knoppie [3.2.1 - Instantiate object]** objSeaTurtle:= ✓ TSeaTurtle.Create ✓ (sTurtleID, sName, sSpecies, iAge) ✓ Vertoon objSeaTurtle in redQ3_2 ✓ deur toString-metode te gebruik ✓	**5**	
3.2.2	**Knoppie [3.2.2 - Observation]** Roep injuryObservation✓ (sObservation) ✓ Toets of ✓ getInjurySeverity✓ 1 .. 4: ✓ Dae := 30 ✓ 5 .. 7: Dae := 60 ✓ 8 ..10: Dae := 120 ✓ Roep setRehabDays(Dae) ✓ Roep toString om in redQ3_2 te vertoon ✓	**9**	
3.2.3	**Knoppie [3.2.3 - Release date]** Vertoon op pnlQ3_2_3: Omgeskakel na string ✓ Roep estimatedReleaseDate ✓ Met dCurrentDate as argument ✓	**3**	
	Subtotaal Vormklas:	**17**	
	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 - Populate arrays]**
	
	AssignFile(tFile, 'Fishing.txt') ✓
	Reset(tFile) ✓
	
	Lus (‘Loop’) deur die tekslêer met regte voorwaarde ✓
	  Inkrementeer iNumEntries ✓
	  ReadIn(tFile, sTemp) ✓
	  ReadIn(tFile, sLine) ✓
	  arrSurnameName[iNumEntries] = sLine ✓
	  ReadIn(tFile, sLine)
	  Inisialiseer NumberFish en TotalWeight as 0 ✓
	  Lus ('loop') deur reëls terwyl sLine <> leeg ✓
	  Inkrementeer NumberFish ✓
	  TotalWeight = TotalWeight + sLine as Real ✓
	  ReadIn(tFile, sLine) ✓ (inisialieer en verander sLine)
	  arrNumFish[iNumEntries] = NumberFish ✓
	  arrTotalWeight[iNumEntries] = TotalWeight ✓
	CloseFile(tFile)
	
	Konsepte:
	Lêer Oop en Reset (2)
	Lus deur tekslêer (1)
	Lees 'n reël uit tekslêer (1)
	Aantal inskrywings (deelnemers) (1)
	Verkry (1) en stoor naam en van in skikking (1)
	Inisialiseer aantal visse en totale gewig (1)
	Verkry (1), bereken (1) en stoor die aantal visse in skikking (1)
	Verkry (1), bereken (1) en stoor die totale gewig in skikking (1)
		14	

4.1.2	**Knoppie [4.1.2 - Competition winners]**
	
	Lus ('loop') van 1 tot iNumEntries ✓
	  Toets elke inskrywing of aantal visse meer is as 0 ✓
	  Bereken die gemiddelde gewig:
	  arrTotalWeight[Loop] ✓ /arrNumFish[Loop] ✓
	  Toets of gemiddelde gewig gelyk is aan
	  rHighestAvg ✓
	  Voeg naam en van by wenner string ✓
	  Toets of gemiddeld groter is as rHighestAvg ✓
	  Vervang wenner string data met
	  arrSurnameName[Loop] ✓
	  Vervang rHighestAvg met die hoogste gemiddeld ✓
	Eindig lus
	Vertoon wenner string in redQ4_1 ✓
		10	

4.2	**Knoppie [4.2 - Pascal's triangle]**
	
	Lus ('Loop') iRy van 3 tot 7 ✓
	  Geneste lus ('loop') ✓ iKol van 2 tot 14 ✓
	  arrPattern[iRy,iKol] = ✓
	  arrPattern[iRy -1,iKol -1] ✓ +
	  arrPattern[iRy - 1,iKol +1] ✓
		6	

	TOTAAL AFDELING D: **GROOTTOTAAL:**	**30** **150**	

BYLAE E: OPLOSSING VIR VRAAG 1

```
unit Question1_U;

interface

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

type
  TfrmQuestion1 = class(TForm)
    grpQ1_1: TGroupBox;
    grpQ1_4: TGroupBox;
    btnQ1_1: TButton;
    lstQ1_4: TListBox;
    redQ1_4: TRichEdit;
    grpQ1_3: TGroupBox;
    lblQ1_3Hex: TLabel;
    btnQ1_3: TButton;
    lblQ1_3Dec: TLabel;
    lblHex: TLabel;
    lblDec: TLabel;
    grpQ1_2: TGroupBox;
    imgFudge: TImage;
    lblArea: TLabel;
    memQ1_2: TMemo;
    btnQ1_2: TButton;
    lblSelectSentence: TLabel;
    btnQ1_4: TButton;
    cmbQ1_2: TComboBox;
    chbQ1_1: TCheckBox;
    procedure btnQ1_1Click(Sender: TObject);
    procedure btnQ1_2Click(Sender: TObject);
    procedure btnQ1_3Click(Sender: TObject);
    procedure btnQ1_4Click(Sender: TObject);
    procedure FormShow(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);
begin
  // Vraag 1.1
  chbQ1_1.Caption := 'Marine World';
  chbQ1_1.Checked := True;
  chbQ1_1.Font.Style := [fsBold];
  chbQ1_1.Enabled := False;
end;
//=============================================================================
// Vraag 1.2                                                          9 punte
// ============================================================================
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
var
  iArea: Integer;
  iNumPieces: Integer;
  iPackets : Integer;
  iExtra : Integer;
begin
  // Provided code
  memQ1_2.Clear;
  // Vraag 1.2
  iArea := StrToInt(cmbQ1_2.Text);
  iNumPieces := iArea DIV 4;
  iPackets := iNumPieces DIV 6;
  iExtra := iNumPieces MOD 6;
  memQ1_2.Lines.Add('Total pieces of fudge: ' + IntToStr(iNumPieces));
  memQ1_2.Lines.Add('Packets of fudge: ' + IntToStr(iPackets));
  memQ1_2.Lines.Add('Remaining pieces of fudge: ' + IntToStr(iExtra));
end;
//=============================================================================
// Vraag 1.3                                                         14 punte
// ============================================================================
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
  sHex: String;
  rDecimal: Real;
  iExp, iValue, iCnt: Integer;
  cChar: Char;
begin
  // Vraag 1.3
  sHex := InputBox('Convert Hexadecimal to Decimal',
          'Enter hexadecimal value:', 'A2B');
  lblQ1_3Hex.Caption := sHex;
  rDecimal := 0;
  iExp := 0;
  for iCnt := Length(sHex) downTo 1 do
  begin
    cChar := sHex[iCnt];
```

```
    case cChar of
      '0' .. '9': iValue := StrToInt(cChar);
      'A': iValue := 10;
      'B': iValue := 11;
      'C': iValue := 12;
      'D': iValue := 13;
      'E': iValue := 14;
      'F': iValue := 15;
//'A'..'F' : iValue := ORD(cChar) - ORD('A') + 10; // Alternatief
    end;
    rDecimal := rDecimal + iValue * Power(16, iExp);
    Inc(iExp);
  end;
  lblQ1_3Dec.Caption := FloatToStrF(rDecimal, ffFixed, 8, 0);
end;

//============================================================================
// Vraag 1.4                                                    13 punte
// ============================================================================
procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var
  sSentence, sWord: String;
  iPos, iLen, iRandom, iIndex: Integer;
begin
  // Provided code
  redQ1_4.Clear;
  // Vraag 1.4
  sSentence := lstQ1_4.Items[lstQ1_4.ItemIndex];
  redQ1_4.Lines.Add(sSentence);

  repeat
    iRandom := Random(length(sSentence) + 1) + 1;
  until sSentence[iRandom] = ' ';

  redQ1_4.Lines.Add('Generated value: ' + IntToStr(iRandom));

  Delete(sSentence, 1, iRandom);
  iPos := Pos(' ', sSentence);
  if iPos > 0 then
    sWord := Copy(sSentence, 1, iPos - 1)
  else
    sWord := Copy(sSentence, 1, length(sSentence) - 1);

  redQ1_4.Lines.Add('Extracted word: ' + sWord);
end;

// ============================================================================
// Provided code
// ============================================================================
{$REGION 'Provided code - DO NOT CHANGE'}
procedure TfrmQuestion1.FormShow(Sender: TObject);
begin
  lstQ1_4.ItemIndex := 0;
end;
{$ENDREGION}
end.
```

BYLAE F: OPLOSSING VIR VRAAG 2

```
unit Question2_U;

interface

uses
  Windows, Messages, SysUtils, Variants,
  Classes, Graphics, Controls, Forms, Dialogs, DB, ADODB, DBConnect_U,
  Grids, DBGrids, StdCtrls, ExtCtrls, ComCtrls, Buttons;

type
  TfrmQuestion2 = class(TForm)
    dbgOne: TDBGrid;
    dbgMany: TDBGrid;
    dbgSQL: TDBGrid;
    pgcQuestion2: TPageControl;
    tsQuestion2_1: TTabSheet;
    tsQuestion2_2: TTabSheet;
    pnlHabitats: TPanel;
    pnlCreatures: TPanel;
    redQ2_2: TRichEdit;
    btnQ2_2_1: TButton;
    btnQ2_2_2: TButton;
    btnReset: TBitBtn;
    btnQ2_1_1: TButton;
    btnQ2_1_3: TButton;
    edtQ2_1_3: TEdit;
    btnQ2_1_2: TButton;
    gpb2_1_3: TGroupBox;
    btnQ2_1_4: TButton;
    btnQ2_1_5: TButton;
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure btnQ2_2_2Click(Sender: TObject);
    procedure btnQ2_2_1Click(Sender: TObject);
    procedure btnResetClick(Sender: TObject);
    procedure btnQ2_1_1Click(Sender: TObject);
    procedure edtQ2_1_3Change(Sender: TObject);
    procedure btnQ2_1_2Click(Sender: TObject);
    procedure btnQ2_1_4Click(Sender: TObject);
    procedure btnQ2_1_5Click(Sender: TObject);
  private
    procedure runSQL(sSQL: string);
  public
    { Public declarations }
  end;

var
  frmQuestion2: TfrmQuestion2;
  objDB: TDBConnect;
  tblCreatures, tblHabitats: TADOTable;
  sSQL : String;

implementation
{$R *.dfm}
```

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

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

  sSQL := 'SELECT * ' +
          'FROM tblCreatures ' +
          'ORDER BY CreatureName';

  //Provided code
  runSQL(sSQL);
end;
//=============================================================================
//  Vraag 2.1.2                                                        3 punte
//=============================================================================
procedure TfrmQuestion2.btnQ2_1_2Click(Sender: TObject);
begin
  //Vraag 2.1.2

  sSQL := 'SELECT HabitatID, Depth, MinTemp ' +
          'FROM tblHabitats '  +
          'WHERE Depth < 25 AND MinTemp >= 10';

  //Provided code
  runSQL(sSQL);
end;
//=============================================================================
//  Vraag 2.1.3                                                        4 punte
//=============================================================================
procedure TfrmQuestion2.edtQ2_1_3Change(Sender: TObject);
// Provided code
//DO NOT change
var
  sFind : String;
begin
  //Vraag 2.1.3
  sFind := edtQ2_1_3.Text;
  //Provided code

  sSQL := 'SELECT HabitatName, Description ' +
          'FROM tblHabitats ' +
          'WHERE Description LIKE "%' + sFind + '%"';

  //Provided code
  runSQL(sSQL);
end;
```

```
//=============================================================================
//  Vraag 2.1.4                                                      4 punte
//=============================================================================
procedure TfrmQuestion2.btnQ2_1_4Click(Sender: TObject);
begin
  //Vraag 2.1.4

  sSQL := 'SELECT HabitatID, SalinityConst, RIGHT(SalinityConst,2) -
                LEFT(SalinityConst,2) AS Salinity FROM tblHabitats ';

  //Provided code
  runSQL(sSQL);
end;
//=============================================================================
//  Vraag 2.1.5                                                      7 punte
//=============================================================================
procedure TfrmQuestion2.btnQ2_1_5Click(Sender: TObject);
begin
  //Vraag 2.1.5

  sSQL := 'SELECT H.HabitatName, ROUND(AVG(CreatureLength),2)
     AS AvgLength ' +
          'FROM tblHabitats H, tblCreatures C ' +
          'WHERE H.HabitatID = C.HabitatID ' +
          'GROUP BY HabitatName ' +
          'HAVING AVG(CreatureLength) > 1';

  //Provided code
  runSQL(sSQL);
end;
```

```
//=============================================================================
// Vraag 2.2 – Section Delphi code
//=============================================================================

//=============================================================================
//  Vraag 2.2.1                                                      9 punte
//=============================================================================
procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);
var
  sCreature: String;
begin
  // Provided code
    sCreature := cmbQ2_2_1.Text;
    redQ2_2.Clear;
    redQ2_2.Lines.Add('Creature name' + #9 + 'Habitat ID');
    redQ2_2.Lines.Add('');

  // Vraag 2.2.1
  tblCreatures.First;
  while not tblCreatures.EOF do
  begin
    if Pos( sCreature, tblCreatures['CreatureName'] ) > 0 then
    begin
      tblCreatures.Edit;
      tblCreatures['CreatureName'] := tblCreatures['CreatureName'] +'*';
      tblCreatures.Post;
      redQ2_2.Lines.Add(tblCreatures['CreatureName'] + #9 +
      tblCreatures['HabitatID']);
    end;
     tblCreatures.Next;
  end;
end;
//=============================================================================
//  Vraag 2.2.2                                                     10 punte
//=============================================================================
procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);
var
  iCount: Integer;
begin
  // Provided code
  redQ2_2.Clear;
  redQ2_2.Lines.Add('Habitat name' + #9 + 'Number of creatures');
  redQ2_2.Lines.Add('');

  // Vraag 2.2.2
  tblHabitats.First;
  while not tblHabitats.EOF do
  begin
    iCount := 0;
    tblCreatures.First;
    while not tblCreatures.EOF do
    begin
      if tblHabitats['HabitatID'] = tblCreatures['HabitatID'] then
        Inc(iCount);
      tblCreatures.Next;
    end;
```

```
    if iCount > 0 then
      redQ2_2.Lines.Add(tblHabitats['HabitatName'] + #9 +
                IntToStr(iCount));

    tblHabitats.Next;
  end;

end;

// ============================================================================
// Provided code
// ============================================================================
procedure TfrmQuestion2.btnResetClick(Sender: TObject);
begin
  objDB.RestoreDatabase;
  objDB.setupGrids(dbgOne, dbgMany, dbgSQL);

  redQ2_2.Clear;
end;

procedure TfrmQuestion2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  objDB.DisconnectFromDB;
  objDB.destroy;
end;

procedure TfrmQuestion2.FormCreate(Sender: TObject);
begin
  objDB := TDBConnect.create;
  tblCreatures := objDB.getManyTbl;
  tblHabitats := objDB.getOneTbl;

  dbgOne.DataSource := objDB.getOneDS;
  dbgMany.DataSource := objDB.getManyDS;

  objDB.setupGrids(dbgOne, dbgMany, dbgSQL);

  redQ2_2.Clear;
  redQ2_2.Paragraph.Tabcount := 2;
  redQ2_2.Paragraph.Tab[0] := 150;
  redQ2_2.Paragraph.Tab[1] := 200;
end;

procedure TfrmQuestion2.runSQL(sSQL: string);
begin
  dbgSQL.DataSource := objDB.ExecuteSQL(sSQL);
  objDB.setupGrids(dbgOne, dbgMany, dbgSQL);
end;
{$ENDREGION}

end.
```

BYLAE G: OPLOSSING VIR VRAAG 3

Objekklas:

```
unit SeaTurtle_U;

interface

uses
  SysUtils, Math;

type
  TSeaTurtle = class
  private
    fTurtleID: String;
    fName: String;
    fSpecies: String;
    fAge: Integer;
    fInjurySeverity: Integer;
    fRehabDays: Integer;

  public
    Constructor Create(sTurtleID, sName, sSpecies: String; iAge: Integer);
    function generateTurtleID: String;
    function toString: String;
    procedure injuryObservation(sObservation: String);
    function getInjurySeverity: Integer;
    procedure setRehabDays(iDays: Integer);
    function estimateReleaseDate(dDate: TDate): TDate;
  end;
implementation

{ TSeaTurtle }
// ============================================================================
// Vraag 3.1.1                                                        5 punte
// ============================================================================
function TSeaTurtle.generateTurtleID: String;
begin
  Result := UpperCase(Copy(fSpecies, 1, 3) + copy(fName, length(fName)-1))
                + IntToStr(fAge);
end;
// ============================================================================
// Vraag 3.1.2                                                        4 punte
// ============================================================================
Constructor TSeaTurtle.Create(sTurtleID, sName, sSpecies: String;
                iAge: Integer);
begin
  fTurtleID := sTurtleID;
  fName := sName;
  fSpecies := sSpecies;
  fAge := iAge;
  fInjurySeverity := 5;
  fRehabDays := 60;
  if fTurtleID = 'None' then
    fTurtleID := generateTurtleID;
end;
```

```
// ========================================================================
// Vraag 3.1.3                                                    6 punte
// ========================================================================
procedure TSeaTurtle.injuryObservation(sObservation: String);
begin
  if sObservation = 'Improving' then
  begin
      fInjurySeverity := fInjurySeverity - 2;
      if fInjurySeverity < 0 then
           fInjurySeverity := 1;
  end;
  if sObservation = 'Unstable' then
    fInjurySeverity := 10;
end;
// ========================================================================
// Vraag 3.1.4                                                    2 punte
// ========================================================================
function TSeaTurtle.getInjurySeverity: integer;
begin
  Result := fInjurySeverity;
end;
// ========================================================================
// Vraag 3.1.5                                                    3 punte
// ========================================================================
procedure TSeaTurtle.setRehabDays(iDays: Integer);
begin
  fRehabDays := iDays;
end;
// ========================================================================
// Vraag 3.1.6                                                    3 punte
// ========================================================================
function TSeaTurtle.estimateReleaseDate(dDate: TDate): TDate;
begin
  Result := dDate + fRehabDays;
end;
// ========================================================================
// Provided code
// ========================================================================
function TSeaTurtle.toString: String;
begin
  Result := fName + '#' + fTurtleID + #13 +
            'Species: ' + fSpecies + #13 +
            'Age: ' + IntToStr(fAge) + #13 +
            'Injury Severity: ' + IntToStr(fInjurySeverity) + #13 +
            'Rehabilitation duration: ' + IntToStr(fRehabDays);
end;

end.
```

Hoofvormeenheid:

```
unit Question3_U;

interface

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

type
  TfrmQuestion3 = class(TForm)
    gpbQ3_2_1: TGroupBox;
    lblTagID: TLabel;
    lblName: TLabel;
    cmbQ3_2_1: TComboBox;
    edtQ3_2_1: TEdit;
    rgpQ3_2_1: TRadioGroup;
    lblAge: TLabel;
    spnQ3_2_1: TSpinEdit;
    btnQ3_2_1: TButton;
    imgTurtle: TImage;
    gpb3_2_2: TGroupBox;
    btnQ3_2_2: TButton;
    gpbQ3_2_3: TGroupBox;
    btnQ3_2_3: TButton;
    redQ3_2: TRichEdit;
    rgpQ3_2_2: TRadioGroup;
    pnlQ3_2_3: TPanel;
    dtpQ3_2_3Current: TDateTimePicker;
    lblCurrentDate: TLabel;
    procedure cmbQ3_2_1Change(Sender: TObject);
    procedure btnQ3_2_2Click(Sender: TObject);
    procedure btnQ3_2_3Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnQ3_2_1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmQuestion3: TfrmQuestion3;
  objSeaTurtle: TSeaTurtle;

implementation

{$R *.dfm}
```

```
// ========================================================================
// Vraag 3.2.1                                                    5 punte
// ========================================================================
procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);
var
  sOneline, sName, sSpecies, sTurtleID: String;
  iPlace, iAge: Integer;
begin
  // Provided code
  sTurtleID := cmbQ3_2_1.Text;
  sName := edtQ3_2_1.Text;
  sSpecies := rgpQ3_2_1.Items[rgpQ3_2_1.ItemIndex];
  iAge := spnQ3_2_1.Value;
  redQ3_2.Clear;
  // Vraag 3.2.1
  objSeaTurtle := TSeaTurtle.Create(sTurtleID, sName, sSpecies, iAge);
  redQ3_2.Lines.Add(objSeaTurtle.toString);
end;
// ========================================================================
// Vraag 3.2.2                                                    9 punte
// ========================================================================
procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);
var
  sObservation: String;
begin
  // Provided code
  sObservation := rgpQ3_2_2.Items[rgpQ3_2_2.ItemIndex];
  // Vraag 3.2.2
  objSeaTurtle.injuryObservation(sObservation);
  case objSeaTurtle.getInjurySeverity of
    1 .. 4: objSeaTurtle.setRehabDays(30);
    5 .. 7: objSeaTurtle.setRehabDays(60);
    8 .. 10: objSeaTurtle.setRehabDays(120);
  end;

  redQ3_2.Lines.Clear;
  redQ3_2.Lines.Add(objSeaTurtle.toString)
end;
// ========================================================================
// Vraag 3.2.3                                                    3 punte
// ========================================================================
procedure TfrmQuestion3.btnQ3_2_3Click(Sender: TObject);
var
  dCurrentDate: TDate;
begin
  // Provided code
  dCurrentDate := dtpQ3_2_3Current.Date;
  // Vraag 3.2.3
  pnlQ3_2_3.caption :=
DateToStr(objSeaTurtle.estimateReleaseDate(dCurrentDate));
end;
```

```
// ============================================================================
// Provided code
// ============================================================================
{$REGION 'Provide code'}
procedure TfrmQuestion3.cmbQ3_2_1Change(Sender: TObject);
begin
  if cmbQ3_2_1.ItemIndex <> 0 then
  begin
    edtQ3_2_1.Text := Populate_U.arrNames[cmbQ3_2_1.ItemIndex];
    rgpQ3_2_1.ItemIndex := Populate_U.arrSpecies[cmbQ3_2_1.ItemIndex];
    spnQ3_2_1.Value := Populate_U.arrAges[cmbQ3_2_1.ItemIndex];
  end
  else
  begin
    edtQ3_2_1.Text := '';
    rgpQ3_2_1.ItemIndex := -1;
    spnQ3_2_1.Value := 0;
  end
end;

procedure TfrmQuestion3.FormShow(Sender: TObject);
begin
  dtpQ3_2_3Current.Date := DATE;
end;
{$ENDREGION}
end.
```

BYLAE H: OPLOSSING VIR VRAAG 4

```
unit Question4_U;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, ComCtrls, jpeg, StdCtrls, Math, Grids;
type
  TfrmQuestion4 = class(TForm)
    pgcQ4: TPageControl;
    tsQ4_1: TTabSheet;
    tsQ4_2: TTabSheet;
    imgFishing: TImage;
    pnlQ4_1: TPanel;
    redQ4_1: TRichEdit;
    btnQ4_1_1: TButton;
    btnQ4_1_2: TButton;
    btnQ4_2: TButton;
    sgdQ4_2: TStringGrid;
    procedure btnQ4_2Click(Sender: TObject);
    procedure btnQ4_1_1Click(Sender: TObject);
    procedure btnQ4_1_2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure displayInfo;
  private
    { Private declarations }
    procedure display2D;
  public
    { Public declarations }
  end;
var
  frmQuestion4: TfrmQuestion4;
  arrSurnameName: array [1 .. 30] of String;
  arrNumFish: array [1 .. 30] of Integer;
  arrTotalWeight: array [1 .. 30] of Real;
  iNumEntries: Integer = 0;

  arrPattern: array [1 .. 7, 1 .. 15] of Integer =
   ((0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 1, 0, 1,
      0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
    (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0,
      0, 0, 0, 0, 0, 0), (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0),
    (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0));
implementation
{$R *.dfm}
{ TfrmQuestion4 }
```

```
// ============================================================================
// Vraag 4.1.1                                                        14 punte
// ============================================================================
procedure TfrmQuestion4.btnQ4_1_1Click(Sender: TObject);
var
  tFile: Textfile;
  sLine, sTemp: String;
  iNumFish: Integer;
  rTotalWeight: Real;
begin
  //Vraag 4.1.1
```

```
  AssignFile(tFile, 'Fishing.txt');
  Reset(tFile);
  while NOT Eof(tFile) do
  begin
    Inc(iNumEntries);
    Readln(tFile, sTemp);
    Readln(tFile, arrSurnameName[iNumEntries]);
    Readln(tFile, sLine);
    iNumFish := 0;
    rTotalWeight := 0;
    while sLine <> '' do
    begin
      inc(iNumFish);
      rTotalWeight := rTotalWeight + StrToFloat(sLine);
      Readln(tFile, sLine);
    end;
   arrNumFish[iNumEntries] := iNumFish;
   arrTotalWeight[iNumEntries] := rTotalWeight;
  end;
  CloseFile(tFile);

  //Provided code
  displayInfo;
end;
// ============================================================================
// Vraag 4.1.2                                                      10 punte
// ============================================================================
procedure TfrmQuestion4.btnQ4_1_2Click(Sender: TObject);
var
  rHighestAvg, rAvg: Real;
  iCnt, iNumFish: Integer;
  sOutput: String;
begin
  //Provided code
  redQ4_1.Clear;
  rHighestAvg := 0;

  //Vraag 4.1.2
  for iCnt := 1 to iNumEntries do
  begin
    if arrNumFish[iCnt] > 0 then
    begin
      rAvg := arrTotalWeight[iCnt] / arrNumFish[iCnt];

      if rAvg = rHighestAvg then
        sOutput := sOutput + arrSurnameName[iCnt] + #13;

      if rAvg > rHighestAvg then
      begin
        sOutput := arrSurnameName[iCnt] + #13;
        rHighestAvg := rAvg;
      end;
    end;
  end;
  redQ4_1.Lines.Add(sOutput);

 //Provided code
  redQ4_1.Lines.Add('Highest average: ' +
                      FloatToStrF(rHighestAvg,ffFixed,8,2));
end;
```

```
// ========================================================================
// Vraag 4.2                                                    6 punte
// ========================================================================
procedure TfrmQuestion4.btnQ4_2Click(Sender: TObject);
var
  iRow, iCol: Integer;
begin
  //Vraag 4.2

  for iRow := 3 to 7 do
    for iCol := 2 to 14 do
      arrPattern[iRow, iCol] := arrPattern[iRow - 1, iCol - 1] +
                                arrPattern[iRow - 1, iCol + 1];
// Provided code
  display2D;
end;

// ========================================================================
// Provided code
// ========================================================================

{$REGION 'Provided code - Do NOT change'}

procedure TfrmQuestion4.display2D;
var
  iRow, iCol: Integer;
  iLeftPad, iRightPad: Integer;
begin
  for iRow := 0 to 6 do
    for iCol := 0 to 14 do
      if arrPattern[iRow + 1, iCol + 1] = 0 then
        sgdQ4_2.cells[iCol, iRow] := ''
      else
        if Length(IntToStr(arrPattern[iRow+1, iCol+1])) = 1 then
          sgdQ4_2.Cells[iCol,iRow] := '  ' +
                          IntToStr(arrPattern[iRow+1, iCol+1])
        else
          sgdQ4_2.Cells[iCol,iRow]:=' '+ IntToStr(arrPattern[iRow+1, iCol+1]);
end;

procedure TfrmQuestion4.displayInfo;
var
  iCnt : integer;
begin
   redQ4_1.Clear;
   redQ4_1.Paragraph.TabCount := 3;
   redQ4_1.Paragraph.Tab[0] := 50;
   redQ4_1.Paragraph.Tab[1] := 150;
       redQ4_1.Lines.Add('Surname and Name' + #9 + 'Number of fish' + #9 +
                          'Total weight');
   for iCnt := 1 to 26 do
      redQ4_1.Lines.Add(arrSurnameName[iCnt] + #9 + IntToStr(arrNumFish[iCnt])
                          + #9#9#9 + FloatToStr(arrTotalWeight[iCnt]));
end;
```

```
procedure TfrmQuestion4.FormCreate(Sender: TObject);
var
  iRow, iCol: Integer;
  iLeftPad, iRightPad: Integer;
begin
    for iRow := 0 to 1 do
    for iCol := 0 to 14 do
      if arrPattern[iRow + 1, iCol + 1] = 0 then
        sgdQ4_2.cells[iCol, iRow] := ''
      else
        if Length(IntToStr(arrPattern[iRow+1, iCol+1])) = 1 then
          sgdQ4_2.Cells[iCol,iRow] := '  ' +
                          IntToStr(arrPattern[iRow+1, iCol+1])
        else
          sgdQ4_2.Cells[iCol,iRow] := ' ' +
                          IntToStr(arrPattern[iRow+1, iCol+1]);
end;

{$ENDREGION}

end.
```

Kopiereg voorbehou