basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2023

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 24 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 24) 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	**Knoppie [1.1 – Display name and age]** Onttrek naam uit edtQ1_1 en stoor in sName veranderlike ✓ Onttrek ouderdom uit spnQ1_1 en stoor in iAge veranderlike ✓ Vertoon deur 'n afvoerdialoogblokkie te gebruik die naam ✓ die ouderdom, omgeskakel na 'n string ✓ met #13 ✓ vir volgende reël	**5**	
1.2	**Knoppie [1.2 – Hockey teams]** Verklaar heelgetal-veranderlike(s) ✓ Onttrek die getal leerders uit die edit box, ✓ omgeskakel na 'n heelgetal ✓ Bereken die aantal spanne Getal leerders DIV ✓ PLAYERS ✓ (gebruik konstante) Bereken die aantal reserwes Getal leerders MOD ✓ PLAYERS ✓ Vertoon die aantal spanne en aantal reserwes ✓ Omgeskakel na string in die memo memQ1_2 ✓ **LET WEL**: Die gegewe konstante PLAYERS moet ten minste eenmaal gebruik word	**9**	
1.3	**Knoppie [1.3 – Calculate]** `Formule: d = Sqrt ✓ (power ✓ ((X - Y),4) ) ✓` Vertoon die waarde van **d** in edtQ1_3 ✓ Geformateer met 3 desimale plekke ✓	**5**	

1.4	**Knoppie [1.4 – Marathon results]** case iPosisie of ✓ Opsie 1 ✓ Vertoon 'You receive a gold medal' ✓ Opsie 2 tot 3 ✓ Vertoon 'You receive a silver medal' Opsie 4 .. 20 ✓ Vertoon 'You receive a bronze medal' Else Vertoon 'You receive a participation certificate' ✓ End // case	**6**	
1.5	**Knoppie [1.5 – Average mark]** Verklaar lêerveranderlike ✓ Inisialiseer iTotaal en iTel na 0 ✓ Reset (tFile) ✓ AssignFile (tFile, 'Details.txt') ✓ Terwyl nie einde van lêer nie ✓ Lees reel uit teksleer ✓ Vind die posisie van die # skeiteken ✓ Onttrek punt uit reel ✓ deur regte indekse te gebruik ✓ Skakel punt om ne heelgetal ✓en voeg by totaal ✓ Inkrementeer itel ✓ Eindig while Maak lêer toe ✓ Bereken gemiddeld deur iTotaal en iTel te gebruik ✓ Vertoon gemiddelde punt in pnlQ1_5 tot naaste heelgetal ✓	**15**	
	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 – Large enrolments]**
	`SELECT * ✓ FROM tblCourses ✓`
	`WHERE MaxStudents > 99 ✓`
	
	Alternatief: `MaxStudents >= 100`
		3	

2.1.2	**Knoppie [2.1.2 – Lecturer gender]**
	`SELECT LecturerName, LecturerSurname, ✓`
	`LEFT (Gender, 1) ✓ AS [Gender (M/F)] ✓`
	`FROM tblLecturers ✓`
		4	

2.1.3	**Knoppie [2.1.3 – Multilingual lecturers]**
	`SELECT CourseID, CourseName`
	`FROM tblLecturers, tblCourses ✓`
	`WHERE (tblLecturers✓.LecturerID =`
	`tblCourses.LecturerID) ✓ AND ✓`
	`(Multilingual = True) ✓`
	`ORDER BY CourseName ✓`
		6	

2.1.4	**Knoppie [2.1.4 – Lecturer salaries]**
	`SELECT LecturerID,`
	`FORMAT(Count(*) ✓ * 10000 ✓, "CURRENCY") ✓`
	`AS [Salary]`
	`FROM tblCourses ✓`
	`GROUP BY LecturerID ✓`
		5	

2.1.5	**Knoppie [2.1.5 – Change online option]**
	`UPDATE tblCourses ✓ SET OnlineOption = FALSE ✓`
	`WHERE CourseName Like ✓ "%Programming%" ✓`
		4	

	Subtotaal:	**22**	

VRAAG 2: NASIENRUBRIEK – VERVOLG

2.2	**Databasismanipulasie**		
2.2.1	**Knoppie [2.2.1 – Average duration of courses]** Gaan na die eerste rekord in tblLecturers ✓ Stap met lus ('loop') deur tblLecturers ✓ Vertoon opskrif deur LecturerID, LecturerName, en LecturerSurname in die regte formaat te gebruik ✓ Inisialiseer Teller en Som veranderlikes ✓ Gaan na die eerste rekord in tblCourses ✓ Stap met lus ('loop') deur tblCourses ✓ Toets of (tblLecturers ['LecturerID'] = tblCourses['LecturerID']) ✓ Inkrementeer Teller ✓ en Tel duration by Som ✓ Vertoon die Teller-waarde en kursus se naam ✓ tblCourses.Next ✓ Eindig lus (tblCourses) Bereken gemiddelde duur: Som / Teller ✓ Vertoon gemiddelde duur geformatteer na twee desimale ✓ tblLecturers.Next ✓ Eindig lus (tblLecturers)	**14**	
2.2.2	**Knoppie [2.2.2 – Register new lecturer]** tblLecturers.Insert; ✓ tblLecturers['LecturerID'] := 'ZT032'; tblLecturers['LecturerName'] := 'Zander'; tblLecturers['LecturerSurname'] := 'Thomas'; tblLecturers['Gender'] := 'Male'; ✓ tblLecturers['Multilingual'] := True; ✓ tblLecturers.Post; ✓	**4**	
	Subtotaal:	**18**	
	TOTAAL AFDELING B:	**40**	

BYLAAG C

VRAAG 3: NASIENRUBRIEK – OBJEK-GEÖRIENTEERDE PROGRAMMERING

SENTRUMNOMMER:	**EKSAMENNOMMER:**		
VRAAG	**BESKRYWING**	**MAKS. PUNTE**	**LEERDER-PUNT**
3.1.1	**Constructor Create** Stel attribute (fSchoolName, fTotalLearners, fPublicSchool) ✓ Na regte parameters ✓ Ken 'Z' toe aan fRating ✓	**3**	
3.1.2	**Function getPublicSchool** Funksie-opskrif met Boolese waarde as terugstuurtipe ✓ result = fPublicSchool ✓	**2**	
3.1.3	**Procedure updateRating** Prosedure-opskrif ✓ met heelgetalparameter ✓ slaagPersentasie = leerdersGeslaag / fTotalLearners ✓ as slaagPersentasie >= 80 ✓ fRating = 'A' ✓ anders if slaagPersentasie >= 60 ✓ fRating = 'B' ✓ anders fRating = 'C' ✓ Aanvaar ook ander oplossings	**8**	
3.1.4	**Function calcFunding** Funksie-opskrif met real as terugstuur-datatipe ✓ befondsing = fTotalLearners ✓ * 145.50 ✓ result = befondsing ✓	**4**	

3.1.5	**Function toString** met string as terugstuur-datatipe Bou string met fSchoolName en '----------' op volgende reel line ✓ Voeg 'Totale getal leerders: ' en fTotalLearners by die string ✓ Voeg 'Gradering: ' en fRating by die string ✓ As fPublicSchool ✓ Voeg 'Publieke skool' by ✓ Anders Voeg 'Privaatskool' by ✓ Stuur string terug ✓	**7**	
	Subtotaal: Objekklas	**24**	

VRAAG 3: NASIENRUBRIEK (VERVOLG)

VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER - PUNT
3.2.1	**Knoppie [3.2.1 – Instantiate Object]** Onttrek skool se naam uit edtQ3_2_1 ✓ Onttrek getal leerders uit spnQ3_2_1 ✓ Onttrek publieke skool uit chbQ3_2_1 ✓ objSchool ✓ := TSchool.create ✓ Gebruik drie argumente in regte volgorde ✓ `(sSkoolNaam, iAantLeerders, bPubliekeSkool)` Vertoon objek objSchool in redQ3 deur toString-metode te gebruik ✓	**7**	
3.2.2	**Knoppie [3.2.2 – Rating]** Onttrek getal leerders wat geslaag het uit spnQ3_2_2 ✓ Roep updateRating ✓ met regte argument ✓ Vertoon objSchool in redQ3 deur toString-metode te gebruik ✓	**4**	
3.2.3	**Knoppie [3.2.3 – Funding]** Toets of getPublicSchool = TRUE ✓ Roep calcFunding-metode ✓ En vertoon in geldeenheid (currency)-formaat ✓ in redQ3 met boodskap ✓ anders Vertoon boodskap – No funding available ✓	**5**	
	Subtotaal Vormklas:	**16**	
	TOTAAL AFDELING C:	**40**	

BYLAAG D

VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING

SENTRUMNOMMER:	**EKSAMENNOMMER:**		
VRAAG	**BESKRYWING**	**MAKS. PUNTE**	**LEERDER-PUNT**
4.1	**Knoppie [4.1 – Codes]** Lus ('loop') van 1 tot lengte van skikking (of 5) ✓ Inisialiseer sReel ← leë string ✓ Lus ('loop') van 1 tot ✓ die lengte van kode ✓ Toets of karakter in die kode ✓ 'n letter ('a'.. 'z' OR 'A'.. 'Z') ✓ of 'n syfer ('0'.. '9') ✓ is Voeg die karakter by die sReel-afvoerstring ✓ Eindig binneste lus Bepaal die aantal spesiale karakters wat verwyder is Lengte(arrCodes[i]) ✓– Lengte(sLine) ✓ // Of gebruik 'n teller in die binneste lus Voeg die sReel-kode ✓ in die list box in die regte format met hakies en aantal karakters wat verwyder is ✓ Eindig buitenste lus	**12**	
4.2.1	**Knoppie [4.2.1 – Extra IT periods]** Lus ('loop') iCnt van 1 tot 4 (Maandag – Donderdag) ✓ Stel teller op 1 ✓ Terwyl (die sel nie leeg is nie) ✓ Vermeerder teller met 1 ✓ Ken 'IT' ✓ toe aan arrTimeTable[iCnt, teller] ✓ Eindig lus	**6**	

4.2.2	**Knoppie [4.2.2 – Group IT]**
	
	Lus I van 1 tot 4 (Maandag – Donderdag) ✓
	Inisialiseer Teller ✓
	Lus J van 1 tot lengte van arrTimeTable[I] ✓
	As arrTimeTable[I, J] = 'IT' ✓
	As Teller = 1 ✓
	Stoor indeks J (J_1) by eerste voorkoms van 'IT' ✓
	As Teller = 2 ✓
	Stoor indeks J (J_2)by tweede voorkoms van 'IT' ✓
	// ruil ander vakkode met IT
	Stel sTemp na arrTimeTable[I, J_1+1 ✓] ✓
	Stel arrTimeTable[I, J_1+1] na arrTimeTable[I,J_2] ✓
	Stel arrTimeTable[I, J_2] na sTemp ✓
	Eindig binneste lus
	Eindig buitenste lus
	
	Konsepte
	Skep 'n veranderlike om boek te hou van die posisie van IT
	Lus deur rye (1 tot 4) //1
	Inisialiseer Teller-veranderlike //1
	Lus deur die kolomme //1
	Toets of die selwaarde = 'IT' //1
	Stoor die indeks/posisie van eerste voorkoms //2
	Stoor die indeks/posisie van tweede voorkoms //2
	
	Ruil die vakkode na die eerste voorkoms met die eerste voorkoms van IT met die tweede voorkoms van IT //4
		12	

	TOTAAL AFDELING D:	**30**	

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

```
unit Question1_u;

interface

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

type
  TfrmQuestion1 = class(TForm)
    grpQ1_2: TGroupBox;
    btnQ1_2: TButton;
    grpQ1_1: TGroupBox;
    edtQ1_1: TEdit;
    spnQ1_1: TSpinEdit;
    lblQ1_1_Name: TLabel;
    lblQ1_1_Age: TLabel;
    btnQ1_1: TButton;
    grpQ1_3: TGroupBox;
    imgQ1_3: TImage;
    btnQ1_3: TButton;
    edtQ1_3: TEdit;
    Label3: TLabel;
    grpQ1_5: TGroupBox;
    Label4: TLabel;
    Label5: TLabel;
    cmbQ1_5: TComboBox;
    btnQ1_5: TButton;
    grpQ1_: TGroupBox;
    btnQ1_4: TButton;
    pnlQ1_5: TPanel;
    Label6: TLabel;
    edtQ1_2: TEdit;
    memQ1_2: TMemo;
    lblQ1_4: TLabel;
    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}
```

```
// ============================================================================
// 1.1 Display name and age                                          5 marks
// ============================================================================

procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);
var
  sName: String;
  iAge: integer; // Provided code
begin
  sName := edtQ1_1.Text;
  iAge := spnQ1_1.Value;
  ShowMessage(sName + #13 + IntToStr(iAge));
end;

// ============================================================================
// 1.2 Hockey teams                                                  9 marks
// ============================================================================

procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);
const
  PLAYERS = 11;
var
  iAantLeerders, iAantSpanne, iAantReserves: integer;
begin
  // Provided code
  memQ1_2.Clear;
  //-----------------------------------------
  iAantLeerders := StrToInt(edtQ1_2.Text);
  iAantSpanne := iAantLeerders DIV PLAYERS;
  iAantReserves := iAantLeerders MOD PLAYERS;
  memQ1_2.Lines.Add('Number of hockey teams: ' + IntToStr(iAantSpanne));
  memQ1_2.Lines.Add('Number of learners on reserve list: ' + IntToStr
      (iAantReserves));end;

// ============================================================================
// 1.3 Calculate                                                     5 marks
// ============================================================================

procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);
var
  rX, rY: real; // Provided code
  rD: real;
begin
  // Provided code
  rX := 12.46;
  rY := 8.54;

  rD := sqrt(power((rX - rY), 4));
  edtQ1_3.Text := FloatToStrF(rD, ffFixed, 8, 3);
end;
```

```
// ============================================================================
// 1.4 Marathon results                                                6 marks
// ============================================================================

procedure TfrmQuestion1.btnQ1_4Click(Sender: TObject);
var
  iPosisie: integer; // Provided code
begin
// Provided code
  iPosisie := StrToInt(InputBox('Marathon results',
      'Enter the position the athlete achieved', '1'));
// -------------------------------------------------
  case iPosisie of
    1:       lblQ1_4.Caption := 'You receive a gold medal.';
    2, 3:    lblQ1_4.Caption := 'You receive a silver medal.';
    4 .. 20: lblQ1_4.Caption := 'You receive a bronze medal.'
    else lblQ1_4.Caption := 'You receive a participation certificate.';
  end;
end;

// ============================================================================
// 1.5 Average mark                                                   15 marks
// ============================================================================

procedure TfrmQuestion1.btnQ1_5Click(Sender: TObject);
var
  tLeer: TextFile;
  sReel: String;
  iTotaal, iPunt, iTel, iPosHash: integer;
  rGemid: real;
begin
  iTotaal := 0;
  iTel := 0;
  AssignFile(tLeer, 'Details.txt');
  Reset(tLeer);
  while NOT(eof(tLeer))do
   begin
      readln(tLeer, sReel);
      iPosHash := pos('#', sReel);
      iPunt := StrToInt(copy(sReel, iPosHash + 1, 2));
      iTotaal := iTotaal + iPunt;
      inc(iTel);
   end;
  closeFile(tLeer);
  rGemid := iTotaal / iTel;
  pnlQ1_5.Caption := FloatToStrF(rGemid, ffFixed, 3, 0);
end;

end.
```

BYLAAG F: OPLOSSING VIR VRAAG 2

```
// ============================================================================
// 2.1 - Afdeling: SQL-stellings
// ============================================================================

// ============================================================================
//  2.1.1 Large courses                                                3 marks
// ============================================================================

     sSQL1 := 'SELECT * ' +
              'FROM tblCourses ' +
              'WHERE MaxStudents > 99';

// ============================================================================
//  2.1.2 Lecturer gender                                              4 marks
// ============================================================================

     sSQL2 := 'SELECT LecturerName, LecturerSurname, ' +
              'Left(Gender, 1) AS [Gender (M/F)]' +
              'FROM tblLecturers';

// ============================================================================
//  2.1.3 Multilingual lecturers                                       6 marks
// ============================================================================

     sSQL3 := 'SELECT CourseID, CourseName ' +
              'FROM tblLecturers , tblCourses ' +
              'WHERE (tblLecturers.LecturerID = tblCourses.LecturerID) AND
               (Multilingual = True) ' +
              'ORDER BY CourseName';
// ============================================================================
//  2.1.4 Lecturer salaries                                            5 marks
// ============================================================================

     sSQL4 := 'SELECT LecturerID, ' +
              'FORMAT(Count(*)*10000, "CURRENCY") ' +
              'AS [Salary] ' +
              'FROM tblCourses ' +
              'GROUP BY LecturerID';

// ============================================================================
//  2.1.5 Change online option                                         4 marks
// ============================================================================

     sSQL5 := 'UPDATE tblCourses SET OnlineOption = FALSE ' +
              'WHERE CourseName LIKE "%Programming%"';
```

```
// ============================================================================
// 2.2 – Afdeling: Delphi-kode
// ============================================================================

// ============================================================================
// 2.2.1 Average duration of courses                                  14 marks
// ============================================================================
procedure TfrmQuestion2.btnQ2_2_1Click(Sender: TObject);
var
  iTelKursusse, iSomTydsduur: integer;
  rGemTydsduur: real;
begin
  // Provided code
  redQ2_2_1.Clear;

  // 2.2.1 Average duration of courses

  tblLecturers.First;
  while NOT tblLecturers.Eof do
  begin
    redQ2_2_1.Lines.Add(tblLecturers['LecturerID'] + ': ' + tblLecturers
        ['LecturerName'] + ' ' + tblLecturers['LecturerSurname']);
    tblCourses.First;
    iTelKursusse := 0;
    iSomTydsduur := 0;
    while NOT tblCourses.Eof do
    begin
      if tblLecturers['LecturerID'] = tblCourses['LecturerID'] then
      begin
        inc(iTelKursusse);
        redQ2_2_1.Lines.Add(IntToStr(iTelKursusse) + '. ' + tblCourses
            ['CourseName']);
        iSomTydsduur := iSomTydsduur + tblCourses['Duration'];
      end;
      tblCourses.Next;
    end;
    rGemTydsduur := iSomTydsduur / iTelKursusse;
    redQ2_2_1.Lines.Add('Average duration of courses: ' + #9 + FloatToStrF
        (rGemTydsduur, ffFixed, 8, 2) + #13);
    tblLecturers.Next;
  end;
end;
// ============================================================================
// 2.2.2 Register new lecturer                                         4 marks
// ============================================================================
procedure TfrmQuestion2.btnQ2_2_2Click(Sender: TObject);
begin
  tblLecturers.Insert;
  tblLecturers['LecturerID'] := 'ZT032';
  tblLecturers['LecturerName'] := 'Zander';
  tblLecturers['LecturerSurname'] := 'Thomas';
  tblLecturers['Gender'] := 'Male';
  tblLecturers['Multilingual'] := True;
  tblLecturers.Post;
end;
```

```
// ============================================================================
// {$ENDREGION}
// ============================================================================
// {$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}
// ============================================================================

procedure TfrmQuestion2.FormClose(Sender: TObject; var Action:
TCloseAction);
begin
  // Disconnects from database and closes all open connections
  dbCONN.dbDisconnect;
end;

procedure TfrmQuestion2.FormCreate(Sender: TObject);
begin
  redQ2_2_1.Paragraph.TabCount := 2;
  redQ2_2_1.Paragraph.Tab[0] := 100;
  redQ2_2_1.Paragraph.Tab[1] := 150;
  redQ2_2_1.Paragraph.Tab[2] := 200;
end;

procedure TfrmQuestion2.FormShow(Sender: TObject);
begin
  // Sets up the connection to database and opens the tables.
  dbCONN := TConnection.Create;
  dbCONN.dbConnect;
  tblLecturers := dbCONN.tblOne;
  tblCourses := dbCONN.tblMany;
  dbCONN.setupGrids(dbgLecturers, dbgCourses, dbgrdSQL);
  pgcDBAdmin.ActivePageIndex := 0;
end;
// ============================================================================
// {$ENDREGION}

end.
```

BYLAAG G: OPLOSSING VIR VRAAG 3

Objekklas:

```
unit School_U;

interface

type
  TSchool = class(TObject)
  private
  var
    fSchoolName: String;
    fTotalLearners: Integer;
    fPublicSchool: boolean;
    fRating: char;
  public
    // Provide code
    constructor create(sSchoolName: String; iTotalLearners: integer;
      bPublicSchool: Boolean);
    // Code here

    function getPublicSchool: boolean;
    procedure updateRating(iLearnersPassed: integer);
    function calcFunding: real;
    function toString: String;
  end;

implementation

uses
  SysUtils, Math;
// ============================================================================
// 3.1.1 Constructor Create                                          3 marks
// ============================================================================

constructor TSchool.create(sSchoolName: String; iTotalLearners: integer;
  bPublicSchool: boolean);
begin
  // 3.1.1 Contructor Create
  fSchoolName := sSchoolName;
  fTotalLearners := iTotalLearners;
  fPublicSchool := bPublicSchool;
  fRating := 'Z';
end;
// ============================================================================
// 3.1.2 Function getPublicSchool                                    2 marks
// ============================================================================

function TSchool.getPublicSchool: boolean;
begin
  Result := fPublicSchool;
end;
```

```
// ============================================================================
// 3.1.3 Procedure updateRating                                        8 marks
// ============================================================================

procedure TSchool.updateRating(iLeerdersGeslaag: integer);
var
  rSlaagPer: real;
begin
  rSlaagPer := iLeerdersGeslaag / fTotalLearners * 100;

  if rSlaagPer >= 80 then
  begin
    fRating := 'A';
  end
  else if (rSlaagPer >= 60) AND (rSlaagPer < 80) then
  begin
    fRating := 'B';
  end
  else
  begin
    fRating := 'C';
  end;
end;

// ============================================================================
// 3.1.4 Function calcFunding                                          4 marks
// ============================================================================

function TSchool.calcFunding: real;
var
  rFondse: real;
begin
  rFondse := 145.50 * fTotalLearners;
  Result := rFondse;
end;function TSchool.calcFunding: real;

// ============================================================================
// 3.1.5 Function toString                                             7 marks
// ============================================================================

function TSchool.toString: String;
var
  sAfvoerStr: String;
begin
  sAfvoerStr := fSchoolName + #13 + '--------------------------' + #13;
  sAfvoerStr := sAfvoerStr + 'Total number of learners: ' +
     IntToStr(fTotalLearners)+ #13;
  sAfvoerStr := sAfvoerStr + 'Rating: ' + fRating + #13;
  if fPublicSchool then
    sAfvoerStr := sAfvoerStr + 'Public school ' + #13
  else
    sAfvoerStr := sAfvoerStr + 'Private school ' + #13;
  Result := sAfvoerStr;
end;

end.
```

Hoofvormeenheid:

```
unit Question3_U;

interface

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

type
  TfrmQuestion3 = class(TForm)
    gbxQ3_2_1: TGroupBox;
    gbxQ3_2_3: TGroupBox;
    redQ3: TRichEdit;
    btnQ3_2_1: TButton;
    gbxQ3_2_2: TGroupBox;
    btnQ3_2_2: TButton;
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ3_2_3: TButton;
    Image1: TImage;
    Label6: TLabel;
    edtQ3_2_1: TEdit;
    Label2: TLabel;
    spnQ3_2_1: TSpinEdit;
    chbQ3_2_1: TCheckBox;
    Label1: TLabel;c
    sedQ3_2_2: TSpinEdit;
    procedure btnQ3_2_1Click(Sender: TObject);
    procedure btnQ3_2_2Click(Sender: TObject);
    procedure btnQ3_2_3Click(Sender: TObject);
  private

  public

  end;

var
  frmQuestion3: TfrmQuestion3;

implementation

{$R *.dfm}

uses
  School_U;

var
  objSchool: TSchool;
```

```
// ============================================================================
// 3.2.1 Instantiate object                                          7 marks
// ============================================================================

procedure TfrmQuestion3.btnQ3_2_1Click(Sender: TObject);
var
  sSchoolName : String;
  iNumLearners : integer;
  bPublicSchool : boolean;
begin
  // Provided code
  redQ3.Clear;

  // 3.2.1 Instantiate object
  sSchoolName := edtQ3_2_1.Text;
  iNumLearners := spnQ3_2_1.Value;
  bPublicSchool := chbQ3_2_1.Checked;

  objSchool := TSchool.create(sSchoolName, iNumLearners, bPublicSchool);
  redQ3.Lines.Add(objSchool.toString);
end;
// ============================================================================
// 3.2.2 Rating                                                      4 marks
// ============================================================================

procedure TfrmQuestion3.btnQ3_2_2Click(Sender: TObject);
var
  iAantGeslaag: integer;
begin
  // Provided code
  redQ3.Clear;

  // 3.2.2 Rating
  iAantGeslaag := spnQ3_2_2.Value;
  objSchool.updateRating(iAantGeslaag);
  redQ3.Lines.Add(objSchool.toString);
end;

// ============================================================================
// 3.2.3 Funding                                                     5 marks
// ============================================================================

procedure TfrmQuestion3.btnQ3_2_3Click(Sender: TObject);
begin

  // 3.2.3 Funding
  if objSchool.getPublicSchool then
    redQ3.Lines.Add('Public school will receive ' + FloatToStrF
        (objSchool.calcFunding, ffCurrency, 8, 2))
  else
      redQ3.Lines.Add('No funding available ');
end;

end.
```

BYLAAG H: OPLOSSING VIR VRAAG 4

```
unit Question4_U;

interface

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

type
  TfrmQuestion4 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    btnQ4_2_2: TButton;
    redQ4: TRichEdit;
    GroupBox1: TGroupBox;
    btnQ4_2_1: TButton;
    pgcQ4: TPageControl;
    tshQ4_1: TTabSheet;
    tshQ4_2: TTabSheet;
    btnQ4_1: TButton;
    lstQ4_1: TListBox;
    GroupBox2: TGroupBox;
    procedure btnQ4_2_2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btnQ4_2_1Click(Sender: TObject);
    procedure btnQ4_1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
    procedure populate;
    procedure display;
  end;
var
  frmQuestion4: TfrmQuestion4;

  // Provided code for Question 4.1
  arrCodes: array [1 .. 5] of String =
           ('An7J*Q#D&N', 'pL78K#$.%BV', '89@FGh0&Y56#$Q','Bn4m321&*#T',
            'P2QwER%$#a');

  // Provided code for Question 4.2
  arrDays: array [1 .. 5] of String = ('Mon.', 'Tue.', 'Wed.', 'Thu.',
'Fri.');
  arrSubjectCodes: array [1 .. 5] of String =
                                      ('IT','HL', 'ACC', 'PHY', 'MAT' );
  arrTimeTable: array [1 .. 5, 1 .. 7] of String;

implementation
```

```
// ============================================================================
// 4.1 Codes                                                          12 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_1Click(Sender: TObject);
var
  I, J, iAantSpesKar: integer;
  sReel: String;
begin
  // 4.1 Codes
  for I := 1 to length(arrCodes) do
  begin
    sReel := '';
    for J := 1 to length(arrCodes[I]) do
    begin
      if arrCodes[I][J] IN ['A' .. 'Z', 'a' .. 'z', '0' .. '9'] then
      begin
        sReel := sReel + arrCodes[I][J];
      end;
    end;
    iAantSpesKar := length(arrCodes[I]) - length(sReel);
    lstQ4_1.Items.Add(sReel + '(' + intToStr(iAantSpesKar) + ')');
  end;
end;

// ============================================================================
// 4.2.1  Extra IT periods                                            6 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_2_1Click(Sender: TObject);
var
  iRy, iKol: integer;
begin
  // 4.2.1 Extra IT periods
  for iRy := 1 to 4 do
  begin
    iKol := 1;
    While NOT(arrTimeTable[iRy, iKol] = '') do
    begin
      inc(iKol);
    end;
    arrTimeTable[iRy, iKol] := 'IT';
  end;
  // Provided code
  display;
end;

// ============================================================================
// 4.2.2 Group IT                                                    12 marks
// ============================================================================
procedure TfrmQuestion4.btnQ4_2_2Click(Sender: TObject);
var
  I: integer;
  J: integer;
  iTel, iEerste, iTweede: integer;
  sTemp: String;

begin
```

```
  // 4.2.2 Group IT
  for I := 1 to 4 do
  begin
    iTel := 0;
    for J := 1 to 7 do
    Begin
      if arrTimeTable[I, J] = 'IT' then
      begin
        inc(iTel);
        if iTel = 1 then
          iEerste := J + 1;
        if iTel = 2 then
        begin
          iTweede := J;
          sTemp := arrTimeTable[I, iEerste];
          arrTimeTable[I, iEerste] := arrTimeTable[I, iTweede];
          arrTimeTable[I, iTweede] := sTemp;
        end;
      end;
    end;
  end;
  // Provided code
  display;
end;

// ============================================================================
// Provided code - Do not change
// ============================================================================

procedure TfrmQuestion4.populate;
var
  sSubjCode: String;
  iPeriod, iRand, iRow, iCol, iCnt: integer;
  arrLocal: array [1 .. 5] of String;
begin
  for iCnt := 1 to 5 do
  begin
    repeat
      iRand := RandomRange(1, 6);
      if length(arrLocal[iRand]) = 0 then
        arrLocal[iCnt] := arrSubjectCodes[iCnt];
    until length(arrLocal[iCnt]) > 0;
  end;
  for iCol := 1 to 5 do
  begin
    for iRow := 1 to 5 do
    begin
      repeat
        iRand := RandomRange(1, 8);
      until (arrTimeTable[iRow, iRand] = '');
      arrTimeTable[iRow, iRand] := arrLocal[iCol];
    end;
  end;
  display;
end;
```

```
procedure TfrmQuestion4.FormShow(Sender: TObject);
begin
  redQ4.Paragraph.TabCount := 9;
  redQ4.Paragraph.Tab[0] := 50;
  redQ4.Paragraph.Tab[1] := 100;
  redQ4.Paragraph.Tab[2] := 150;
  redQ4.Paragraph.Tab[3] := 200;
  redQ4.Paragraph.Tab[4] := 250;
  redQ4.Paragraph.Tab[5] := 300;
  redQ4.Paragraph.Tab[6] := 350;
  redQ4.Paragraph.Tab[7] := 400;
  redQ4.Paragraph.Tab[8] := 450;
  display;
  populate;
end;

procedure TfrmQuestion4.display;
var
  iRow, iCol, iCnt: integer;
  sLine: String;
begin
  sLine := #9;
  for iCnt := 1 to 7 do
    sLine := sLine + intToStr(iCnt) + #9;
  redQ4.Clear;
  redQ4.Lines.Add(sLine);
  for iRow := 1 to 5 do
  begin
    sLine := arrDays[iRow];
    for iCol := 1 to 7 do
    begin
      sLine := sLine + #9 + arrTimeTable[iRow, iCol];
    end;
    redQ4.Lines.Add(sLine);
  end;
end;

// ========================================================
// End of provided code
// ========================================================

end.
```