



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2023

PUNTE: 150

TYD: 3 uur

**Hierdie vraestel bestaan uit 25 bladsye, 2 bladsye met data en
2 bladsye vir beplanning.**

INSTRUKSIES EN INLIGTING

1. Hierdie vraestel is in VIER afdelings ingedeel. Kandidate moet AL die vrae in al VIER afdelings beantwoord.
2. Twee leë bladsye wat vir beplanningsdoeleindes gebruik kan word, is aan die einde van die vraestel ingesluit.
3. Die duur van hierdie eksamen is drie uur. As gevolg van die aard van hierdie eksamen is dit belangrik om daarop te let dat jy nie toegelaat sal word om die eksamenlokaal voor die einde van die eksamensessie te verlaat nie.
4. Hierdie vraestel is opgestel met programmeringsterme wat spesifiek is tot Delphi as programmeringstaal. Die Delphi-programmeringstaal moet gebruik word om die vrae te beantwoord.
5. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal volgens die voorgeskrewe vereistes toegeken word.
6. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, as die vraag nie datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
7. Jou programme moet op só 'n manier gekodeer word dat dit met enige data sal werk en nie net met die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel verskyn nie.
8. Roetines, soos soek, sorteer en seleksie, moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van die Delphi-programmeringstaal vir enige van hierdie roetines gebruik NIE.
9. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture verskaf word.
10. Jy moet jou werk gereeld stoor op die disket/CD/DVD/geheuestokkie wat aan jou gegee is of op die skyfspasie wat vir hierdie eksamensessie aan jou toegeken is.
11. Maak seker dat jou eksamennommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
12. Indien dit vereis word, druk die programmeringskode van al die programme/klasse wat jy voltooi het. Jou eksamennommer moet op alle drukstukke voorkom. Jy sal ná die eksamensessie 'n halfuur tyd vir drukwerk gegee word.
13. Aan die einde van hierdie eksamensessie moet jy 'n disket/CD/DVD/geheuestokkie inlewer met al jou werk daarop gestoor OF jy moet seker maak dat al jou werk op die skyfspasie gestoor is wat vir hierdie eksamensessie aan jou toegeken is. Maak seker dat al die lêers gelees kan word.

14. Die lêers wat jy benodig om hierdie vraestel te voltooi, is aan jou gegee op 'n disket/CD/DVD/geheuestokkie of op die skyfspasie wat aan jou toegeken is. Die lêers word in die vorm van wagwoordbeskernde uitvoerbare lêers verskaf.

Doen die volgende:

- Dubbelklik op die volgende uitvoerbare lêer wat met 'n wagwoord beskerm is:
DataNov2023.exe
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **%Learn4Life@**

Nadat dit onttrek ('extracted') is, sal die volgende lys lêers in die lêergids ('folder') **DataNov2023** beskikbaar wees:

Vraag 1/Question 1:

Details.txt
Question1_P.dpr
Question1_P.dproj
Question1_P.res
Question1_U.dfm
Question1_U.pas

Vraag 3/Question 3:

Question3_P.dpr
Question3_P.dproj
Question3_P.res
Question3_U.dfm
Question3_U.pas
School_U.pas

Vraag 2/Question 2:

ConnectDB_U.pas
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas
UniversityDB - Copy.mdb
UniversityDB.mdb

Vraag 4/Question 4:

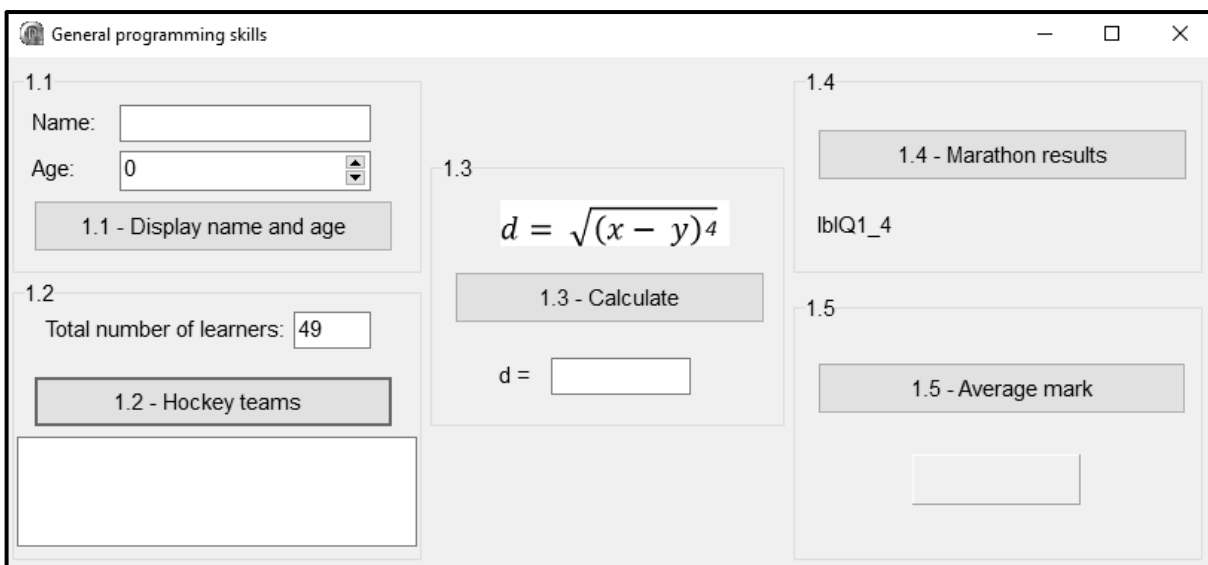
Question4_P.dpr
Question4_P.dproj
Question4_P.res
Question4_U.dfm
Question4_U.pas

AFDELING A**VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

Doen die volgende:

- Maak die onvolledige program in die **Question 1**-lêergids oop.
- Sleutel jou eksamenommer as kommentaar in die eerste reël van die **Question1_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK ('GUI')):



- Voltooi die kode vir elke afdeling van VRAAG 1, soos wat beskryf is in VRAAG 1.1 tot VRAAG 1.5 wat volg.

1.1 Knoppie [1.1 - Display name and age]

Die gebruiker moet die volgende doen:

- Sleutel 'n naam in die redigeerblokkie ('edit box') **edtQ1_1** in.
- Sleutel in/Selekteer die persoon se ouderdom in die spin-redigeerblokkie ('spin edit box') **spnQ1_1**.

Skryf kode om die volgende te doen:

- Verkry ('Extract') die naam wat ingesleutel is uit die redigeerblokkie **edtQ1_1** en stoor die naam in die veranderlike **sName** wat voorsien is.
- Verkry die ouderdom wat ingesleutel/geselekteer is uit die spin-redigeerblokkie **spnQ1_1** en stoor die ouderdom in die veranderlike **iAge** wat voorsien is.
- Vertoon die naam en ouderdom, die een onder die ander, met gebruik van 'n afvoerdialoogblokkie ('output dialogue box').

Voorbeeld van toevoer:

1.1

Name:

Age:

Voorbeeld van afvoer:

Question1_p

Martin
34

(5)

1.2 Knoppie [1.2 - Hockey teams]

Leerders wat daarin belangstel om hokkie te speel word in spanne van 11 spelers verdeel. 'n Totaal van 11 spelers verteenwoordig 'n volle span in hokkie. Die oorblywende spelers wat nie 'n volledige span kon vorm nie, word op 'n reserwelys geplaas.

'n Konstante veranderlike wat as `PLAYERS = 11` verklaar is, is voorsien.

Skryf kode om die volgende te doen:

- Verkry die totale getal leerders uit die redigeerblokkie **edtQ1_2**.
- Gebruik die totale getal leerders en die konstante `PLAYERS` om die volgende te bereken:
 - Getal hokkiespanne wat uit 11 spelers bestaan
 - Getal leerders op die reserwelys

1.2

Total number of learners:

Number of hockey teams: 4
Number of learners on reserve list: 5

(9)

1.3 Knoppie [1.3 - Calculate]

Die waarde van d moet bereken word deur die formule hieronder te gebruik:

$$d = \sqrt{(x - y)^4}$$

Kode is voorsien om waardes toe te ken aan die veranderlikes rX en rY wat die x - en y -waardes in die formule verteenwoordig.

Skryf kode om die volgende te doen:

- Gebruik die veranderlikes rX en rY wat voorsien is en geskikte wiskundige funksies om die waarde van d te bereken.
- Vertoon die waarde van d in die redigeerblokkie **edtQ1_3**, tot DRIE desimale plekke afgerond.

Voorbeeld van afvoer vir die waardes $rX = 12.46$ en $rY = 8.54$ wat voorsien is:

The screenshot shows a window with the title '1.3'. Inside the window, the formula $d = \sqrt{(x - y)^4}$ is displayed. Below the formula is a button labeled '1.3 - Calculate'. At the bottom of the window, the text 'd =' is followed by a text box containing the value '15.366'.

(5)

1.4 Knoppie [1.4 - Marathon results]

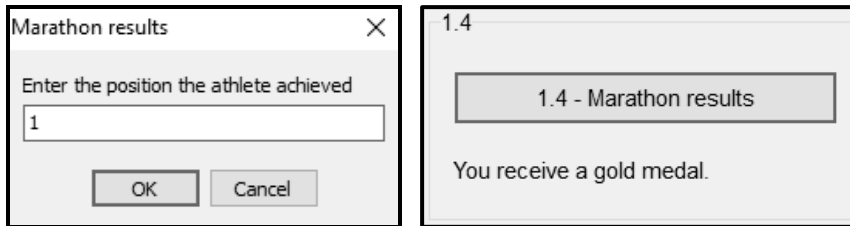
Deelnemers wat 'n marathon voltooi en in een van die top 20-posisies klaarmaak, sal 'n goue, silwer- of bronsmedalje ontvang, gebaseer op die posisie waarin hulle klaargemaak het. Al die ander deelnemers wat ná die 20^{ste} posisie klaarmaak, sal 'n deelname-sertifikaat ontvang.

Kode is vir die gebruiker voorsien met gebruik van 'n toevoerdialoogblokkie ('input dialogue box') om die posisie waarin die deelnemer klaargemaak het, in te sleutel. Die gebruikertoevoer is aan die veranderlike **iPosition** toegeken.

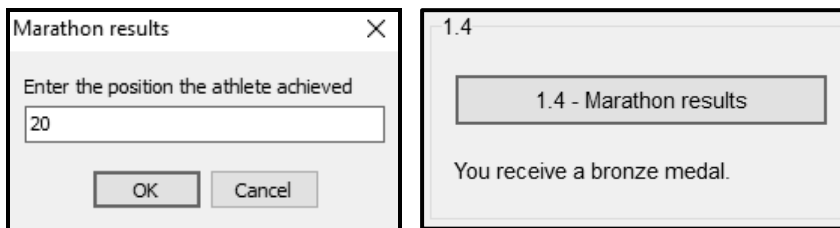
Skryf kode wat 'n gevalstelling ('case statement') gebruik om 'n geskikte boodskap in die etiket ('label') **lblQ1_4**, wat op die inligting in die tabel hieronder gebaseer is, te vertoon.

| Posisie klaargemaak | Toekenning |
|---------------------|----------------------|
| 1 | Goue medalje |
| 2 en 3 | Silwermedalje |
| 4 tot 20 | Bronsmedalje |
| Na 20 | Deelname-sertifikaat |

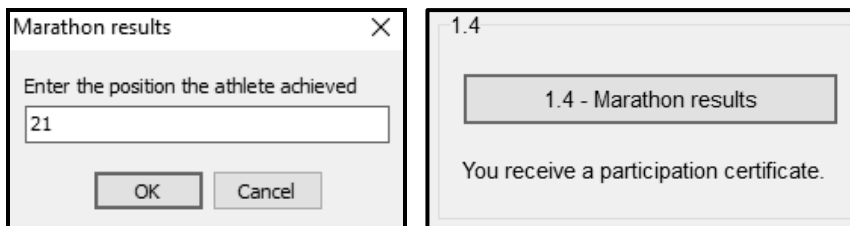
Voorbeeld van toevoer en afvoer indien 'n leerder die eerste plek behaal het:



Voorbeeld van toevoer en afvoer indien 'n leerder die 20^{ste} plek behaal het:



Voorbeeld van toevoer en afvoer indien 'n leerder die 21^{ste} plek behaal het:



(6)

1.5 Knoppie [1.5 - Average mark]

'n Tekslêër met die naam '**Details.txt**' bevat die name en punte van leerders in die volgende formaat:

```
<Naam>#<Punt>
```

Voorbeeld van die eerste vyf teksreëls in die tekslêër:

```
Erinn Stansell#87
Michael Dinjes#90
Gabrielle Wadhams#23
Mirelda Berendsen#47
Elmore Skalls#32
```

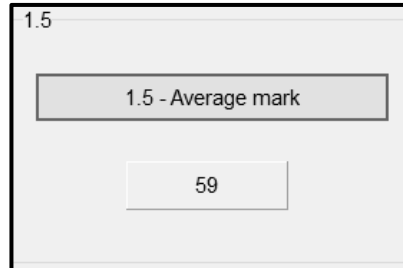
Skryf kode om die volgende te doen:

- Maak die tekslêër **Details.txt** oop, lees deur die teksreëls in die tekslêër en bepaal die gemiddelde punt van al die leerders.
- Vertoon die gemiddelde punt op die paneel **pnIQ1_5**, afgerond tot die naaste heelgetal.

LET WEL: Jou kode moet korrek werk vir enige getal teksreëls in die lêër.

WENK: Gebruik die posisie van die huts-karakter (#) om die punt vir elke leerder te onttrek.

Voorbeeld van afvoer:



(15)

- Sleutel jou eksamennummer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING A: 40

AFDELING B**VRAAG 2: DATABASISPROGRAMMERING**

Universiteite maak staat op doeltreffende administrasie en kommunikasie om die glatte verloop van hulle prosesse te fasiliteer.

'n Databasis met die naam **UniversityDB.mdb**, wat inligting bevat oor die verskillende dosente ('lecturers') en die kursusse wat hulle onderrig, is ontwikkel.

Die databasis bevat twee tabelle, **tblLecturers** en **tblCourses**.

LET WEL: Die bladsye met data aan die einde van die vraestel verskaf inligting oor die ontwerp en inhoud van die databasis.

Doen die volgende:

- Maak die onvolledige projeklêer met die naam **Question2_P.dpr** in die **Question 2-lêergids** oop.
- Sleutel jou eksamennommer as kommentaar in die eerste reël van die **Question2_U.pas**-eenheidlêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie. Die inhoud van die tabelle word vertoon, soos wat hieronder getoon word oor die seleksie van oortjieblad ('tab sheet') **2.2 - Delphi code**.

Database programming

2.1 - SQL | 2.2 - Delphi code

| LecturerID | LecturerName | LecturerSurname | Gender | Multilingual |
|------------|--------------|-----------------|--------|--------------|
| DG022 | Dean | Gillian | Male | False |
| FK681 | Frances | Kobokoane | Female | True |
| MP014 | Marie | du Plessis | Female | True |
| NM612 | Ntswaki | Mokoena | Female | False |

| CourseID | CourseName | Duration | MaxStudents | OnlineOption | LecturerID |
|----------|----------------------------|----------|-------------|--------------|------------|
| ADVOFF | Spreadsheets and Databases | 6 | 150 | True | NM612 |
| AIAPP | AI Application Development | 8 | 50 | False | TT663 |
| AIFUN | AI Fundamentals | 6 | 50 | True | TT663 |
| AJAX | AJAX Development | 10 | 100 | True | SC884 |
| APPSTAT | Applied Statistics | 8 | 50 | False | TT663 |
| BASPHY | Basic Python Programming | 5 | 50 | True | SC884 |
| BIOCEL | Science of Materials | 8 | 100 | False | MP014 |

2.2.1

2.2.2

2.2.1 - Average duration of courses

2.2.2 - Register new lecturer

Restore database | Close

- Volg die instruksies hieronder om die kode vir elke afdeling, soos wat in VRAAG 2.1 en VRAAG 2.2 beskryf word, te voltooi.
- Gebruik SQL-stellings om VRAAG 2.1 te beantwoord en Delphi-kode om VRAAG 2.2 te beantwoord.

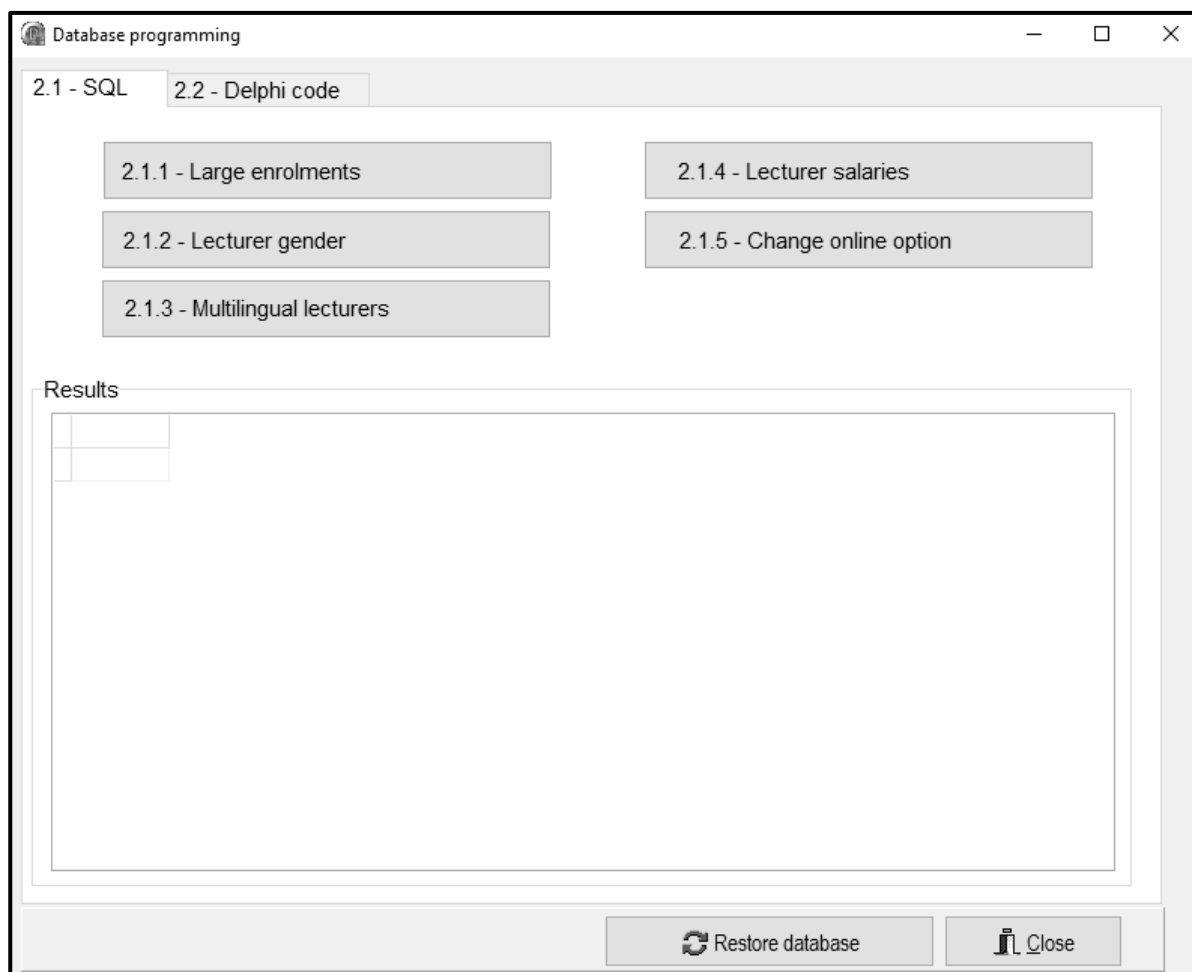
LET WEL:

- Die 'Restore database'-knoppie word voorsien om die data wat in die databasis voorkom, na die oorspronklike inhoud te herstel.
- Die inhoud van die databasis is met 'n wagwoord beskerm, m.a.w. jy sal NIE in staat wees om direkte toegang tot die inhoud van die databasis te verkry deur Microsoft Access te gebruik NIE.
- Kode word voorsien om die GGK('GUI')-komponente met die databasis te koppel. MOENIE enige van die kode wat voorsien word, verander NIE.
- TWEE veranderlikes word as publieke veranderlikes verklaar, soos in die tabel hieronder beskryf word.

| Veranderlike | Datatype | Beskrywing |
|--------------|-----------|---|
| tblLecturers | TADOTable | Verwys na die tabel tblLecturers |
| tblCourses | TADOTable | Verwys na die tabel tblCourses |

2.1 Oortjieblad ('Tab sheet') [2.1 - SQL]

Voorbeeld van grafiese gebruikerskoppelvlak (GGK) vir VRAAG 2.1:



LET WEL:

- Gebruik SLEGS SQL-stellings om VRAAG 2.1.1 tot VRAAG 2.1.5 te beantwoord.
- Kode om die SQL-stellings uit te voer en die resultate van die navrae te vertoon, is voorsien. Die SQL-stellings wat aan die veranderlikes **sSQL1**, **sSQL2**, **sSQL3**, **sSQL4** en **sSQL5** toegeken is, is onvolledig.

Voltooi die SQL-stellings om die take wat in VRAAG 2.1.1 tot VRAAG 2.1.5 hieronder beskryf word, uit te voer.

2.1.1 Knoppie [2.1.1 - Large enrolments]

Vertoon die besonderhede van al die kursusse in die **tblCourses**-tabel, wat 100 of meer studente kan akkommodeer.

Voorbeeld van afvoer van die eerste vyf rekords:

| CourseID | CourseName | Duration | MaxStudents | OnlineOption | LecturerID |
|----------|--------------------------------|----------|-------------|--------------|------------|
| DATALG | Data Structures and Algorithms | 5 | 100 | False | MP014 |
| OFFICE | Intro to Office Applications | 5 | 150 | True | NM612 |
| ADVOFF | Spreadsheets and Databases | 6 | 150 | True | NM612 |
| AJAX | AJAX Development | 10 | 100 | True | SC884 |
| PYTPRO | Advanced Python Programming | 10 | 100 | True | ST046 |

(3)

2.1.2 Knoppie [2.1.2 - Lecturer gender]

Vertoon die **LecturerName**, **LecturerSurname** en die eerste letter van die geslag van al die dosente. Vertoon die geslag deur 'n kolomopskrif met die naam **Gender (M/F)** te gebruik.

Voorbeeld van afvoer van die eerste vyf rekords:

| LecturerName | LecturerSurname | Gender (M/F) |
|--------------|-----------------|--------------|
| Thabiso | Tau | M |
| William | Dibiase | M |
| Susan | Tokoane | F |
| Dean | Gillian | M |
| Steven | Conradie | M |

(4)

2.1.3 Knoppie [2.1.3 - Multilingual lecturers]

Vertoon die **CourseID** en **CourseName** van al die kursusse wat 'n meertalige ('multilingual') dosent het. Die resultate moet alfabeties volgens die kursusnaam gesorteer wees.

Voorbeeld van afvoer van die eerste vyf rekords:

| CourseID | CourseName |
|----------|--------------------------------|
| AJAX | AJAX Development |
| DELPHI | Basic Delphi Programming |
| BASPHY | Basic Python Programming |
| SOLID | Basics of Solid Works |
| DATALG | Data Structures and Algorithms |

(6)

2.1.4 Knoppie [2.1.4 - Lecturer salaries]

'n Dosent se salaris word bepaal deur die getal kursusse wat deur daardie dosent gefasiliteer word. 'n Bedrag van R10 000 word betaal vir elke kursus wat deur die dosent gefasiliteer word. Byvoorbeeld, as 'n dosent drie kursusse fasiliteer, sal die dosent 'n totale salaris van R30 000 ontvang.

Vertoon die **LecturerID** en die totale salaris in 'n nuwe kolom met die opskrif **Salary**, as geldeenheid ('currency') geformateer.

Voorbeeld van afvoer van die eerste vyf rekords:

| LecturerID | Salary |
|------------|------------|
| DG022 | R20 000.00 |
| FK681 | R10 000.00 |
| MP014 | R20 000.00 |
| NM612 | R30 000.00 |
| RW111 | R30 000.00 |

(5)

2.1.5 Knoppie [2.1.5 - Change online option]

Skryf kode om die **OnlineOption**-veld na 'false' te verander as die **CourseName**-veld die woord 'Programming' bevat.

Kode is voorsien om 'n boodskap te vertoon wat aandui dat die inhoud van die databasis verander het.

Voorbeeld van die eerste ses rekords van die **tblCourses**-tabel voordat die aanlyn opsie verander is:

| CourseID | CourseName | Duration | MaxStudents | OnlineOption | LecturerID |
|----------|--------------------------------|----------|-------------|--------------|------------|
| DATALG | Data Structures and Algorithms | 5 | 100 | False | MP014 |
| OFFICE | Intro to Office Applications | 5 | 150 | True | NM612 |
| ADVOFF | Spreadsheets and Databases | 6 | 150 | True | NM612 |
| AJAX | AJAX Development | 10 | 100 | True | SC884 |
| PYTPRO | Advanced Python Programming | 10 | 100 | True | ST046 |
| OSFEAT | Embedded Linux Development | 8 | 150 | False | RW111 |

Voorbeeld van die eerste ses rekords van die **tblCourses**-tabel nadat die aanlyn opsie vir al die programmeringskursusse na 'false' verander is:

| CourseID | CourseName | Duration | MaxStudents | OnlineOption | LecturerID |
|----------|--------------------------------|----------|-------------|--------------|------------|
| DATALG | Data Structures and Algorithms | 5 | 100 | False | MP014 |
| OFFICE | Intro to Office Applications | 5 | 150 | True | NM612 |
| ADVOFF | Spreadsheets and Databases | 6 | 150 | True | NM612 |
| AJAX | AJAX Development | 10 | 100 | True | SC884 |
| PYTPRO | Advanced Python Programming | 10 | 100 | False | ST046 |
| OSFEAT | Embedded Linux Development | 8 | 150 | False | RW111 |

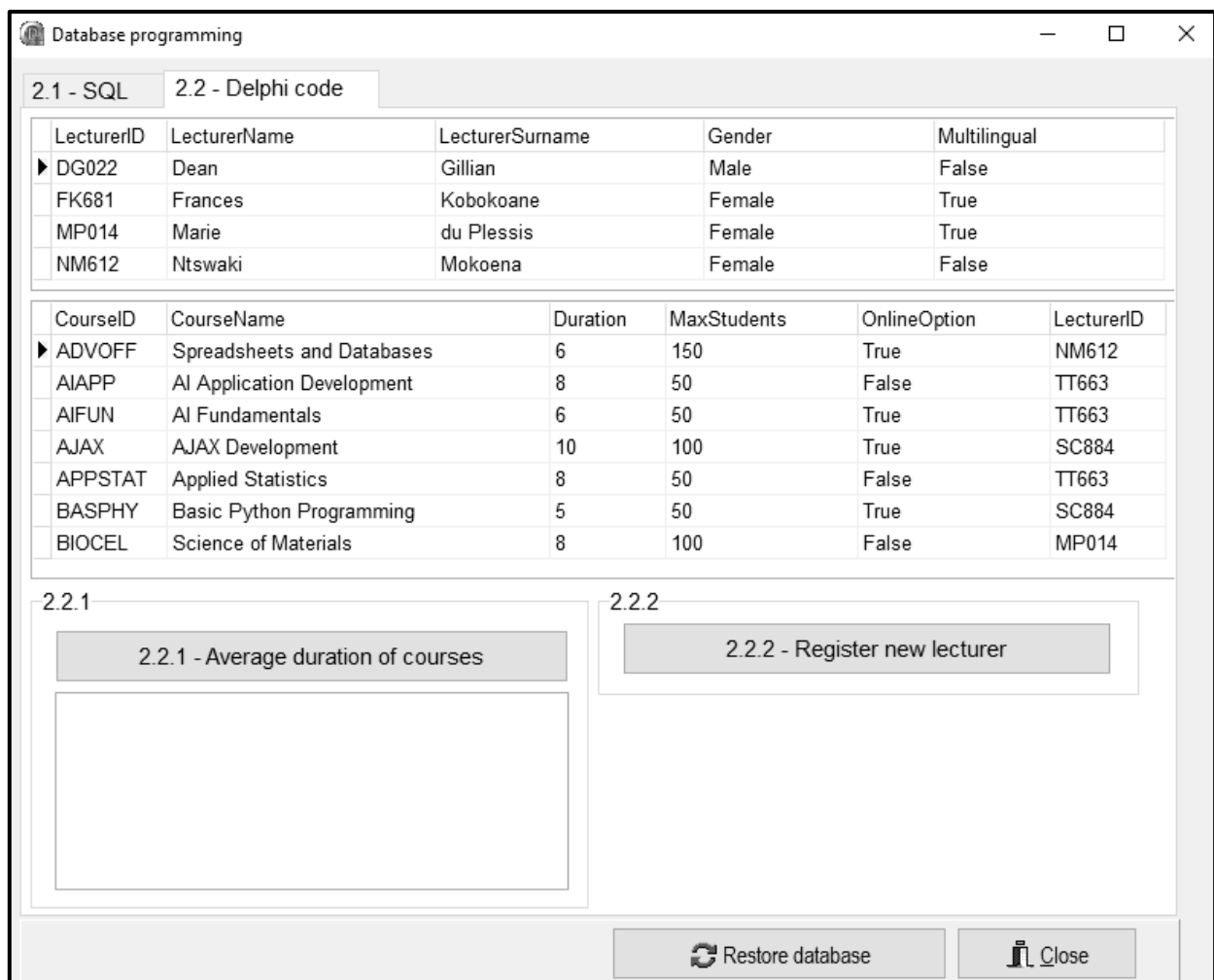
(4)

2.2 Oortjieblad ('Tab sheet') [2.2 - Delphi code]

LET WEL:

- Gebruik SLEGS Delphi-programmeringskode om VRAAG 2.2.1 en VRAAG 2.2.2 te beantwoord.
- GEEN punte sal vir SQL-stellings in VRAAG 2.2 toegeken word nie.

Voorbeeld van grafiese gebruikerskoppelvlak (GGK ('GUI')) vir VRAAG 2.2:



2.2.1 Knoppie [2.2.1 - Average duration of courses]

Die werkslading van al die dosente moet bepaal word sodat 'n regverdig aantal kursusse aan hulle toegewys kan word om te fasiliteer.

Gebruik die **redQ2_2_1**-komponent om die afvoer te vertoon.

Kode is voorsien om die **redQ2_2_1**-komponent leeg te maak.

Skryf kode om die volgende te doen:

- Vertoon die **LecturerID**, **LecturerName** en **LecturerSurname** as 'n opskrif in die volgende formaat:
<LecturerID>: <LecturerName> <LecturerSurname>
- Vertoon 'n genommerde lys met kursusname wat deur elke dosent gefasiliteer word.
- Bereken en vertoon die gemiddelde duur van die kursusse wat deur elke dosent gefasiliteer word, tot TWEE desimale plekke geformateer.

Voorbeeld van afvoer vir die eerste twee dosente:

```
DG022: Dean Gillian
1. Intro to Computer Literacy
2. Fundamentals of Physics
Average duration of courses: 7.50

FK681: Frances Kobokoane
1. Basics of Solid Works
Average duration of courses: 6.00
```

(14)

2.2.2 Knoppie [2.2.2 - Register new lecturer]

Wanneer 'n nuwe dosent aangestel word, moet hulle besonderhede in die databasis bygevoeg word.

Skryf kode om 'n nuwe rekord in die **tblLecturers**-tabel by te voeg deur die volgende besonderhede te gebruik:

- LecturerID - ZT032
- LecturerName - Zander
- LecturerSurname - Thomas
- Gender - Male
- Multilingual - True

Voorbeeld van rekords in die **tblLecturers**-tabel wat aandui dat die rekord suksesvol in die tabel bygevoeg is:

| LecturerID | LecturerName | LecturerSurname | Gender | Multilingual |
|------------|--------------|-----------------|--------|--------------|
| TJ225 | Trevor | Jones | Male | False |
| TT663 | Thabiso | Tau | Male | False |
| WD010 | William | Dibiase | Male | True |
| ZT032 | Zander | Thomas | Male | True |

(4)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING B: 40

AFDELING C**VRAAG 3: OBJEK-GEORIËNTEERDE PROGRAMMERING**

Die plaaslike skole-distrikskantoor vereis 'n verifikasie van skole in die distrik en 'n verslag met die uitslae van die skole om die bedrag vir befondsing wat elke skool sal ontvang, te bepaal.

Doen die volgende:

- Maak die onvolledige program in die **Question 3**-lêergids oop.
- Maak die onvolledige objekklas **School_U.pas** oop.
- Sleutel jou eksamennummer as kommentaar in die eerste reël van beide die **Question3_U.pas**-lêer en die **School_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit.

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK):

The screenshot shows a window titled "Object-orientated programming" with a subtitle "School rating and funding". The interface is divided into three main sections:

- 3.2.1**: Contains input fields for "School name" (filled with "Alan Turing High School"), "Total number of learners enrolled" (filled with "1250"), and a checked checkbox for "Public school". Below these is a button labeled "3.2.1 – Instantiate object".
- 3.2.2**: Contains a spinner control for "Number of learners who passed" (set to "1183") and a button labeled "3.2.2 – Rating".
- 3.2.3**: Contains a button labeled "3.2.3 – Funding".

At the bottom right of the window, there is a small image of a stack of books.

- Voltooi die kode soos gespesifiseer in VRAAG 3.1 en VRAAG 3.2 wat volg.

- 3.1 Die onvoltooide objekklas (**TSchool**) wat voorsien is, bevat die verklaring van vier attribute wat 'n **School**-objek beskryf.

Die attribute vir 'n **School**-objek is soos volg verklaar:

| Attribuut | Datatype | Beskrywing |
|----------------|----------|---|
| fSchoolName | String | Die naam van die skool |
| fTotalLearners | Integer | Die totale getal leerders wat by die skool ingeskryf is |
| fPublicSchool | Boolean | 'True' as dit 'n openbare skool is, anders 'false' |
| fRating | Char | 'n Gradering (A, B, C of Z) wat aan elke skool op grond van die slaagsyferpersentasie van die skool toegeken word: A – 80% of hoër B – Van 60% tot 79% C – Laer as 60% Z – Gradering nie toegeken nie |

'n Onvolledige **konstruktor**-metode is voorsien.

LET WEL: Jy mag NIE enige addisionele attribute of gebruikergedefinieerde metodes byvoeg NIE, tensy jy uitdruklik in een van die vrae aangesê word om dit te doen.

Voltooi die kode in die objekklas soos wat in VRAAG 3.1.1 tot VRAAG 3.1.5 hieronder beskryf word.

- 3.1.1 **Constructor Create** is van drie parameters voorsien.

Skryf kode om die konstruktormetode soos volg te voltooi:

- Ken die waardes in die parameterlys aan die ooreenstemmende attribute, **fSchoolName**, **fTotalLearners** en **fPublicSchool**, toe.
- Ken 'n verstekgradering van 'Z' aan die **fRating**-attribuut toe. (3)

- 3.1.2 Skryf 'n toegangsmetode ('accessor method') met die naam **getPublicSchool** vir die **fPublicSchool**-attribuut. (2)

- 3.1.3 Die gradering van die skool sal elke jaar opgedateer word op grond van die slaagsyferpersentasie van die skool. Die slaagsyferpersentasie word bereken deur die totale getal leerders wat deurgekom het en die totale getal leerders in die skool te gebruik.

Die inligting in die tabel hieronder word gebruik om die skool se gradering te bepaal:

| Gradering | Slaagsyferpersentasie |
|-----------|-----------------------|
| A | 80% of hoër |
| B | Van 60% tot 79% |
| C | Laer as 60% |

Skryf kode vir 'n metode met die naam **updateRating** om die volgende te doen:

- Ontvang die totale getal leerders wat geslaag het as 'n parameterwaarde.
- Gebruik die parameterwaarde en die **fTotalLearners**-attribuut om die skool se slaagsyferpersentasie te bereken.
- Gebruik die tabel wat voorsien is om die waarde van die **fRating**-attribuut te bepaal en te stel.

(8)

3.1.4 'n Bedrag van R145,50 is toegeken per leerder wat by die skool ingeskryf is.

Skryf kode vir 'n metode met die naam **calcFunding** om die totale bedrag wat die skool vir befondsing sal ontvang, te bereken en terug te stuur.

Die befondsing word bereken deur die volgende formule te gebruik:

$$\text{befondsing} = \text{leerders ingeskryf} \times 145.50$$

(4)

3.1.5 Skryf kode vir 'n **toString**-metode om 'n string terug te stuur wat die objek beskryf. Die Boolese attribuut **fPublicSchool** moet gebruik word om te bepaal of die frase 'Public school' of 'Private school' by die string gevoeg moet word.

Die formaat van die string wat teruggestuur moet word, word hieronder getoon.

```
<fSchoolName>
-----
Total number of learners: < fTotalLearners >
Rating: <fRating>
Public school/Private school
```

Voorbeeld:

```
Alan Turing High School
-----
Total number of learners: 1250
Rating: Z
Public school
```

(7)

3.2 'n Onvolledige program is in die **Question 3**-lêergids voorsien. Die program bevat kode vir die objekklas om toeganklik te wees en verklaar 'n objekveranderlike met die naam **objSchool**.

Skryf kode om die take uit te voer wat beskryf word in VRAAG 3.2.1 tot VRAAG 3.2.3 wat volg.

3.2.1 Knoppie [3.2.1 – Instantiate object]

Die gebruiker moet die skool se naam in die redigeerblokkie ('edit box') **edtQ3_2_1** insleutel, die totale getal leerders in die spin-redigeerblokkie **spnQ3_2_1** selekteer en die kontroleblokkie ('check box') **chbQ3_2_1** merk indien die skool 'n openbare skool is.

Skryf kode om die volgende te doen:

- Verkry die skool se naam uit die redigeerblokkie **edtQ3_2_1**, die totale getal leerders in die spin-redigeerblokkie **spnQ3_2_1** en bepaal of die kontroleblokkie **chbQ3_2_1** gemerk is of nie.
- Gebruik die inligting wat verkry is om 'n nuwe **School**-objek te instansieer.
- Gebruik die **toString**-metode om die inligting van die **School**-objek in die 'rich edit' **redQ3** te vertoon.

Voorbeeld van toevoer en afvoer vir 'n openbare skool:

3.2.1

School name:

Total number of learners enrolled:

Public school:

3.2.1 – Instantiate object

```
Alan Turing High School
-----
Total number of learners: 1250
Rating: Z
Public school
```

Voorbeeld van toevoer en afvoer vir 'n privaat skool:

3.2.1

School name:

Total number of learners enrolled:

Public school:

3.2.1 – Instantiate object

```
Steve Jobs Academy
-----
Total number of learners: 820
Rating: Z
Private school
```

(7)

3.2.2 Knoppie [3.2.2 – Rating]

Die gradering van 'n skool word deur die slaagsyferpersentasie van die skool bepaal.

Die spin-redigeerblokkie **spnQ3_2_2** moet gebruik word om die totale getal leerders wat geslaag het, in te sleutel.

Skryf kode om die volgende te doen:

- Roep die **updateRating**-metode deur die waarde wat uit die spin-redigeerblokkie **spnQ3_2_2** verkry is, as 'n argument te gebruik.
- Roep die **toString**-metode om die opgedateerde inligting van die **School**-objek in die rich edit **redQ3** te vertoon.

Voorbeeld van afvoer as die totale getal leerders 1 250 is en die totale getal leerders wat geslaag het, 1 183 is:

```
Alan Turing High School
-----
Total number of learners: 1250
Rating: A
Public school
```

Voorbeeld van afvoer as die totale getal leerders 1 250 is en die totale getal leerders wat geslaag het, 999 is:

```
Alan Turing High School
-----
Total number of learners: 1250
Rating: B
Public school
```

Voorbeeld van afvoer as die totale getal leerders 1 250 is en die totale getal leerders wat geslaag het, 500 is:

```
Alan Turing High School
-----
Total number of learners: 1250
Rating: C
Public school
```

(4)

3.2.3 Knoppie [3.2.3 – Funding]

Befondsing sal slegs vir openbare skole beskikbaar wees. Gebruik die 'rich edit' **redQ3** om die afvoer te vertoon.

Skryf kode om die volgende te doen:

Gebruik die relevante metode om te toets of die skool 'n openbare skool is.

- As die skool 'n openbare skool is, roep die **calcFunding**-metode om die befondsingsbedrag wat die skool sal ontvang, te vertoon, as geldeenheid ('currency') geformateer, met 'n boodskap soos wat op die volgende bladsy aangedui word.
- As die skool nie 'n openbare skool is nie, vertoon 'n geskikte boodskap wat aandui dat geen fondse beskikbaar is nie.

Voorbeeld van afvoer as die skool 'n openbare skool is en 'n totale getal leerders van 1 250 het:

Public school will receive R181 875.00

Voorbeeld van afvoer as die skool NIE 'n openbare skool is NIE:

No funding available

(5)

- Sleutel jou eksamenommer as kommentaar in die eerste reël van die objekklas en die vormklas in.
- Stoor jou program.
- Druk die kode in die objekklas en die vormklas indien dit vereis word.

TOTAAL AFDELING C: 40

AFDELING D

VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING

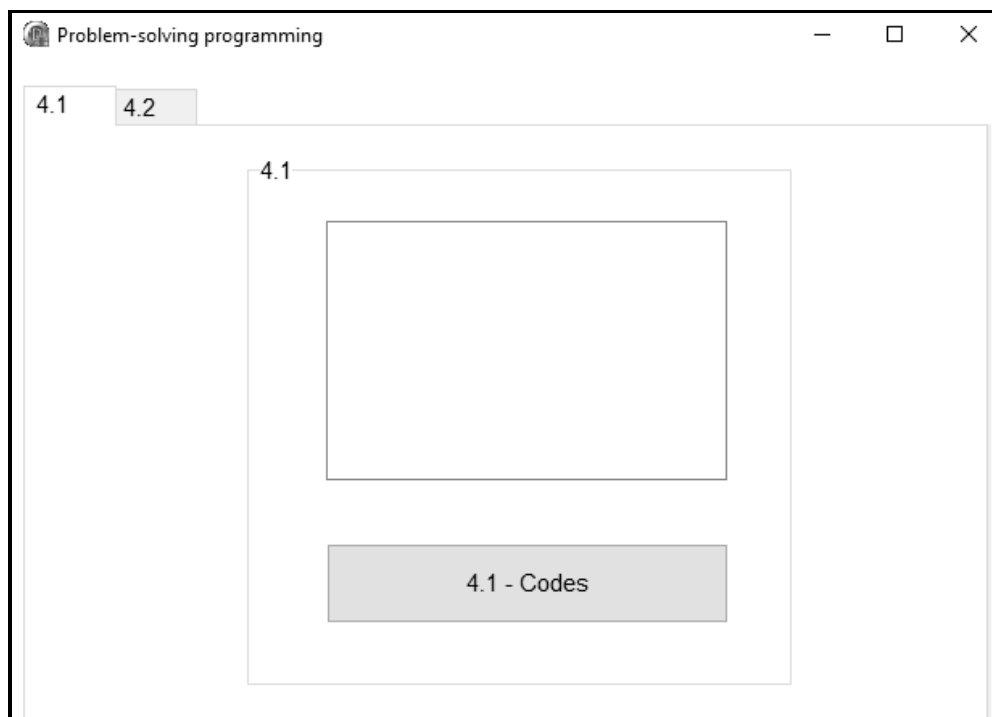
Doen die volgende:

- Maak die onvolledige program in die **Question 4**-lêergids oop.
- Sleutel jou eksamenommer as kommentaar in die eerste reël van die **Question4_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.
- Twee oortjieblaie ('tab sheets') met die name **tshQ4_1** en **tshQ4_2** is op die grafiese gebruikerskoppelvlak (GGK/'GUI') voorsien.

Voltooi die kode vir elke afdeling van VRAAG 4, soos in VRAAG 4.1 en VRAAG 4.2 hieronder beskryf word.

4.1 Oortjieblad ('Tab sheet') [4.1]

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK) vir oortjieblad ('tab sheet') 4.1:



Jy is voorsien van 'n skikking met die naam **arrCodes** wat vyf kodes bevat. Elke kode bevat letters, syfers en spesiale karakters.

```
arrCodes: array [1 .. 5] of String =
```

```
( 'An7J*Q#D&N', 'pL78K#$.%BV', '89@FGh0&Y56#$Q', 'Bn4m321&*#T',  
'P2QwER%$$a');
```

'n Spesiale karakter verwys na 'n karakter wat nie 'n letter of 'n syfer is nie.

Knoppie [4.1 - Codes]

Skryf kode om die volgende te doen:

Verwyder al die spesiale karakters uit elke kode in die skikking om 'n nuwe kode te skep. Tel hoeveel spesiale karakters uit elke kode verwyder is.

Voeg die nuwe kode en die getal spesiale karakters wat daaruit verwyder is, by die lysblokkie ('list box') **IstQ4_1** in die volgende formaat:

<Nuwe kode>(getal spesiale karakters wat verwyder is)

LET WEL: Jou kode moet vir enige stel data in die skikking werk.

Voorbeeld van afvoer vir die kodes wat tans in die skikking is:



(12)

4.2 Oortjieblad ('Tab sheet') [4.2]

'n Skoolrooster het vyf dae en sewe periodes per dag.

Jy is van die volgende verklarings voorsien:

```
arrDays: array [1 .. 5] of String =
    ('Mon.', 'Tue.', 'Wed.', 'Thu.', 'Fri.');
```

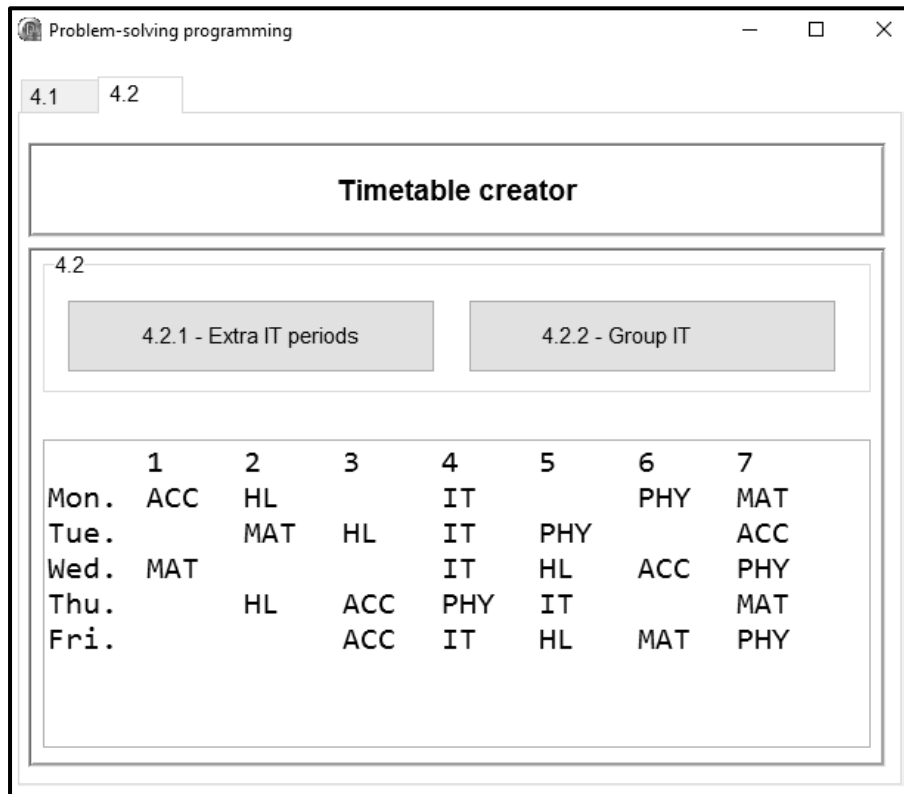
```
arrTimeTable: array [1 .. 5, 1 .. 7] of String;
```

```
arrSubjectCodes: array [1 .. 5] of String =
    ('IT', 'HL', 'ACC', 'PHY', 'MAT');
```

LET WEL: Kode is voorsien om die volgende te doen:

- Vul die **arrTimeTable**-skikking deur die vyf vakkodes IT, HL, ACC, PHY en MAT ewekansig in die skikking te plaas, een keer per dag vir die vyf dae. Daar sal elke dag twee vry periodes (leë ruimtes) wees.
- Vertoon die sewe periodes per dag vir elk van die vyf dae op die rooster (Maandag tot Vrydag).

Voorbeeld van die grafiese gebruikerskoppelvlak (GGK) vir oortjieblad ('tab sheet') 4.2:



4.2.1 Knoppie [4.2.1 - Extra IT periods]

Die skoolhoof het besluit om vier ekstra periodes, een keer per dag (Maandag tot Vrydag), vir die 'IT'-vakkode op die rooster by te voeg.

Die ekstra IT-periodes moet van Maandag tot Donderdag in die **arrTimeTable**-skikking in die **eerste beskikbare periode** geplaas word deur die kode 'IT' te gebruik.

Voorbeeld van afvoer nadat die ekstra IT-periodes in die eerste beskikbare periodes van Maandag tot Donderdag geplaas is:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------|-----|-----|-----|-----|-----|-----|-----|
| Mon. | ACC | HL | IT | IT | | PHY | MAT |
| Tue. | IT | MAT | HL | IT | PHY | | ACC |
| Wed. | MAT | IT | | IT | HL | ACC | PHY |
| Thu. | IT | HL | ACC | PHY | IT | | MAT |
| Fri. | | | ACC | IT | HL | MAT | PHY |

LET WEL: Daar sal by verstek ses vry periodes op die rooster beskikbaar wees nadat al die vakkodes geplaas is, die ekstra IT-periodes ingesluit.

As gevolg van die ewekansige plasing van vakkodes mag jou afvoer van die voorbeeld hierbo verskil.

(6)

4.2.2 Knoppie [4.2.2 - Group IT]

Die IT-onderwyser het versoek dat beide IT-periodes van Maandag tot Donderdag, opeenvolgend (die een na die ander een) geplaas moet word.

Skryf kode om die volgende te doen:

Herrangskik die periodes vir elke dag om die twee IT-periodes langs mekaar te plaas. Die eerste IT-periode vir die dag moet in sy plek bly, terwyl die tweede IT-periode met 'n ander periode omgeruil moet word om langs die eerste IT-periode geplaas te kan word.

Voorbeeld van afvoer nadat die IT-periodes saamgegroepeer is:

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|------|-----|----|-----|-----|-----|-----|-----|
| Mon. | ACC | HL | IT | IT | | PHY | MAT |
| Tue. | IT | IT | HL | MAT | PHY | | ACC |
| Wed. | MAT | IT | IT | | HL | ACC | PHY |
| Thu. | IT | IT | ACC | PHY | HL | | MAT |
| Fri. | | | ACC | IT | HL | MAT | PHY |

(12)

- Sleutel jou eksamenommer as kommentaar in die eerste reël van die programlêer in.
- Stoor jou program.
- Druk die kode indien dit vereis word.

TOTAAL AFDELING D:
GROOTTOTAAL:

30
150

INLIGTINGSTEGNOLOGIE V1**DATABASIS-INLIGTING VIR VRAAG 2:**

Die ontwerp van die databasistabelle vir die databasis **UniversityDB** is soos volg:

Tabel: **tblLecturers**

Hierdie tabel bevat die besonderhede van die dosente.

| Veldnaam | Datatype | Beskrywing |
|-----------------|-----------------|---|
| LecturerID | Text (5) | 'n Unieke ID wat gebruik word om 'n dosent te identifiseer |
| LecturerName | Text (25) | Die naam van die dosent |
| LecturerSurname | Text (25) | Die van van die dosent |
| Gender | Text (6) | Die geslag van die dosent |
| Multilingual | Boolean | 'n Veld wat aandui of 'n dosent 'n kursus in meer as een taal kan aanbied |

Voorbeeld van die rekords van die **tblLecturers**-tabel:

| LecturerID | LecturerName | LecturerSurname | Gender | Multilingual |
|------------|--------------|-----------------|--------|--------------|
| DG022 | Dean | Gillian | Male | False |
| FK681 | Frances | Kobokoane | Female | True |
| MP014 | Marie | du Plessis | Female | True |
| NM612 | Ntswaki | Mokoena | Female | False |
| RW111 | Richard | Wright | Male | True |
| SC884 | Steven | Conradie | Male | True |
| ST046 | Susan | Tokoane | Female | False |
| TJ225 | Trevor | Jones | Male | False |
| TT663 | Thabiso | Tau | Male | False |
| WD010 | William | Dibiase | Male | True |

Tabel: **tblCourses**

Hierdie tabel bevat inligting oor die kursusse wat by die universiteit aangebied word.

| Veldnaam | Datatype | Beskrywing |
|-----------------|-----------------|--|
| CourseID | Text (10) | 'n Unieke ID wat gebruik word om 'n kursus te identifiseer |
| CourseName | Text (35) | Die naam van die kursus |
| Duration | Number | Die duur van die kursus (in weke) |
| MaxStudents | Number | Die maksimum getal studente wat in die kursus toegelaat word |
| OnlineOption | Boolean | 'n Veld wat aandui of 'n aanlyn opsie vir studente beskikbaar is |
| LecturerID | Text (5) | Die ID van die dosent wat die kursus aanbied |

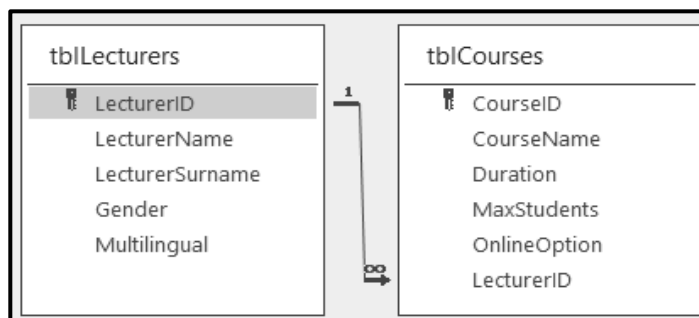
Voorbeeld van die eerste tien rekords van die **tblCourses**-tabel:

| CourseID | CourseName | Duration | MaxStudents | OnlineOption | LecturerID |
|----------|--------------------------------|----------|-------------|--------------|------------|
| ADVOFF | Spreadsheets and Databases | 6 | 150 | True | NM612 |
| AIAPP | AI Application Development | 8 | 50 | False | TT663 |
| AIFUN | AI Fundamentals | 6 | 50 | True | TT663 |
| AJAX | AJAX Development | 10 | 100 | True | SC884 |
| APPSTAT | Applied Statistics | 8 | 50 | False | TT663 |
| BASPHY | Basic Python Programming | 5 | 50 | True | SC884 |
| BIOCEL | Science of Materials | 8 | 100 | False | MP014 |
| COMLIT | Intro to Computer Literacy | 5 | 150 | True | DG022 |
| DATALG | Data Structures and Algorithms | 5 | 100 | False | MP014 |
| DATANL | Study of Data Analytics | 5 | 75 | False | TJ225 |

LET WEL:

- Konneksiekode is voorsien.
- Die databasis word met 'n wagwoord beskerm, dus sal jy nie direkte toegang tot die databasis kan verkry nie.

Die volgende een-tot-baie-verwantskap met referensiële integriteit ('referential integrity') bestaan tussen die twee tabelle in die databasis:



BEPLANNINGSBLADSY 1

BEPLANNINGSBLADSY 2



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 | <p>Knoppie [1.1 – Display name and age]</p> <p>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 ✓ met #13/#10/sLineBreak ✓ vir volgende reël die ouderdom, omgeskakel na 'n string ✓</p> | 5 | |
| 1.2 | <p>Knoppie [1.2 – Hockey teams]</p> <p>Onttrek die getal leerders uit die edit box, ✓ omgeskakel na 'n heelgetal/reële getal ✓ Bereken die aantal spanne Getal leerders DIV ✓ PLAYERS ✓ (gebruik konstante) Bereken die aantal reserwes Getal leerders MOD ✓ PLAYERS ✓ Vertoon die aantal spanne omgeskakel na String ✓ in die memoQ1_2 ✓ Vertoon ook die aantal reserwes ✓</p> <p>AANVAAR OOK: Alternatiewe wiskundige funksies wat die korrekte antwoord sal gee.</p> <p>LET WEL: Die gegewe konstante PLAYERS moet ten minste eenmaal gebruik word</p> | 9 | |
| 1.3 | <p>Knoppie [1.3 – Calculate]</p> <p>Formule: $d = \text{Sqrt} \left(\text{power} \left((rX - rY), 4 \right) \right)$ ✓ Vertoon die waarde van d in edtQ1_3 ✓ Geformateer met 3 desimale plekke ✓</p> <p>AANVAAR OOK: Alternatiewe wiskundige funksies wat die korrekte antwoord sal gee.</p> <p>MOENIE harde kodering aanvaar i.p.v. wiskundige funksies NIE.</p> | 5 | |

| | | | |
|---------------------------|---|-----------|--|
| 1.4 | <p>Knoppie [1.4 – Marathon results]</p> <p>case iPosisie of ✓ 1: ✓ Vertoon 'You receive a gold medal' ✓ 2, 3: Vertoon 'You receive a silver medal' ✓ 4..20: Vertoon 'You receive a bronze medal' ✓ Else Vertoon 'You receive a participation certificate' ✓ End // case</p> <p>NOTA: Die eerste twee punte is vir die struktuur van 'n gevalstelling, sal verbeur word wanneer veelvuldige keuse stelling gebruik word.</p> | 6 | |
| 1.5 | <p>Knoppie [1.5 – Average mark]</p> <p>Verklaar lêerveranderlike ✓ Inisialiseer Totaal en iTel na 0 ✓ AssignFile (tFile, 'Details.txt') ✓ Reset (tFile) ✓ Terwyl nie einde van lêer nie ✓ Begin Lees reel uit teksleer ✓ Inkrementeer iTel ✓ Vind die posisie van die # skeiteken ✓ Onttrek punt uit reel ✓ deur regte indekse te gebruik ✓ Skakel punt om na heel-/reële getal ✓ en voeg by Totaal ✓ Eindig Terwyl lus Eindig while Maak lêer toe Bereken gemiddeld deur Totaal en iTel te gebruik ✓ Afgerond tot die naaste heelgetal ✓ Vertoon gemiddelde punt in pnlQ1_5 ✓</p> | 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 ✓ Alternatief: MID (Gender, 1,1) | 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 ✓ Alternatiiewe: Multilingual Multilingual LIKE True ORDER BY 2 | 6 | |
| 2.1.4 | Knoppie [2.1.4 – Lecturer salaries] SELECT LecturerID, FORMAT(Count(*) ✓ * 10000 ✓, "CURRENCY") ✓ AS Salary FROM tblCourses ✓ GROUP BY LecturerID ✓ Let wel: Count kan enige veldnaam in plaas van * gebruik. | 5 | |
| 2.1.5 | Knoppie [2.1.5 – Change online option] UPDATE tblCourses ✓ SET OnlineOption = FALSE ✓ WHERE CourseName LIKE ✓ "%Programming%" ✓ Alternatief: CourseName LIKE "%Programming" | 4 | |
| | Subtotaal: | 22 | |

VRAAG 2: NASIENRUBRIEK – VERVOLG

| | | | |
|-------|--|-----------|--|
| 2.2 | Databasismanipulasie | | |
| 2.2.1 | Knoppie [2.2.1 – Average duration of courses] | | |
| | <p>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 ✓</p> <p>Inisialiseer Teller en Som veranderlikes ✓</p> <p>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)</p> <p>Bereken gemiddelde duur: Som / Teller ✓ Vertoon gemiddelde duur geformatteer na twee desimale ✓</p> <p>tblLecturers.Next ✓ Eindig lus (tblLecturers)</p> | 14 | |
| 2.2.2 | Knoppie [2.2.2 – Register new lecturer] | | |
| | <p>tblLecturers.Insert; ✓ tblLecturers['LecturerID'] := 'ZT032'; tblLecturers['LecturerName'] := 'Zander'; tblLecturers['LecturerSurname'] := 'Thomas'; tblLecturers['Gender'] := 'Male'; tblLecturers['Multilingual'] := True; tblLecturers.Post; ✓</p> <p>Alternatiewe: Append i.p.v. Insert Enige navigeer-bevel in plaas van Post</p> <p>Toeken van veldwaardes: 1 punt vir die regte gebruik van al die veldname 1 punt vir die gebruik van regte waardes Trek 1 punt af vir elke fout tot 'n maksimum van twee punte</p> | 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 | <p>Constructor Create</p> <p>Stel attribute (fSchoolName, fTotalLearners, fPublicSchool) ✓ Na regte parameters ✓ Ken 'Z' toe aan fRating ✓</p> | 3 | |
| 3.1.2 | <p>Function getPublicSchool</p> <p>Funksie-opskrif met Boolese waarde as terugstuurtype ✓ result = fPublicSchool ✓</p> | 2 | |
| 3.1.3 | <p>Procedure updateRating</p> <p>Prosedure-opskrif ✓ met heelgetalparameter ✓ slaagPersentasie = parameter / fTotalLearners *100 ✓ as slaagPersentasie >= 80 ✓ fRating = 'A' ✓ anders if slaagPersentasie >= 60 ✓ fRating = 'B' ✓ anders fRating = 'C' ✓</p> <p>Aanvaar ook ander oplossings</p> <p>Let wel: Die reeks 79 – 80 kan ook as 'n aparte reeks hanteer word – beide ingesluit of uitgesluit.</p> | 8 | |
| 3.1.4 | <p>Function calcFunding</p> <p>Funksie-opskrif met real as terugstuur-datatype ✓ Result ✓ = fTotalLearners ✓ * 145.50 ✓</p> | 4 | |

| | | | |
|-------|---|-----------|--|
| 3.1.5 | <p>Function toString met string as terugstuur-datatype</p> <p>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 ✓</p> <p>As fPublicSchool ✓ Voeg 'Publieke skool' by ✓ Anders Voeg 'Privaatskool' by ✓ Stuur string terug ✓</p> | 7 | |
| | Subtotaal: Objekklas | 24 | |

VRAAG 3: NASIENRUBRIEK (VERVOLG)

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

BYLAAG D

VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING

| SENTRUMNOMMER: | | EKSAMENNOMMER: | |
|----------------|---|----------------|--------------|
| VRAAG | BESKRYWING | MAKS. PUNTE | LEERDER-PUNT |
| 4.1 | <p>Knoppie [4.1 – Codes]</p> <p>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</p> <p>Konsepte:</p> <p>Buitenste i lus (1 tot 5) (1) Binneste j lus (1 tot lengte(arrCodes[i])) (2) Toets karakter [i][j] (1) as getal OF letter (2) Verwyder karakters / bou string (2) Tel karakters verwyder (2) Voeg nuwe kode by op IstQ4_1 (1) en aantal karakters verwyder (1)</p> | 12 | |
| 4.2.1 | <p>Knoppie [4.2.1 – Extra IT periods]</p> <p>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</p> <p>Konsepte:</p> <p>Lus deur die dae van 1 tot 4 (1) // aanvaar ook 1 tot 5 Voorwaardelike lus/'Break'-stelling met for-lus (1) Bepaal indeks van (1) eerste leë spasie in 'n ry (1) Ken nuwe waarde aan leë spasie toe (2)</p> | 6 | |

| | | | |
|---------------------------|--|-----------|--|
| 4.2.2 | <p>Knoppie [4.2.2 – Group IT]</p> <p>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</p> <p>Konsepte</p> <p>Lus deur rye (1 tot 4) //1 <i>Bepaal die posisie van die eerste voorkoms van IT:</i> Inisialiseer/skep Teller-veranderlike van 1^{ste} posisie//1 Lus deur die kolomme //1 Toets of die selwaarde = 'IT' //1 Stoor/bepaal die indeks/posisie van 1st voorkoms //2 <i>Bepaal die posisie van die tweede voorkoms van IT:</i> Stoor/bepaal die indeks/posisie van 2^{de} voorkoms //2</p> <p>Ruil die vakkode na die eerste voorkoms met die tweede voorkoms van IT //4</p> | 12 | |
| TOTAAL AFDELING D: | | 30 | |

OPSOMMING VAN LEERDER SE PUNTE:

| SENTRUMNUMMER: | | 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 | | | | | |

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  
    iAantLeerdere, iAantSpanne, iAantReserve: integer;  
begin  
    // Provided code  
    memQ1_2.Clear;  
    //-----  
    iAantLeerdere := StrToInt(edtQ1_2.Text);  
    iAantSpanne := iAantLeerdere DIV PLAYERS;  
    iAantReserve := iAantLeerdere MOD PLAYERS;  
    memQ1_2.Lines.Add('Number of hockey teams: ' + IntToStr(iAantSpanne));  
    memQ1_2.Lines.Add('Number of learners on reserve list: ' + IntToStr  
        (iAantReserve));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(TextBox('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));  
            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; iTotallLearners: 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; iTotallLearners: integer;
  bPublicSchool: boolean);
begin
  // 3.1.1 Constructor Create
  fSchoolName := sSchoolName;
  fTotalLearners := iTotallLearners;
  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;
    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;
  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;

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

end.
```