



# basic education

---

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

**NASIONALE  
SENIOR SERTIFIKAAT**

**GRAAD 12**

**INLIGTINGSTEGNOLOGIE V1**

**NOVEMBER 2019**

**PUNTE: 150**

**TYD: 3 uur**

**Hierdie vraestel bestaan uit 21 bladsye en 2 bladsye met data.**

**INSTRUKSIES EN INLIGTING**

1. Hierdie vraestel is in VIER afdelings ingedeel. Kandidate moet AL die vrae in elk van die VIER afdelings beantwoord.
2. 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.
3. Hierdie vraestel is opgestel met programmeringsterme wat spesifiek tot die Delphi-programmeringstaal is.
4. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal volgens die voorgeskrewe vereistes toegeken word.
5. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, indien die vraag nie datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
6. 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.
7. Roetines, soos soek, sorteer en seleksie, moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van Delphi vir enige van hierdie roetines gebruik NIE.
8. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture voorsien word.
9. Jy moet jou werk gereeld op die disket/CD/DVD/geheuestokkie stoor wat aan jou gegee is, of op die skyfspasie wat vir hierdie eksamensessie aan jou toegeken is.
10. Maak seker dat jou eksamennommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
11. Indien dit vereis word, druk die programmeringskode van al die programme/klasse wat jy voltooi het. Jy sal ná die eksamensessie 'n halfuur tyd vir drukwerk gegee word.
12. 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.

13. 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 wagwoordbeskermdede uitvoerbare lêers verskaf.

**LET WEL:** Kandidate moet die lêer **DataAFRNov2019.exe** gebruik.

Doen die volgende:

- Dubbelklik op die wagwoordbeskermdede uitvoerbare lêer:  
**DataAFRNov2019.exe**
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **SPoRt@ScHOOL%**

Nadat dit onttrek ('extracted') is, sal die volgende lys lêers in die lêergids ('folder') **DataAFRNov2019** beskikbaar wees:

**LÊERS WAT VOORSIEN IS:**

**Vraag 1:**

Vraag1\_P.dpr  
Vraag1\_P.dproj  
Vraag1\_P.res  
Vraag1\_U.dfm  
Vraag1\_U.pas

**Vraag 2:**

ConnectDB\_U.dcu  
ConnectDB\_U.pas  
HokkieDB.mdb  
Vraag2\_P.dpr  
Vraag2\_P.dproj  
Vraag2\_P.res  
Vraag2\_U.dfm  
Vraag2\_U.pas

**Vraag 3:**

Speler\_U.pas  
Vraag3\_P.dpr  
Vraag3\_P.dproj  
Vraag3\_P.res  
Vraag3\_U.dfm  
Vraag3\_U.pas

**Vraag 4:**

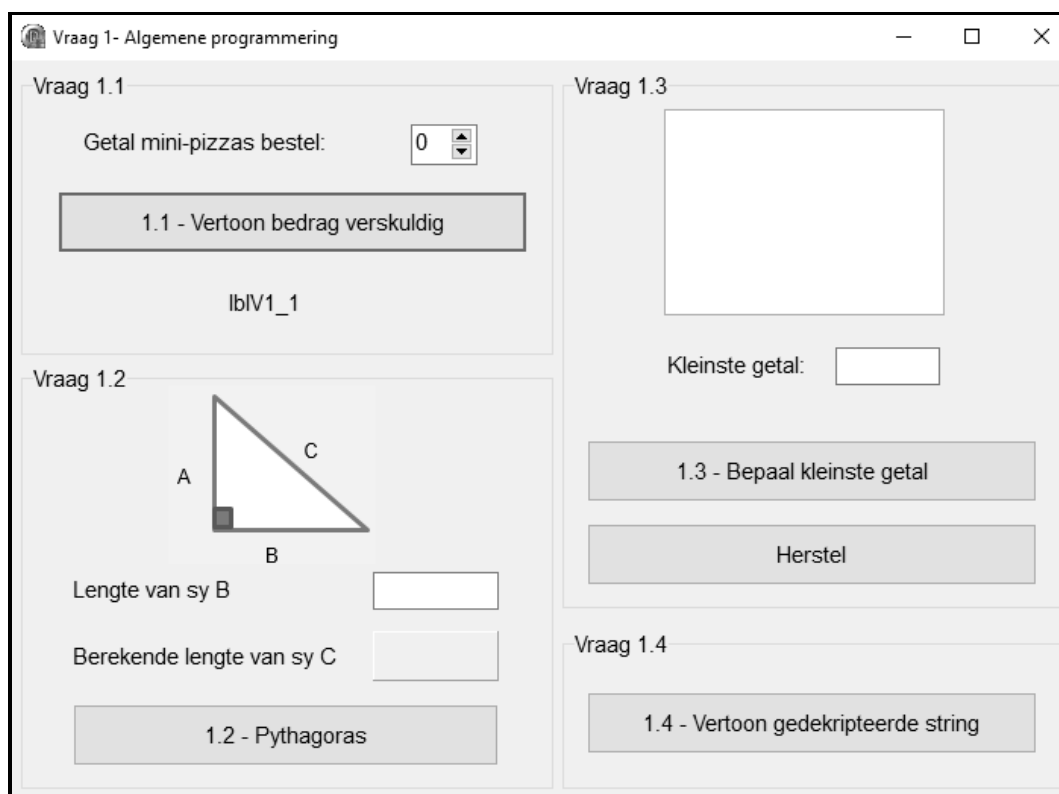
Doolhof 1.txt  
Doolhof 2.txt  
Doolhof 3.txt  
Vraag4\_P.dpr  
Vraag4\_P.dproj  
Vraag4\_P.res  
Vraag4\_U.dfm  
Vraag4\_U.pas

**AFDELING A****VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

Doen die volgende:

- Maak die onvolledige projeklêer met die naam **Vraag1\_P.dpr** in die **Vraag 1-lêergids** oop.
- Sleutel jou eksamenommer as 'n kommentaar in die eerste reël van die **Vraag1\_U.pas**-lêer in.
- Kompileer en voer die program uit. Die gebruikerskoppelvlak vertoon VIER verskillende afdelings wat Vraag 1.1 tot Vraag 1.4 genoem word. Die program het tans geen funksionaliteit nie.

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



- Voltooi die kode vir ELKE afdeling van VRAAG 1, soos beskryf in VRAAG 1.1 tot VRAAG 1.4 wat volg.

1.1 **Knoppie [1.1 - Vertoon bedrag verskuldig]**

Mini-pizzas kan teen R14,95 elk by die snoepie by 'n skool bestel word.

Skryf kode om die volgende te doen:

- Verklaar 'n konstante veranderlike, **PRYS**, om die waarde 14.95 te bevat.
- Verklaar TWEE veranderlikes om die getal mini-pizzas wat bestel is en die bedrag wat verskuldig is, onderskeidelik te stoor.
- Verander die fontgrootte van die byskrif ('label') **lbIV1\_1** na 20 pt.
- Herwin ('Retrieve') die getal mini-pizzas wat bestel is uit die **spnV1\_1**-spin-redigeerblokkie ('spin edit box').
- Bereken die bedrag verskuldig deur die konstante veranderlike **PRYS** en veranderlike vir die getal mini-pizzas wat bestel is, te gebruik.
- Vertoon die bedrag verskuldig op die byskrif **lbIV1\_1**, as geldeenheid geformatteer.

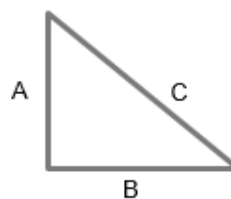
Voorbeeld van afvoer as die getal mini-pizzas bestel, 5 is:

(8)

1.2 **Knoppie [1.2 - Pythagoras]**

Pythagoras se teorie is in Wiskunde die fundamentele verhouding in Euklidiese meetkunde ten opsigte van die drie sye van 'n reghoekige driehoek.

Die formule  $C^2 = A^2 + B^2$  word gebruik om die lengte van die sy teenoor die reghoek in die driehoek hieronder te bereken.



$$C^2 = A^2 + B^2$$

$$C = \sqrt{A^2 + B^2}$$

Skryf kode om die volgende te doen:

- Verklaar veranderlikes vir sye A, B en C.
- Ken die waarde van 4 aan die veranderlike vir sy A toe.
- Verkry die lengte van sy B vanaf die **edtV1\_2**-redigeerblokkie en ken dit aan die veranderlike vir sy B toe.
- Bereken die lengte van sy C deur die formule  $C = \sqrt{A^2 + B^2}$  te gebruik.
- Vertoon die lengte van sy C in die **pnIV1\_2**-komponent as 'n reële getal, tot EEN desimale plek afgerond.

Voorbeeld van afvoer as 6.5 as die lengte van sy B ingesleutel is:

(10)

### 1.3 Knoppie [1.3 - Bepaal kleinste getal]

'n Globale veranderlike, **iKleinste**, word verklaar en na die waarde 100 geïnisialiseer.

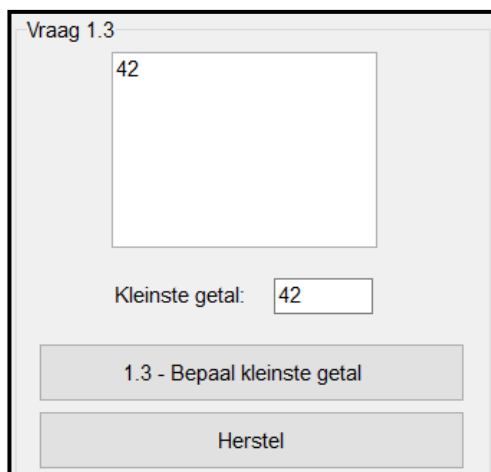
Skryf kode om die volgende te doen as die **Bepaal kleinste getal**-knoppie geklik word:

- Genereer en ken 'n willekeurige getal in die reeks van 1 tot 100 (inklusief) toe aan 'n lokale veranderlike **iGetal** wat verskaf is.
- Vertoon die getal wat in die **redV1\_3** 'rich edit' gegenereer is.
- Vervang die huidige kleinste getal wat in die **iKleinste**-veranderlike met die gegenereerde getal gestoor is, indien die gegenereerde getal kleiner is as die getal wat in die **iKleinste**-veranderlike gestoor is.
- Vertoon die kleinste getal in die **edtV1\_3**-redigeerblokkie ('edit box').

#### LET WEL:

- Die willekeurige getal moet by die 'rich edit' gevoeg word met elke klik van die knoppie.
- Kode word in die **Herstel**-knoppie verskaf om die kleinste getal na die beginwaarde van 100 te stel en die inhoud van die toevoer- en afvoercomponente skoon te maak.

Voorbeeld van afvoer as die knoppie een keer geklik word en die willekeurige getal 42 gegenereer word:



Vraag 1.3

42

Kleinste getal: 42

1.3 - Bepaal kleinste getal

Herstel

Voorbeeld van afvoer as die knoppie ses keer geklik word:

**LET WEL:** Die afvoer wat deur jou program vertoon word, kan van die voorbeeldafvoer verskil omdat willekeurige getalle gegenereer word.

(9)

#### 1.4 Knoppie [1.4 - Vertoon gedekripteerde string]

'n Geënkripteerde string wat deur die gebruiker ingesleutel word, moet gedekripteer en vertoon word. Die geënkripteerde string bevat syfers omdat sommige van die alfabetiese karakters in die string met syfers volgens die tabel hieronder vervang is.

ALFABETIESE KARAKTER	VERVANGINGSYFER
A	0
B	1
C	2
D	3
E	4
F	5
G	6
H	7
I	8
J	9

Voorbeeld:

Geënkripteerde weergawe van die string: T8M4!  
Oorspronklike string (gedekripteer): TIME!

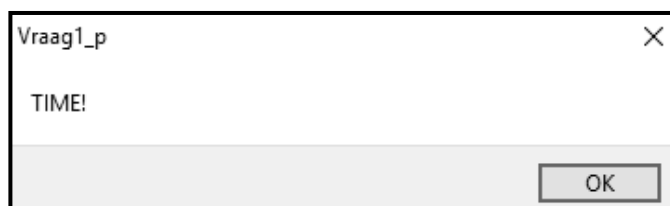
Skryf kode om die volgende te doen om 'n geënkripteerde string te dekripteer:

- Gebruik 'n toevoerblokkie ('input box') om 'n geënkripteerde string in te sleutel.
- Vervang elke syfer wat in die geënkripteerde string voorkom met die alfabetiese karakter wat dit verteenwoordig (sien tabel hierbo).
- Vertoon die gedekripteerde string in 'n boodskapblokkie ('message box').

Voorbeeld van toevoer as T8M4! as 'n geënkripteerde string ingesleutel is:



Afvoer vir die geënkripteerde string T8M4!:



Toets jou kode deur die volgende toetsdata te gebruik:

90R verteenwoordig JAR

03V4NTUR4 T8M4! verteenwoordig ADVENTURE TIME!

039024NT verteenwoordig ADJACENT

(13)

- Maak seker dat jou eksamenommer as kommentaar in die eerste reël van die programlêer ingesleutel is.
- Stoor jou program.
- Druk die kode indien dit vereis word.

**TOTAAL AFDELING A: 40**

**AFDELING B****VRAAG 2: DATABASISPROGRAMMERING**

Die hokkie-organiseerder by jou skool het jou hulp nodig met die administrasie van die spelers, spanne en afrigters.

Die databasis **HokkieDB** bevat twee tabelle met die name **tblSpelers** en **tblSpanne**.

Die bladsye met data wat aan die einde van hierdie vraestel aangeheg is, verskaf inligting oor die ontwerp van die databasis en die inhoud van die tabelle.

Doen die volgende:

- Maak die onvolledige projeklêer met die naam **Vraag2\_P.dpr** in die **Vraag 2-lêergids** oop.
- Sleutel jou eksamennummer as kommentaar in die eerste reël van die **Vraag2\_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 hieronder getoon op die seleksie van die **Oortjieblad ('Tab sheet') Vraag 2.2 - Delphi-kode**.

SpelerID	SpelerVan	SpelerNaam	IDNommer	SpanNaam	VaardigheidsVlak	DoelWagter
▶ 1	Fivaz	Peter	0306170790504	o/16 A	6	False
2	Maphaphu	Bukelwa	0609220225852	o/14 B	3	False
3	Snyders	Con	0602200518279	o/14 B	3	False
4	Jagers	Ben	0403060227644	o/16 B	4	True

SpanNaam	Afrigter	GetalWedstrydeGespeel	GetalWedstrydeGewen
▶ o/14 A	Modikaze, P	8	4
o/14 B	Smith, GP	6	3
o/16 A	Smit, J	8	7
o/16 B	Mpofu, XB	6	3

2.2.1 - Junior spelers in o/18 A-span

Getal junior spelers in o/18 A-span:

2.2.1 - Junior spelers in o/18 A-span

2.2.2 - Afrigter en doelwagter se inligting

2.2.2 - Afrigter en doelwagter se inligting

Herstel databasis
Sluit af

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

### LET WEL:

- Die 'Herstel databasis'(Restore database)-knoppie word verskaf om die data wat in die databasis voorkom, na die oorspronklike inhoud te herstel.
- Die inhoud van die databasis word met 'n wagwoord beskerm, met ander woorde jy sal nie in staat wees om toegang tot die databasis te kry deur Microsoft Access te gebruik nie.
- Kode word verskaf om die GGK('GUI')-komponente met die databasis te koppel. Moet NIE enige van die kode wat verskaf word, verander NIE.
- TWEE veranderlikes word as globale veranderlikes verklaar, soos wat in die tabel hieronder beskryf word.

Variable	Data type	Description
tblSpanne	TADOTable	Verwys na die tabel <b>tblSpanne</b> in die databasis <b>HokkieDB</b>
tblSpelers	TADOTable	Verwys na die tabel <b>tblSpelers</b> in die databasis <b>HokkieDB</b>

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

Voorbeeld van die GGK ('GUI') vir VRAAG 2.1:

### LET WEL:

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

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

### 2.1.1 Knoppie [2.1.1 - Beste spelers]

Vertoon die vanne en name van al die spelers met 'n vaardigheidsvlak van 10.

Voorbeeld van afvoer van die eerste vier rekords:

SpelerVan	SpelerNaam
Mjikelo	Karel
Goliath	Siphokazi
Ncamiso	Nozipo
Baxter	Nomhle

(3)

### 2.1.2 Knoppie [2.1.2 - B-span-afrigters]

Vertoon die name van die afrigters en spanne van al die B-spanne.

Voorbeeld van afvoer:

Afrigter	SpanNaam
Smith, GP	o/14 B
Mpofu, XB	o/16 B
Mullan, NV	o/18 B

(4)

### 2.1.3 Knoppie [2.1.3 - Persentasie wedstryde gewen]

Kode is voorsien om 'n span se naam uit die kombinasieblokkie ('combo box'), **cmbV2\_1\_3**, te onttrek.

Vertoon die naam van die span, hulle afrigter en die **persentasie** wedstryde wat die span gewen het. Stoor die persentasie wedstryde wat gewen is in 'n berekende veld met die naam **PersentasieWedstrydeGewen**.

Voorbeeld van afvoer as die o/14 B-span geselekteer is:

SpanNaam	Afrigter	PersentasieWedstrydeGewen
o/14 B	Smith, GP	50

**LET WEL:** Jy hoef NIE die berekende waarde te formateer NIE.

(4)

### 2.1.4 Knoppie [2.1.4 – Spangemiddeld van meer as 6]

Die gemiddelde vaardigheidsvlakke van spanne word gebruik om spanne te identifiseer met die grootste moontlikheid om hulle wedstryde te wen.

Vertoon die name en gemiddelde vaardigheidsvlakke van alle spanne met 'n gemiddelde vaardigheidsvlak van meer as 6. Die gemiddelde vaardigheidsvlak per span moet in 'n berekende veld met die naam **GemidVaardigheidsVlak** gestoor word, tot EEN desimale plek geformatteer.

Voorbeeld van afvoer:

SpanNaam	GemidVaardigheidsVlak
o/14 A	8.2
o/16 A	7.7
o/18 A	8.4

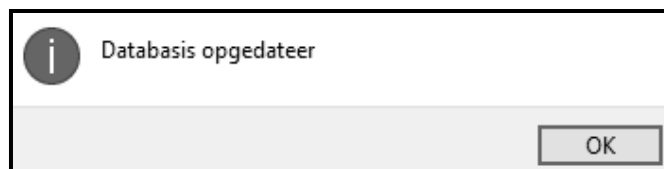
(5)

### 2.1.5 Knoppie [2.1.5 – Dateer wedstryde gewen op]

Die uitslag van die wedstryde wat tydens die laaste sportdag gewen is, moet gebruik word om die **GetalWedstrydeGewen**-veld op te dateer. Slegs die o/14 B-span het hulle wedstryd verloor.

Dateer die data in die **tblSpanne**-tabel op deur 'n waarde van 1 by die **GetalWedstrydeGewen**-veld te tel vir die spanne wat hulle wedstryde gewen het.

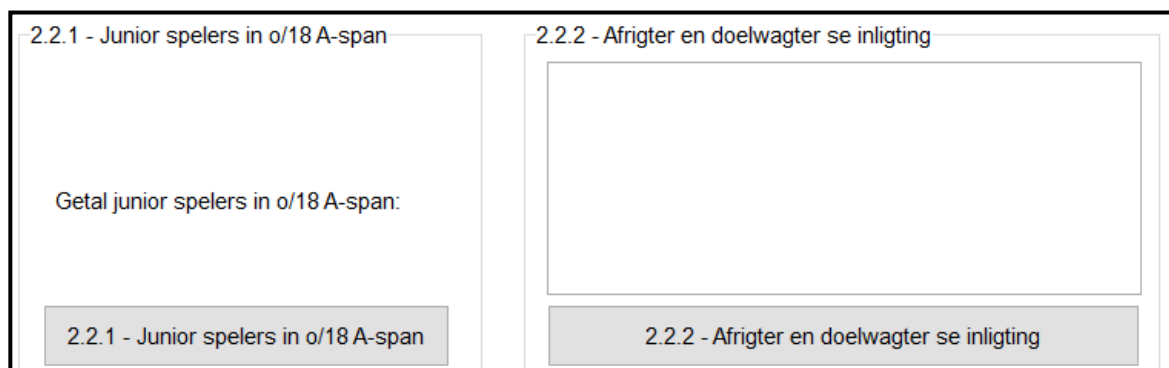
Voorbeeld van afvoer:



(3)

## 2.2 Oortjieblad ('Tab sheet') [Vraag 2.2 - Delphi-kode]

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



### LET WEL:

- Gebruik SLEGS Delphi-programmingskode 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.

### 2.2.1 Knoppie [2.2.1 – Junior spelers in o/18 A-span]

Die o/18 A-span sluit sommige junior spelers in wat uitsonderlik talentvol is. Junior spelers is spelers wat ná die jaar 2002 gebore is.

**LET WEL:** Die eerste twee syfers van die **IDNommer**-veld dui die geboortejaar van 'n speler aan.

Skryf kode om die volgende te doen:

- Stoor die vanne en name van al die junior spelers wat lede van die o/18 A-span is in 'n nuwe tekslêer met die naam **Junior18A.txt**.
- Bepaal die totale getal junior spelers in die o/18 A-span en vertoon die resultaat in die **lbIV2\_2\_1**-byskrif ('label').

Voorbeeld van inhoud van die **Junior18A**-tekslêer:

Cannon Julian Goliath Siphokazi
------------------------------------

Voorbeeld van afvoer wat op die **lbIV2\_2\_1**-byskrif vertoon moet word:

Getal junior spelers in o/18 A-span: 2
--

(11)

### 2.2.2 Knoppie [2.2.2 – Afrigter en doelwagter se inligting]

Die afrigter en doelwagter van al die spanne word na 'n spesiale oefensessie uitgenooi.

Kode is verskaf om al die verskillende kolomwydtes te stel en om die opskrifte te vertoon, soos in die voorbeeld van afvoer getoon.

Skryf kode om 'n lys te vertoon wat die volgende inligting oor elke span bevat:

- Naam van die span
- Van en voorletters van die afrigter
- Van en naam van die doelwagter in die formaat:

<Van>, <Naam>

Voorbeeld van afvoer van die eerste vyf rekords:

Span	Afrigter	Doelwagter
o/14 A	Modikaze, P	Nel, Koos
o/14 B	Smith, GP	Scheepers, Kurtley
o/16 A	Smit, J	Phillips, Moses
o/16 B	Mpofu, XB	Jagers, Ben
o/18 A	Decan, H	David, Ivan

(10)

- |  |
|--|
| <ul style="list-style-type: none"> <li>• Maak seker dat jou eksamennommer as kommentaar in die eerste reël van die programlêer ingesleutel is.</li> <li>• Stoor jou program.</li> <li>• Druk die kode indien dit vereis word.</li> </ul> |
|--|

**TOTAAL AFDELING B: 40**

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

Die skool ontwerp sagteware om die liggaamsmassa-indeks (LMI/'body mass index (BMI)') van die rugbyspelers te bereken en om die geskiktheid van die rugbyspelers vir seleksie te bepaal.

Doen die volgende:

- Maak die onvolledige program in die **Vraag 3**-lêergids oop.
- Maak die onvolledige objekklas **Speler\_U.pas** oop.
- Sleutel jou eksamennommer as kommentaar in die eerste reël van beide die **Vraag3\_U.pas**-lêer en die **Speler\_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.

Voorbeeld van die GGK ('GUI'):

- Voltooi die kode soos in VRAAG 3.1 vir die **Speler\_U**-objekklas en VRAAG 3.2 vir die **Vraag3\_U**-vormklas gespesifiseer.

3.1 Die onvolledige objekklas (**TSpeler**) wat verskaf is, bevat die verklarings van drie attribute wat 'n **Speler**-objek definieer.

Die attribute vir die **Speler**-objek is soos volg verklaar:

Name van attribute	Beskrywing
fSpelerNaam	Die naam van die rugbyspeler
fGewigVanSpeler	Die gewig van die speler
fTelling	Die telling by 'n spesifieke rugbywedstryd behaal

- 3.1.1 Skryf kode vir 'n **konstruktor**-metode wat die speler se naam en gewig as parameterwaardes sal ontvang. Ken hierdie waardes aan die onderskeie attribute toe. Stel die telling-attribuut op die waarde van nul. (4)
- 3.1.2 Skryf kode vir 'n toegangsmetode ('accessor method') met die naam **getTelling** vir die **fTelling**-attribuut. (2)
- 3.1.3 Skryf kode vir 'n metode met die naam **dateerTellingOp** wat 'n heelgetalwaarde as parameter sal ontvang en die waarde wat ontvang is, by die **fTelling**-attribuut sal tel. (3)
- 3.1.4 Skryf kode vir 'n metode met die naam **berekenLMI** wat die lengte van die speler as 'n parameter moet ontvang en die speler se LMI sal bereken en terugstuur, op die volgende formule gebaseer: (3)
- $$\text{LMI} = \text{gewig van speler} / (\text{lengte van speler})^2$$
- 3.1.5 Skryf kode vir 'n metode met die naam **geskikVirSeleksie** wat gebruik kan word om die moontlikheid te bepaal om geselekteer te word om by die Provinsiale Uitdunne-toernooi te speel. Moontlike seleksie word bepaal deur die inhoud van die **fTelling**-attribuut volgens die volgende kategorieë te evalueer:

Telling	Boodskap
0 tot 7 punte	Klein moontlikheid
8 tot 14 punte	Gemiddelde moontlikheid
Meer as 14 punte	Groot moontlikheid

Die metode moet die relevante boodskap terugstuur. (4)

- 3.1.6 Skryf kode vir 'n **toString**-metode om die attribute van die speler-objek in die volgende formaat te vertoon:

```
Naam: <fSpelerNaam>
Gewig: <fGewigVanSpeler>
Huidige telling: <fTelling>
```

Voorbeeld:

```
Naam: Olaff
Gewig: 70.3
Huidige telling: 0
```

(4)

- 3.2 'n Onvolledige eenheid **Vraag3\_U** is verskaf en bevat kode vir die objekklas om toeganklik te wees.

Die volgende globale veranderlike is verklaar:

Die objek **objSpeler**

Doen die volgende om die kode vir VRAAG 3.2.1 tot VRAAG 3.2.4 in die hoofvormeenheid te voltooi:

### 3.2.1 Knoppie [3.2.1 - Instansieer objek]

Skryf kode om die volgende te doen:

- Gebruik die naam en gewig van die speler uit die redigeerblokkies ('edit boxes') wat verskaf is om 'n Speler-objek te instansieer.
- Vertoon 'n boodskap, deur 'n dialoogblokkie ('dialog box') te gebruik, dat die objek geïnstansieer is.

(5)

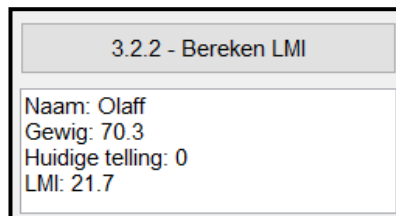
### 3.2.2 Knoppie [3.2.2 - Bereken LMI]

Die **redV3\_2\_2**-komponent moet as die vertoonarea gebruik word.

Skryf kode om die volgende te doen:

- Gebruik 'n toevoerdialoogblokkie ('input dialog box') om die lengte van die speler in te sleutel.
- Roep die relevante metode wat lengte as 'n argument gebruik om die LMI van die speler te bereken.
- Roep die **toString**-metode om die inligting van die speler-objek te vertoon.
- Vertoon die LMI van die speler, tot EEN desimale plek afgerond.

Voorbeeld van afvoer as die gewig van die speler 70,3 kg en die lengte van die speler 1,8 m is:



(7)

### 3.2.3 Knoppie [3.2.3 - Dateer telling op]

Die telling van 'n speler word opgedateer soos wat die wedstryd ontwikkel.

Die gebruiker moet 'n telling in die radiogroep ('radio group') met die naam **rgpV3\_2\_3** selekteer en op die **Dateer telling op**-knoppie klik elke keer as die speler gedurende die wedstryd punte aanteken.

Skryf kode om die volgende te doen:

- Onttrek die telling wat uit die komponent **rgpV3\_2\_3** geselekteer is.
- Roep die korrekte metode om die telling-attriboot op te dateer.
- Roep die korrekte metode om die telling terug te stuur
- Vertoon die opgedateerde telling van die speler in die **pnIV3\_2\_3**-komponent.

Voorbeeld van afvoer as die waarde van 2 as die eerste telling geselekteer is en die **Dateer telling op**-knoppie geklik is:

Voorbeeld van afvoer as die waarde van 5 as die volgende telling geselekteer is en die **Dateer telling op**-knoppie geklik is:

(6)

### 3.2.4 Knoppie [3.2.4 - Geskik vir seleksie]

Skryf kode om die metode wat 'n boodskap terugstuur wat die speler se geskiktheid vir seleksie aandui, te roep. Vertoon die boodskap in die byskrif **IbIV3\_2\_4**.

Voorbeeld van afvoer as die huidige telling van die speler 8 is:

Voorbeeld van afvoer as die huidige telling van die speler 15 is:

(2)

- Maak seker dat jou eksamennummer as kommentaar in die eerste reël van die objekklas en die vormklas ingesleutel is.
- Stoor al die lêers.
- Druk die kode indien dit vereis word.

**TOTAAL AFDELING C: 40**

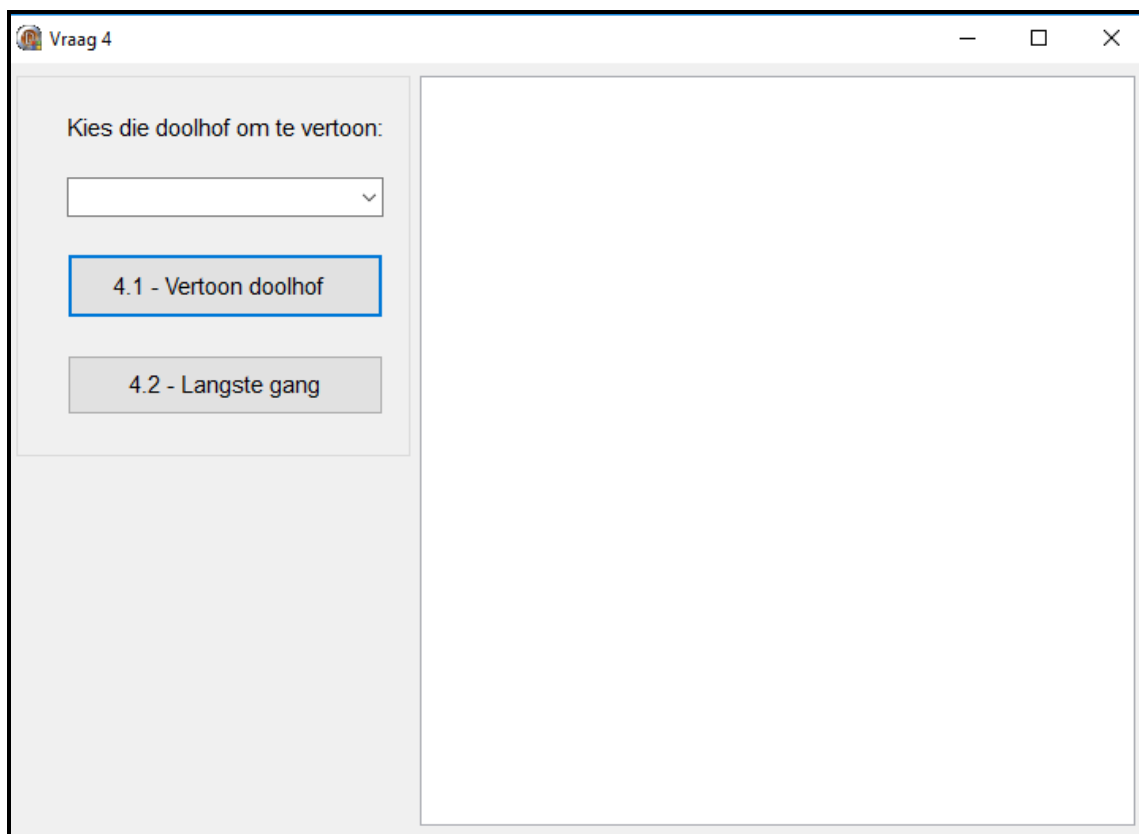
**AFDELING D****VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING****SCENARIO**

'n Skool gebruik 'n doolhof as 'n spanbou-aktiwiteit vir die rugby en netbal se eerste spanne.

Doen die volgende:

- Maak die onvolledige program in die **Vraag 4**-lêergids oop.
- Sleutel jou eksamenommer as kommentaar in die eerste reël van die **Vraag4\_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.

Voorbeeld van die GGK ('GUI'):



Die volgende kode is voorsien:

```
const
    iRyTel = 16;
var
    arrDoolhof : array[1..iRyTel] of String;
```

Die doolhof word voorgestel deur verskillende karakters soos volg te gebruik:

- Die '@'-karakter dui die buitenste grens van die doolhof aan.
- Die '#'-karakter dui 'n hindernis (muur) aan.
- Die strepie('-)-karakter dui 'n oop spasie (gang) aan.

Voltooi die kode van ELKE afdeling van VRAAG 4, soos beskryf in VRAAG 4.1 en VRAAG 4.2 wat volg.

#### 4.1 Knoppie [4.1 - Vertoon doolhof]

'n Kombinasieblokkie ('combo box') met die naam **cmbV4\_1** is met die name van drie doolhowe, naamlik **Doolhof 1**, **Doolhof 2** en **Doolhof 3**, gevul. Drie tekslêers met die name **Doolhof 1.txt**, **Doolhof 2.txt** en **Doolhof 3.txt** is verskaf. Elke tekslêer bevat reëls met karakters wat die uitleg van elke doolhof onderskeidelik verteenwoordig.

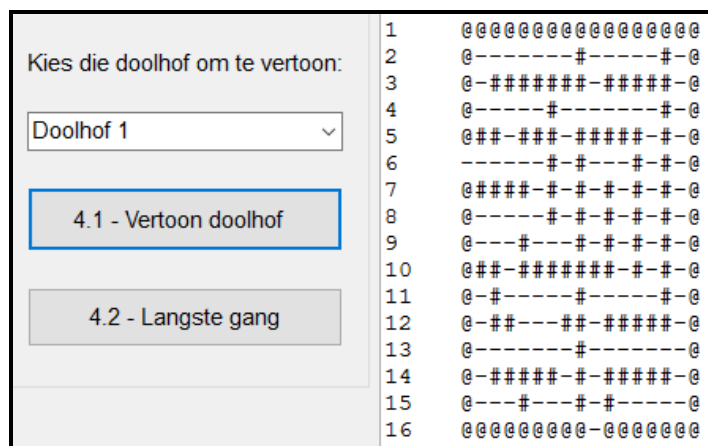
Die gebruiker moet 'n doolhof uit kombinasieblokkie **cmbV4\_1** selekteer.

Kode is verskaf om die komponent met die naam **redV4** skoon te maak.

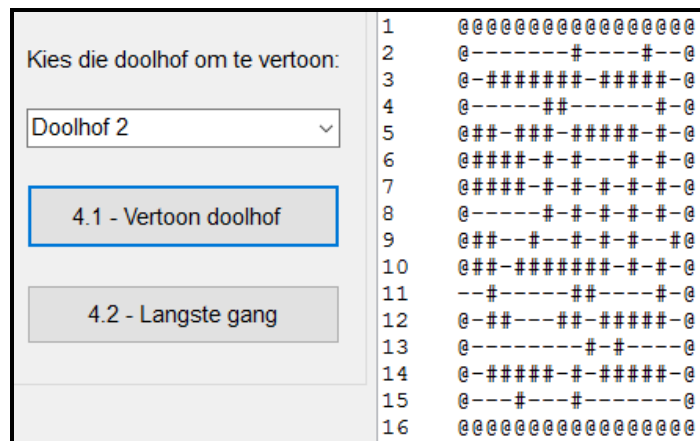
Skryf kode om die volgende te doen:

- Onttrek ('Extract') die naam van die geselekteerde doolhof uit die kombinasieblokkie **cmbV4\_1**.
- Vertoon 'n geskikte boodskap as 'n tekslêer NIE vir die geselekteerde doolhof bestaan NIE.
- Doen die volgende as 'n tekslêer vir die geselekteerde doolhof bestaan:
  - Lees EEN reël op 'n slag uit die tekslêer.
  - Stoor ELKE reël by die regte indeks in die skikking **arrDoolhof** wat verskaf is.
  - Vertoon ELKE reël van die doolhof met sy ooreenstemmende reëlnommer in die afvoercomponent **redV4**.

Voorbeeld van afvoer as **Doolhof 1** uit die kombinasieblokkie geselekteer is:



Voorbeeld van afvoer as **Doolhof 2** uit die kombinasieblokkie geselekteer is:



(11)

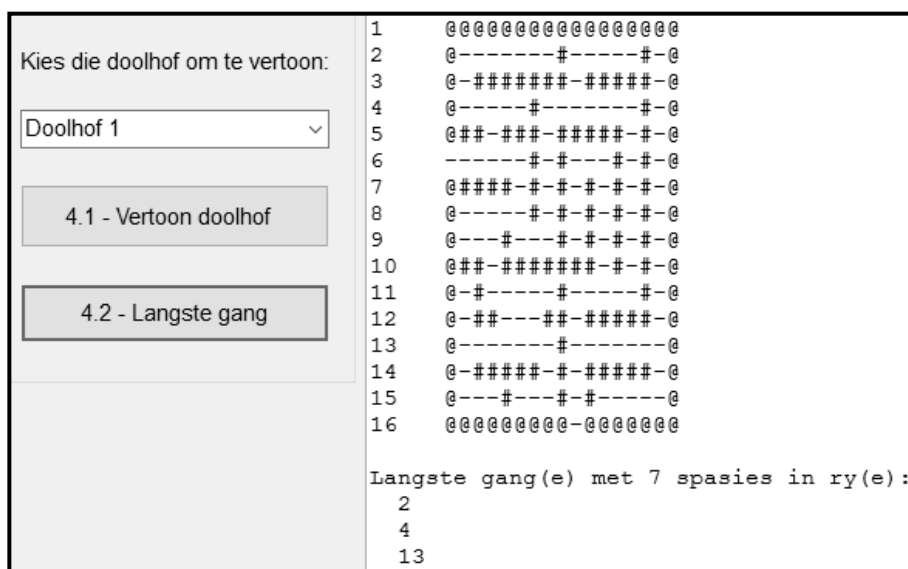
### 4.2 Knoppie [4.2 - Langste gang]

Die langste gang verwys na die maksimum getal opeenvolgende horisontale oop spasies in 'n ry in die doolhof. 'n Oop spasie word deur die strepie('-')-karakter aangedui.

Skryf kode om die volgende te doen:

- Bepaal die maksimum getal opeenvolgende horisontale oop spasies in 'n ry in die skikking **arrDoolhof**.
- Vertoon die maksimum getal opeenvolgende horisontale oop spasies as deel van 'n afvoerstelling.
- Vertoon 'n lys van ry(e) wat 'n gang met die maksimum getal opeenvolgende horisontale oop spasies bevat.

Voorbeeld van afvoer vir **Doolhof 1**:



Voorbeeld van afvoer vir **Doolhof 3**:

```

Kies die doolhof om te vertoon:
Doolhof 3
4.1 - Vertoon doolhof
4.2 - Langste gang

1 @@@@@@@@@@@@@@@@@@
2 @-----#-----#-@
3 @-#####-#####-@
4 @-----#-----#-@
5 @##-##-#####-#-@
6 @-----#-#-#-#-#-@
7 @###-#-#-#-#-#-@
8 @-----#-#-#-#-#-@
9 @--#--##-#-#-#-#-@
10 @##-#####-#-#-@
11 @-#-----#-----#-@
12 @-##---##-#####-@
13 @-----#-----@
14 @-#####-#####-@
15 @--#-----#-----@
16 @@@@@@@@@@-@@@@@@@@

Langste gang(e) met 11 spasies in ry(e):
15
    
```

(19)

- Maak seker dat jou eksamenommer as kommentaar in die eerste reël van die programlêer ingesleutel is.
- Stoor jou program.
- Druk die kode indien dit vereis word.

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

**INLIGTINGSTEGNOLOGIE V1****DATABASISINLIGTING VAN HokkieDB VIR VRAAG 2:**

Die ontwerp van die databasistabelle is soos volg:

Tabel: **tblSpanne**

Hierdie tabel bevat die data van al die hokkie-afrigters.

Field name	Data type	Description
SpanNaam (PK)	Text (10)	'n Unieke spannaam. 'n Span se naam word in die formaat 'o/18 A' gestoor, waar A 'n A-span aandui, B 'n B-span en so aan. Die getal 18 is die ouderdomsgroep van die span se spelers.
Afrigter	Text (25)	Die van en voorletters van die span se afrigter
GetalWedstrydeGespeel	Integer	Die totale getal wedstryde wat die span hierdie seisoen gespeel het
GetalWedstrydeGewen	Integer	Die totale getal wedstryde wat die span hierdie seisoen gewen het

Voorbeeld van die eerste vier rekords in die **tblSpanne**-tabel:

SpanNaam	Afrigter	GetalWedstrydeGespeel	GetalWedstrydeGewen
o/14 A	Modikaze, P	8	4
o/14 B	Smith, GP	6	3
o/16 A	Smit, J	8	7
o/16 B	Mpofu, XB	6	3

Tabel: **tblSpelers**

Die tabel bevat die data van die hokkiespelers van drie verskillende ouderdomsgroepe:

Field name	Data type	Description
SpelerID (PK)	Autonumber	'n Unieke nommer wat aan die speler toegeken word
SpelerVan	Text (25)	Die van van die speler
SpelerNaam	Text (25)	Die naam van die speler
IDNommer	Text (20)	Die Suid-Afrikaanse ID-nommer – eerste 6 syfers is die geboortedatum van die persoon in die formaat yymmdd.
SpanNaam	Text (10)	Die span se naam waarvan die speler 'n lid is, bv. 'o/18 A'
VaardigheidsVlak	Integer	'n Waarde in die reeks 1 tot 10 wat die speler se vaardigheidsvlak aandui. 'n Waarde van 1 dui 'n lae vaardigheidsvlak aan. 'n Waarde van 10 dui 'n hoë vaardigheidsvlak aan.
DoelWagter	Boolean	'n Waarde wat aandui of die speler 'n doelwagter is (true) of nie (false)

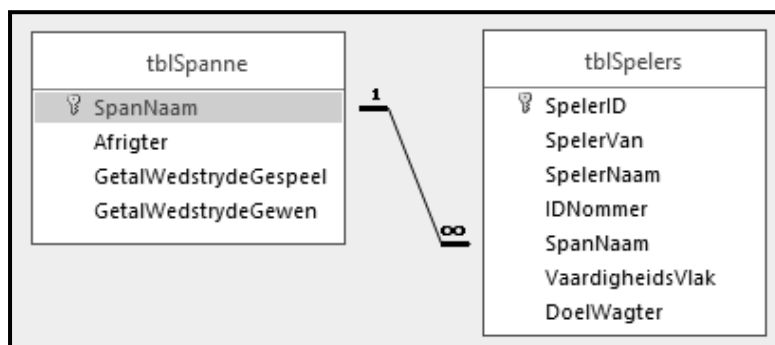
Voorbeeld van die eerste vier rekords van die **tblSpelers**-tabel:

SpelerID	SpelerVan	SpelerNaam	IDNommer	SpanNaam	VaardigheidsVlak	DoelWagter
1	Fivaz	Peter	0306170790504	o/16 A	6	False
2	Maphaphu	Bukelwa	0609220225852	o/14 B	3	False
3	Snyders	Con	0602200518279	o/14 B	3	False
4	Jagers	Ben	0403060227644	o/16 B	4	True

### LET WEL:

- Konneksiekode is verskaf.
- Die databasis is met 'n wagwoord beskerm, daarom sal jy nie in staat wees om direk toegang tot die databasis te verkry nie.

Die volgende een-tot-baie-verwantskap met referensiële integriteit bestaan tussen die twee tabelle in die databasis:





# basic education

Department:  
Basic Education  
**REPUBLIC OF SOUTH AFRICA**

## **NASIONALE SENIOR SERTIFIKAAT**

**GRAAD 12**

**INLIGTINGSTEGNOLOGIE V1**

**NOVEMBER 2019**

**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 3-12) sluit die nasienrubriek vir elke om te gebruik vir enigeen van die twee programmeringstale in.
- **Bylaag E, F, G en H** (bladsy 13-24) 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 3-12) 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><b>Knoppie [1.1 – Vertoon bedrag verskuldig]</b></p> <p>Verklaar 'n konstante veranderlike PRYS en ken waarde 14.95 toe✓            Verklaar 'n reële veranderlike vir die bedrag verskuldig✓            Verklaar 'n veranderlike vir die aantal pizzas✓            Kry getal uit spnV1_1✓            Bereken bedrag deur PRYS te gebruik✓            Verander skrifgrootte van byskrif ('label') na 20pt ✓            Vertoon bedrag op lblV1_1✓ omgeskakel na string en geldeenheid-formaat ✓</p>	8	
1.2	<p><b>Knoppie [1.2 – Pythagoras]</b></p> <p>Skep veranderlikes vir sye A, B en C✓            Ken lengte van 4 toe aan A ✓            Lengte vir sy B ✓ en skakel om na getal formaat✓  <math>C = \text{sqr}(\text{sqr}(A) + \text{sqr}(B))</math>                korrekte gebruik van <math>\text{sqr}(A)</math> OF <math>A * A</math>✓                korrekte gebruik van <math>\text{sqr}(rB)</math> OF <math>B * B</math>✓                kry die som van die twee waardes✓            Vertoon lengte van sy C op paneel✓            geformateer tot een desimale plek ✓</p> <p><b>Aanvaar:</b>            Sqr: Power(A,2)            Sqrt: Power((sqr(A) +sqr(B)),0.5)</p>	10	
1.3	<p><b>Knoppie [1.3 – Bepaal kleinste getal]</b></p> <p>Ken ewekansige getal ✓in die korrekte gebied ✓ aan veranderlike toe            Vertoon getal✓ omgeskakel na string✓            As (if) ✓getal &lt; kleinste ✓                Ken getal toe aan kleinste ✓            Vertoon kleinste in edtV1_3✓ omgeskakel na string✓</p> <p>RandomRange(1,101)            Random(100) + 1            Ceil(Random() *100)</p>	9	

1.4	<p><b>Knoppie [1.4 – Vertoon gedekripteerde string]</b></p> <p>Lees toevoerstring met 'n toevoerdialoogblokkie in ✓ met korrekte parameters ✓</p> <p>Lus ('Loop') ✓ van eerste ✓ tot laaste karakter ✓</p> <p>Korrekte toets uitgevoer (if/case) ✓</p> <p>Verkry karakter by indeks ✓</p> <p>Toets teen syfer ✓ vir vir al 10 syfers ✓</p> <p>Vervang syfer by korrekte indeks ✓</p> <p>Met korrekte karakter ✓ vir al 10 karakters ✓</p> <p>Vertoon ✓</p> <p><b>KONSEPTE:</b></p> <p>Lees toevoerstring met 'n toevoerdialoogblokkie(1) met korrekte parameters (1)</p> <p>Lus ('Loop') (1) van eerste (1) tot laaste karakter (1)</p> <p>Korrekte toets (if/case) (1)</p> <p>Onttrek karakter by indeks (1)</p> <p>Toets teen syfer (1) vir al 10 syfers (1)</p> <p>Vervang syfer by korrekte indeks (1)</p> <p>Met korrekte karakter (1) vir al 10 karakters (1)</p> <p>Vertoon (1)</p>	13	
	<b>TOTAAL AFDELING A:</b>	<b>40</b>	

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	<b>SQL-stellings</b>		
2.1.1	<b>Knoppie [2.1.1 – Beste speler]</b> SELECT SpelerVan, SpelerNaam FROM tblSpelers WHERE VaardigheidsVlak = 10 <b>KONSEPTE:</b> SELECT SpelerVan, SpelerNaam ✓ FROM tblSpelers ✓ WHERE VaardigheidsVlak = 10 ✓	3	
2.1.2	<b>Knoppie [2.1.2 – B-span-afrigters]</b> SELECT Afrigter, SpanNaam FROM tblSpanne WHERE SpanNaam Like "%B" <b>KONSEPTE:</b> SELECT Afrigter, SpanNaam ✓ FROM tblSpanne ✓ WHERE SpanNaam LIKE ✓ "%B" ✓	4	
2.1.3	<b>Knoppie [2.1.3 – Persentasie wedstryde gewen]</b> SELECT SpanNaam, Afrigter, (GetalWedstrydeGewen/GetalWedstrydeGespeel*100) AS [PersentasieWedstrydeGewen] FROM tblSpanne WHERE SpanNaam = ' ' + sSpan + ' ' <b>KONSEPTE:</b> SELECT SpanNaam, Afrigter, ✓ (GetalWedstrydeGewen/GetalWedstrydeGespeel ✓ *100), ✓ AS [PersentasieWedstrydeGewen] FROM tblSpanne WHERE SpanNaam = ' ' + sSpan + ' ' ✓ <b>ALTERNATIEF:</b> 'WHERE SpanNaam = ' + QuotedStr(sSpan)	4	

2.1.4	<b>Knoppie [2.1.4 – Spangemiddeld meer as 6]</b>	5	
	<pre>SELECT SpanNaam, ROUND(AVG(VaardigheidsVlak),1) AS [GemidVaardigheidsVlak] FROM tblSpelers GROUP BY SpanNaam HAVING AVG(VaardigheidsVlak)&gt;6</pre> <p><b>KONSEPTE:</b>  SELECT SpanNaam, ROUND(AVG(VaardigheidsVlak) ✓,1)✓  AS GemidVaardigheidsVlak ✓  FROM tblSpelers  GROUP BY SpanNaam✓  HAVING AVG(VaardigheidsVlak) &gt; 6✓</p>		
2.1.5	<b>Knoppie [2.1.5 – Dateer wedstryde gewen op]</b>	3	
	<pre>UPDATE tblSpanne SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1 WHERE SpanNaam &lt;&gt; "o/14 B"</pre> <p><b>KONSEPTE:</b>  UPDATE tblSpanne ✓  SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1 ✓  WHERE SpanNaam &lt;&gt; "o/14 B"✓</p>		
	<b>Subtotaal:</b>	<b>19</b>	

2.2	<b>Databasismanipulasie deur Delphi-kode te gebruik</b>		
2.2.1	<p><b>Knoppie [2.2.1 – Junior spelers in die o/18 A-span]</b></p> <pre>AssignFile ✓ &amp; Rewrite✓ Stel tblSpelers om by eerste rekord te begin lees✓ Lus ('Loop') terwyl NOT tblSpelers.EOF✓   Toets of SpanNaam 'o/18 A' is ✓ EN     of eerste 2 karakters van IDnommer-veld &gt;=3 is✓       Skryf van en naam na lêer ✓       Vermeerder teller met 1 ✓   Gaan na volgende rekord in tblSpelers✓ Eindig lus Maak lêer toe✓ Vertoon teller van getal junior speler in byskrif ('label') ✓</pre>	11	

2.2.2	<p><b>Knoppie [2.2.2 – Afrigter en doelwagter se inligting]</b></p> <p>Lus ('Loop') terwyl NOT tblSpanne.EOF✓  Stel tblSpelers om te begin lees by eerste rekord✓  Lus ('Loop') terwyl NOT tblSpelers.EOF✓  Toets of  die SpanNaam-veld in tblSpanne gelyk is aan  SpanNaam-veld in tblSpelers ✓  EN DoelWagter = true✓  Voeg span se naam, afrigter se naam ✓  Speler se van en naam  in regte formaat by afvoerstring✓  Gaan na volgende rekord in tblSpelers✓  Vertoon afvoer in reel✓  Gaan na volgende rekord in tblSpanne✓</p> <p><b>Alternatief:</b></p> <p>Lus ('Loop') terwyl NOT tblSpelers.EOF (1)  Toets of DoelWagter = true(1)  Stel tblSpanne om te begin lees by eerste rekord(1)  Lus ('Loop') terwyl NOT tblTeams.EOF(1)  Toets of die SpanNaam-veld in tblSpanne gelyk is aan  SpanNaam-veld in tblSpelers (1)  Voeg span se naam, afrigter se naam (1)  Speler se van en naam  in regte formaat by afvoerstring(1)  Gaan na volgende rekord in tblSpanne(1)  Vertoon afvoer in reel(1)  Gaan na volgende rekord in tblSpelers(1)</p>	10	
	<b>Subtotaal:</b>	<b>21</b>	
	<b>TOTAAL AFDELING B:</b>	<b>40</b>	

**BYLAAG C****VRAAG 3: NASIENRUBRIEK - OBJEK-GEÛRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1.1	<b>Konstruktor-metode:</b> Opskrif met regte parameters en datatipes ✓ Ken naam van speler parameterwaarde toe aan fSpelerNaam ✓ Ken gewig van speler parameterwaarde toe aan fGewigVanSpeler ✓ Stel fTelling op 0 ✓	4	
3.1.2	<b>getTelling-metode:</b> Funksie opskrif met heelgetal as terugstuur-datatipe ✓ fTelling aan 'result' toegeken ✓	2	
3.1.3	<b>dateerTellingOp-metode:</b> Prosedure-opskrif met heelgetalparameter ✓ Vermeerder fTelling ✓ met parameterwaarde ✓	3	
3.1.4	<b>berekenLMI-metode:</b> Funksie verklaar met reële terugstuur datatipe en ontvang reële parameterwaarde and return 'n berekende LMI ✓ Calculation = fGewigVanSpeler / ✓ Sqr(parameterwaarde lengte van speler) ✓	3	
3.1.5	<b>geskikVirSeleksie-metode:</b> Toets of telling 'n waarde tussen 0 en 7 is (inklusief) ✓ Result = Klein moontlikheid ✓ Toets of telling 'n waarde tussen 8 en 14 is (inklusief) Result = Gemiddelde moontlikheid ✓ Toets of telling > 14 is Result = Groot moontlikheid ✓	4	
3.1.6	<b>toString-metode:</b> Byskrifte (Naam, Gewig, Huidige telling) ✓ Regte attribute ✓ Regte omskakelings (gewig – float; telling – integer) ✓ Return string ✓	4	
	<b>Subtotaal: Objekklas</b>	<b>20</b>	

**VRAAG 3: NASIENRUBRIEK (VERVOLG)**

<b>VRAAG</b>	<b>BESKRYWING</b>	<b>MAKS. PUNTE</b>	<b>LEERDER - PUNT</b>
3.2.1	<p><b>Knoppie [3.2.1 - Instansieer speler-objek]</b></p> <p><i>Instansieer die objSpeler-objek:</i>  objSpeler ✓ := TSpeler.Create ✓  Stuur naam en spelergewig in korrekte volgorde ✓  met korrekte datatype (dieselfde as konstruktore) ✓  Gebruik 'n dialoogboks om aan te dui dat die objek ge-  instansieer is ✓</p>	<b>5</b>	
3.2.2	<p><b>Knoppie [3.2.2 – Bereken LMI]</b></p> <p>Gebruik toevoerdialoogblokkie ✓ om lengte in te sleutel  Roep berekenLMI-metode deur gebruik van objek ✓  gebruik lengte as argument ✓ omgeskakel na reële getal  Vertoon die inligting in <b>redV3_2_2</b> ✓  deur die toString-metode van die objek te gebruik ✓  Vertoon die LMI in <b>redV3_2_2</b> ✓  in die regte formaat met een desimale plek ✓</p>	<b>7</b>	
3.2.3	<p><b>Knoppie [3.2.3 – Dateer telling op]</b></p> <p>Kry itemindex uit die radio group ✓  Om die telling te kry ✓  Roep die dateerTellingOp metode ✓  met korrekte argument ✓  Vertoon opgedateerde telling op pnIV3_2_3 ✓  deur die getTelling-metode te gebruik ✓</p>	<b>6</b>	
3.2.4	<p><b>Knoppie [3.2.4 – Geskik vir seleksie]</b></p> <p>Roep die relevante metode as volg:  Vertoon op die byskrif ('label') <b>IbIV3_2_4</b> ✓  Deur gebruik te maak van objSpeler.geskikVirSeleksie ✓</p>	<b>2</b>	
	<b>Subtotaal: Form-klas</b>	<b>20</b>	
	<b>TOTAAL AFDELING C:</b>	<b>40</b>	

**VRAAG 4: NASIENRUBRIEK – PROBLEEMOPLOSSING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
4.1	<b>Knoppie [4.1 – Vertoon Doolhof]</b> Kies doolhof uit kombinasieblokkie ✓ Regte lêernaam + '.txt' ✓ Toets en vertoon 'n boodskap indien die lêer nie bestaan nie ✓ Assign file, Reset die tekslêer ✓ Initialiseer indeks-veranderlike ✓ Lus (Loop) deur die tekslêer ✓ Inkrementeer die waarde van indeks ✓ (Posisie word bepaal deur beginwaarde) Lees reël ✓ ken die reël toe aan die skikking arrDoolhof ✓ Vertoon die reëlnommer ✓ en die reël ✓	11	

4.2	<p><b>Knoppie [4.2 – Langste gang]</b></p> <p>Inisialieer veranderlike na 0 om maksimum waarde te stoor✓  Lus (Loop) deur die skikking✓  Initialiseer veranderlike vir langste gang in die reël ✓  Initialiseer veranderlike om agtereenvolgende '-' in reël te tel✓  Lus (Loop) deur elke karakter✓ in die reël ✓  Toets of karakter 'n '-' is ✓  Inkrementeer dash-teller✓  As dash-teller meer is as langste gang in reël ✓  Vervang langste gang met dash-teller✓  Anders as karakter nie '-' is nie✓  Stel dash-teller na 0✓  Stoor langste gang in die ry in arrRye skikking/ 'n string✓  As langste in ry langer is as langste ✓ – vervang langste ✓  Vertoon boodskap met lengte van langste gang✓  Lus (Loop) deur skikking/string met langste gange per reël ✓  Vertoon reëlnommer van rye ✓  wat dieselfde as langste gang se lengte is✓</p> <p><b>KONSEPTE:</b>  Inisialiseer veranderlike vir langste gang na 0 (1)  Lus deur skikking (1)  Initialiseer dash-teller per reël(1)  en maksimum aantal aaneenlopende dashes per reël (1)  Lus (loop) deur elke karakter in die reël (2)  Toets of karakter dash is (1)  Vermeerder die dash-teller (1)  Toets of die dash-teller langste in die reël (2)  en stoor in maksimum aantal(1)  As karakter nie dash is nie (1)  Herstel dash-teller na 0 (1)</p> <p>Bepaal die langste gang in al die lyne in die doolhof  (Toets en stoor maksimum aantal) (2)  Vertoon langste (1)  Toets vir lyne wat selfde lengte het as langste gang (2)  Vertoon lyn nommer(s) (1)</p>	19	
	<b>TOTAAL AFDELING D:</b>	<b>30</b>	
	<b>GROOTTOTAAL:</b>	<b>150</b>	

**OPSOMMING VAN LEERDER SE PUNTE:**

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

**BYLAAG E: OPLOSSING VIR VRAAG 1**

```
unit Vraag1_u;

interface

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

type
  TfrmV1 = class(TForm)
    gpbV1_1: TGroupBox;
    gpbV1_4: TGroupBox;
    gpbV1_3: TGroupBox;
    btnV1_1: TButton;
    btnV1_4: TButton;
    edtV1_3: TEdit;
    btnV1_3: TButton;
    redV1_3: TRichEdit;
    btnReset: TButton;
    gpbV1_2: TGroupBox;
    btnV1_2: TButton;
    Label1: TLabel;
    pnlV1_2: TPanel;
    lblV1_1: TLabel;
    lbl1_2Pies: TLabel;
    edtV1_2Deursnee: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    edtV1_2Breedte: TEdit;
    Label4: TLabel;
    spnV1_1: TSpinEdit;
    Label5: TLabel;

    procedure btnResetClick(Sender: TObject);
    procedure btnV1_1Click(Sender: TObject);
    procedure btnV1_2Click(Sender: TObject);
    procedure btnV1_3Click(Sender: TObject);
    procedure btnV1_4Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);

  private
    { Private declarations }
  public
    iKleinste: integer; // global
    { Public declarations }
  end;

var
  frmV1: TfrmV1;

implementation

{$R *.dfm}
```

## =====

**Vraag 1.1 = 8 punte**

```
=====
procedure TfrmV1.btnV1_1Click(Sender: TObject);
const
  PRYS = 14.95;
var
  iGetal: integer;
  rBedrag: real;
begin
  lblV1_1.Font.Size := 20;
  iGetal := spnV1_1.Value;
  rBedrag := iGetal * PRYS;
  lblV1_1.Caption := FloatToStrF(rBedrag, ffCurrency, 10, 2);
end;
```

## =====

**Vraag 1.2 = 10 punte**

```
=====
procedure TfrmV1.btnV1_2Click(Sender: TObject);
var
  rA, rB, rC : real;
begin
  rA := 4;
  rB := StrToFloat(edtV1_2.Text);
  rC := Sqrt(Sqr(rA) + Sqr(rB));
  pnlV1_2.Caption := FloatToStrF(rC, ffFixed, 8, 1);
end;
```

## =====

**Vraag 1.3 = 9 punte**

```
=====
procedure TfrmV1.btnV1_3Click(Sender: TObject);
var // Kode wat voorsien is
  iGetal: integer;

begin
  iGetal := random(100) + 1;
  redV1_3.lines.Add(IntToStr(iGetal));
  if iGetal < iKleinste then
    iKleinste := iGetal;
  edtV1_3.Text := IntToStr(iKleinste);
end;
```

## =====

**Vraag 1.4 = 13 punte**

```
=====
procedure TfrmV1.btnV1_4Click(Sender: TObject);
const
  ALFA = 'ABCDEFGHJIJ';

var // Kode wat voorsien is
    sString, sVertoon: String;
    I, iIndeks: integer;

begin
  // Kode wat voorsien is
  sString := InputBox('Sleutel 'n geënkripteerde string in: ', '', '');
  sVertoon := '';
  for I := 1 to Length(sString) do
    if sString[I] in ['0' .. '9'] then
      begin
        iIndeks := StrToInt(sString[I]) + 1;
        sVertoon := sVertoon + ALFA[iIndeks];
      end
    else
      sVertoon := sVertoon + sString[I];
  ShowMessage(sVertoon);

end;
```

## =====

**Kode wat voorsien is**

```
=====
procedure TfrmV1.btnResetClick(Sender: TObject);
begin
  // Kode wat voorsien is - moenie verander nie
  iKleinste := 100;
  redV1_3.Clear;
  edtV1_3.Clear;
end;

procedure TfrmV1.FormActivate(Sender: TObject);
begin
  // Kode wat voorsien is - moenie verander nie
  iKleinste := 100;
  redV1_3.Clear;
  edtV1_3.Clear;
  CurrencyString := 'R';
end;

end.
```

**BYLAAG F: OPLOSSING VIR VRAAG 2****VRAAG 2.1: SQL kode**

```
=====
Vraag 2.1: 2.1.1(3), 2.1.2(4), 2.1.3(4), 2.1.4(5), 2.1.5 (3)
=====
```

```
2.1.1: SELECT SpelerVan, SpelerNaam
        FROM tblSpelers WHERE VaardigheidsVlak = 10

2.1.2: SELECT Afrigter, SpanNaam
        FROM tblSpanne
        WHERE SpanNaam Like "%B"

2.1.3: SELECT SpanNaam, Afrigter,
        (GetalWedstrydeGewen/GetalWedstrydeGespeel*100) AS
        [PersentasieWedstrydeGewen]
        FROM tblSpanne
        WHERE SpanNaam = "' + sSpan + '"

2.1.4: SELECT SpanNaam, round(avg(VaardigheidsVlak),1) AS
        [GemidVaardigheidsVlak]
        FROM tblSpelers
        GROUP BY SpanNaam HAVING avg(VaardigheidsVlak) > 6

2.1.5: UPDATE tblSpanne
        SET GetalWedstrydeGewen = GetalWedstrydeGewen + 1
        WHERE SpanNaam <> "o/14 B"
```

**QUESTION 2.2: DATABASE MANIPULATION met gebruik van Delphi-kode**

```
// {$REGION 'VRAAG 2.2'}
```

```
=====
Vraag 2.2.1 = 11 punte
=====
```

```
procedure TfrmDBVraag2.btnV2_2_1Click(Sender: TObject);
var
    tFile: textfile;
    iTel, iJaar: integer;
begin
    AssignFile(tFile, 'Junior18A.txt');
    Rewrite(tFile);
    tblSpelers.first;
    iTel := 0;
    while not tblSpelers.eof do
    begin
        iJaar := StrToInt(copy(tblSpelers['IDNommer'], 1, 2));
        if (tblSpelers['SpanNaam'] = 'o/18 A') AND (iJaar >= 3) then
        begin
            Writeln(tFile,
                tblSpelers['SpelerVan'] + ' ' + tblSpelers['SpelerNaam']);
            inc(iTel);
        end;
        tblSpelers.Next;
    end;
end;
```

```

CloseFile(tFile);
lblV2_2_1.Caption := 'Getal junior spelers in die o/18 A-span: ' +
IntToStr(iTel);

// Kode wat voorsien is
dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

=====
Vraag 2.2.2 = 10 punte
=====
procedure TfrmDBVraag2.btnV2_2_2Click(Sender: TObject);
var
    sAfvoer: String; // veranderlike om span wat geselekteer is te stoor
begin // Vraag 2.2.2
    // Kode wat voorsien is
    redV2_2_2.Clear;
    redV2_2_2.Paragraph.TabCount := 2;
    redV2_2_2.Paragraph.Tab[0] := 70;
    redV2_2_2.Paragraph.Tab[1] := 160;
    redV2_2_2.Lines.Add('Span' + #9 + 'Afrigter' + #9 + 'Doelwagter');

    // -----
    // Sleutel jou kode hier in:

    tblSpanne.first;
    while NOT tblSpanne.eof do
    begin
        sAfvoer := tblSpanne['SpanNaam'] + #9 + tblSpanne['Afrigter'] + #9;
        tblSpelers.first;
        while NOT tblSpelers.eof do
        begin
            if (tblSpanne['SpanNaam'] = tblSpelers['SpanNaam']) AND
                (tblSpelers['DoelWagter'] = true) then
            begin
                sAfvoer := sAfvoer + tblSpelers['SpelerVan'] + ', ' + tblSpelers
                    ['SpelerNaam'];
            end;
            tblSpelers.Next;
        end; tblSpanne.Next;
        redV2_2_2.Lines.Add(sAfvoer);
    end;

    // Kode wat voorsien is
    dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

```

```
=====
{$REGION 'Provided code: Setup DB connections - DO NOT CHANGE!'}
=====
procedure TfrmDBVraag2.bmbHerstelDBClick(Sender: TObject);
begin
  // Herstel die database
  dbCONN.RestoreDatabase;
  redV2_2_2.Clear;
  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
end;

// =====
procedure TfrmDBVraag2.FormClose(Sender: TObject;

  var Action: TCloseAction);
begin // Diskonnekteer databasis en maak alle oop konneksies toe
  dbCONN.dbDisconnect;
end;

procedure TfrmDBVraag2.FormCreate(Sender: TObject);
begin
  CurrencyString := 'R';
end;

// =====
procedure TfrmDBVraag2.FormShow(Sender: TObject);
begin // Stel konneksie met databasis op en maak die tabelle oop.
  dbCONN := TConnection.Create;
  dbCONN.dbConnect;
  tblSpanne := dbCONN.tblOne;
  tblSpelers := dbCONN.tblMany;

  dbCONN.setupGrids(dbgrdONE, dbgrdMany, dbgrdSQL);
  pgcDBAdmin.ActivePageIndex := 0;
end;
// =====
// {$ENDREGION}

end.
```

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

```

unit Speler_U;

interface

uses StdCtrls, SysUtils;

type
  TSpeler = class(TObject)

    //Kode wat voorsien is - moenie verander nie
  private
    fSpelerNaam: String;
    fGewigVanSpeler : real;
    fTelling : integer;

  public
    //Kode wat voorsien is - moenie verander nie
    constructor create(sSpelerNaam : String; rGewigVanSpeler : real);
    function getTelling : integer;

    //Voeg kode hier in
    function berekenLMI (rLengteVanSpeler : real) : real;
    procedure dateerTellingOp (iTelling : integer);
    function geskikVirSeleksie: String;
    function toString : String;
end;

```

```

implementation

```

```

{ TSpeler }

```

```

=====
Vraag 3.1.1 = 4 punte
=====

```

```

constructor TSpeler.create(sSpelerNaam : String; rGewigVanSpeler :
real);
begin
  fSpelerNaam := sSpelerNaam;
  fGewigVanSpeler := rGewigVanSpeler;
  fTelling := 0;
end;

```

```

=====
Vraag 3.1.2 = 2 punte
=====

```

```

function TSpeler.getTelling: integer;
begin
  result := fTelling;
end;

```

```

=====
Vraag 3.1.3 = 3 punte
=====

```

```

procedure TSpeler.dateerTellingOp(iTelling: integer);
begin
  fTelling := fTelling + iTelling;
end;

```

=====  
**Vraag 3.1.4 = 3 punte**  
=====

```
function TSpeler.berekenLMI(rLengteVanSpeler: real) : real;
begin
  result := fGewigVanSpeler / sqr(rLengteVanSpeler);
end;
```

=====  
**Vraag 3.1.5 = 4 punte**  
=====

```
function TSpeler.geskikVirSeleksie:string;
begin
  if fTelling < 8 then
    result := 'Klein moontlikheid'
  else
    if fTelling < 15 then
      result := 'Gemiddelde moontlikheid'
    else
      result := 'Groot moontlikheid'
end;
```

=====  
**Vraag 3.1.6 = 4 punte**  
=====

```
function TSpeler.toString : String;
begin
  result := 'Naam: ' + fSpelerNaam + #13 + 'Gewig: '+
    FloatToStr(fGewigVanSpeler)+ #13+ 'Huidige telling: ' +
    IntToStr (fTelling);
end;

end.
```

**Hoofvormeenheid**

```
unit Vraag3_U;

interface

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

type
  TfrmVraag3 = class(TForm)
    lblNaamVanSpeler: TLabel;
    edtNaamVanSpeler: TEdit;
    btnV3_2_2: TButton;
    redV3_2_2: TRichEdit;
    pnlV3_2_3: TPanel;
    btnV3_2_3: TButton;
    btnV3_2_1: TButton;
    rgpV3_2_3: TRadioGroup;
    btnV3_2_4: TButton;
    lblV3_2_4: TLabel;
    Label1: TLabel;
    edtGewigVanSpeler: TEdit;
    Label3: TLabel;
    grpV3_2_1: TGroupBox;
    grpV3_2_2: TGroupBox;
    grpV3_2_3: TGroupBox;
    grpV3_2_4: TGroupBox;
    procedure btnV3_2_2Click(Sender: TObject);
    procedure btnV3_2_3Click(Sender: TObject);
    procedure btnV3_2_1Click(Sender: TObject);
    procedure btnV3_2_4Click(Sender: TObject);
    // procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag3: TfrmVraag3;
  // Veranderlike wat reeds verklaar is
  objSpeler : TSpeler;
implementation

{$R *.dfm}
```

## =====

**Vraag 3.2.1 = 5 punte**

```
=====
procedure TfrmVraag3.btnV3_2_1Click(Sender: TObject);
var
  sNaamVanSpeler : String;
  rGewigVanSpeler : real;
begin
  sNaamVanSpeler := edtNaamVanSpeler.Text; rGewigVanSpeler :=
  StrToFloat(edtGewigVanSpeler.Text); objSpeler :=
  TSpeler.create(sNaamVanSpeler,rGewigVanSpeler);
  ShowMessage('Speler-objek is ge-instansieer.');
```

## =====

**Vraag 3.2.2 = 7 punte**

```
=====
procedure TfrmVraag3.btnV3_2_2Click(Sender: TObject);
var
  rLengteVanSpeler, rLMI : real;
  sLMI : String;

begin
  redV3_2_2.Clear;
  rLengteVanSpeler := StrToFloat(InputBox('Lengte van die speler',
  'Sleutel die lengte van die speler in meter in: ',''));
  rLMI := objSpeler.berekenLMI(rLengteVanSpeler);
  sLMI := FloatToStrF(rLMI,ffFixed,3,1);
  redV3_2_2.Lines.Add(objSpeler.toString);
  redV3_2_2.Lines.Add ('LMI: '+ sLMI);
end;
```

## =====

**Vraag 3.2.3 = 6 punte**

```
=====
procedure TfrmVraag3.btnV3_2_3Click(Sender: TObject);
var
  iTelling: integer;
begin
  iTelling := StrToInt(rgpV3_2_3.Items[rgpV3_2_3.ItemIndex]);
  objSpeler.dateerTellingOp(iTelling);
  pnlV3_2_3.Caption := ('Opgedateerde telling: ' +
  IntToStr(objSpeler.getTelling));
end;
```

## =====

**Vraag 3.2.4 = 2 punte**

```
=====
procedure TfrmVraag3.btnV3_2_4Click(Sender: TObject);
begin
  lblV3_2_4.Caption := objSpeler.geskikVirSeleksie;
end;
end.
```

**BYLAAG H: OPLOSSING VIR VRAAG 4**

```
unit Vraag4_U;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ComCtrls;
type
  TfrmV4 = class(TForm)
    gpbVrae: TGroupBox;
    btnV4_1: TButton;
    btnV4_2: TButton;
    cmbV4_1: TComboBox;
    Label1: TLabel;
    redV4: TRichEdit;
    procedure btnV4_1Click(Sender: TObject);
    procedure btnV4_2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  // Kode wat voorsien is
const
  iRyTel = 16;

var
  frmV4: TfrmV4;
  arrDoolhof: array [1..iRyTel] of String;

implementation
{$R *.dfm}
```

```
=====
                          Vraag 4.1 = 11 punte
=====
```

```
procedure TfrmV4.btnV4_1Click(Sender: TObject);
var
  iIndeks: integer;
  sLeerNaam, sReel: String;
  tNaamLeer: TextFile;
begin
```

```
=====
                          Kode wat voorsien is
=====
```

```
redV4.Clear;
```

```
=====
                          Kode wat voltooi moet word
=====
```

```
iIndeks := 1;
sLeerNaam := cmbV4_1.Text + '.txt';
if FileExists(sLeerNaam) then
begin
  AssignFile(tNaamLeer, sLeerNaam);
  Reset(tNaamLeer);
```

```

while not (eof(tNaamLeer)) do
begin
  Readln(tNaamLeer, sReel); arrDoolhof[indeks] :=
  sReel; redV4.Lines.Add(IntToStr(iIndeks) + #9 +
  sReel); Inc(iIndeks);
end;
end;
end;

```

---

**Vraag 4.2 = 19 punte**

---

```

procedure TfrmV4.btnV4_2Click(Sender: TObject);
var
  iRy, iKol, iTel, iLangsteInReel, iMaks: integer;
  arrRye: array [1..iRyTel] of integer;
  sRy: String;
begin
  iMaks := 0;
  for iRy := 1 to iRyTel do
  begin
    sRy := arrDoolhof[iRy];

    iLangsteInReel := 0;
    iTel := 0;
    for iKol := 1 to Length(sRy) do
    begin
      if sRy[iKol] = '-' then
      begin
        Inc(iTel);
        if iTel > iLangsteInReel then
          iLangsteInReel := iTel
        end
      else
        iTel := 0;
      end;
    end;
    arrRye[iRy] := iLangsteInReel;

    if iMaks < arrRye[iRy] then
      iMaks := arrRye[iRy];
    end;

    redV4.Lines.Add('');
    redV4.Lines.Add('Langste gang(e) met ' + IntToStr(iMaks)
    + ' spasies in ry(e):');
    for iRy := 1 to Length(arrRye) do
    begin
      if arrRye[iRy] = iMaks then
        redV4.Lines.Add(' ' + IntToStr(iRy));
      end;
    end;
  end;
end.

```