

Vertroulik



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2025

PUNTE: 150

TYD: 3 uur

**Hierdie vraestel bestaan uit 31 bladsye, 2 bladsye met data,
2 bladsye vir beplanning en 'n aparte inligtingsblad.**

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. 'n Inligtingsblad is voorsien wat jy aan die einde van die eksamensessie moet voltooi. Maak seker dat AL die inligting wat jy verskaf het, korrek is en lewer die inligtingsblad in voordat jy die eksamenlokaal verlaat.
4. 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.
5. Hierdie vraestel is opgestel met programmeringsterme wat kenmerkend vir Delphi as programmeringstaal is. Die Delphi-programmeringstaal moet gebruik word om die vrae te beantwoord.
6. Maak seker dat jy die vrae beantwoord volgens die spesifikasies wat in elke vraag gegee word. Punte sal volgens die voorgeskrewe vereistes toegeken word.
7. Beantwoord slegs wat in elke vraag gevra word. Byvoorbeeld, indien die vraag nie vir datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
8. 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.
9. 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.
10. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture verskaf word.
11. 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.
12. Maak seker dat jou eksamennommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
13. 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.

14. 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...
15. 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 wagwoordbeskermdes uitvoerbare lêers verskaf.

Doen die volgende:

- Dubbelklik op die volgende wagwoordbeskermdes uitvoerbare lêer:
DataNov2025.exe
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **SuSTF@rm2025**

Nadat dit onttrek ('extracted') is, sal die volgende lys lêers in die lêergids ('folder') **DataNov2025** beskikbaar wees:

Question 1:

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

Question 2:

ConnectDB_U.pas
FarmManagementDB - Copy.mdb
FarmManagementDB.mdb
MixedFarms.txt
Question2_P.dpr
Question2_P.dproj
Question2_P.res
Question2_U.dfm
Question2_U.pas

Question 3:

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

Question 4:

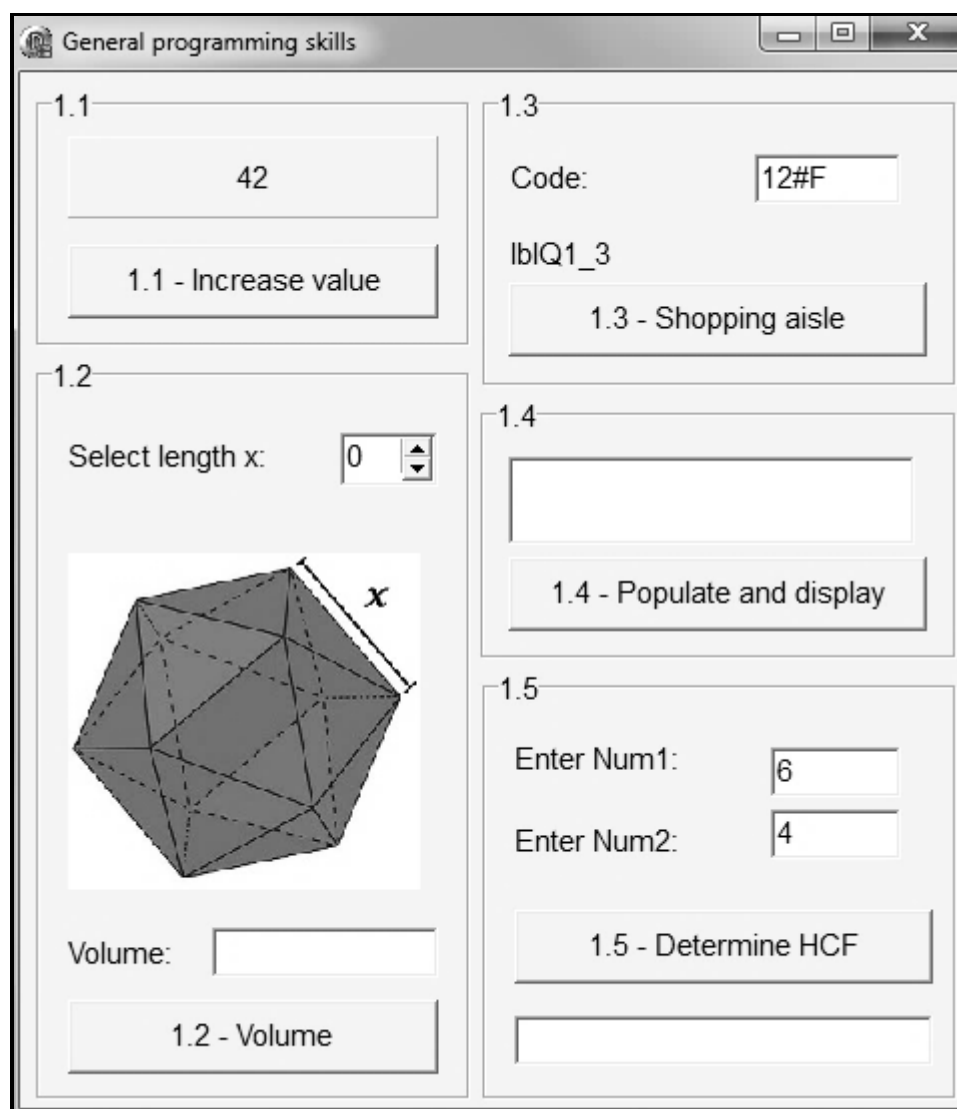
Question4_P.dpr
Question4_P.dproj
Question4_P.res
Question4_U.dfm
Question4_U.pas

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 in VRAAG 1.1 tot VRAAG 1.5 beskryf is.

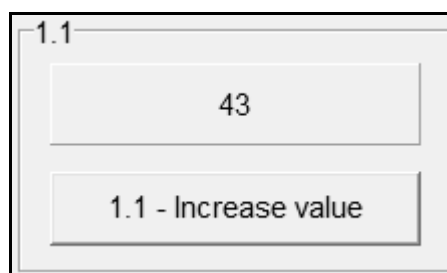
1.1 Knoppie [1.1 - Increase value]

Die paneel **pnIQ1_1** vertoon tans 'n heelgetalwaarde wat met die waarde van 1 moet toeneem wanneer die knoppie geklik word.

Skryf kode om die volgende te doen:

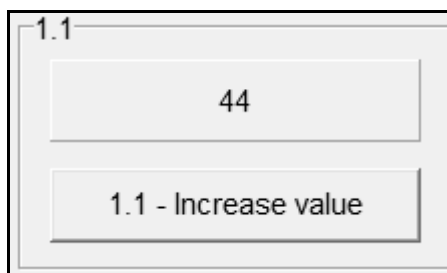
- Onttrek ('Extract') die heelgetalwaarde wat op die paneel **pnIQ1_1** vertoon word.
- Vermeerder die waarde wat onttrek is, met die waarde van 1.
- Vertoon die waarde wat toegeneem het op die paneel **pnIQ1_1**.

Voorbeeld van afvoer nadat die knoppie geklik is en die waarde wat onttrek is, met die waarde van 1 toegeneem het:



LET WEL: Elke keer wanneer die gebruiker op die knoppie 'Increase value' klik, moet die waarde wat tans op die paneel vertoon word, met die waarde van 1 toeneem.

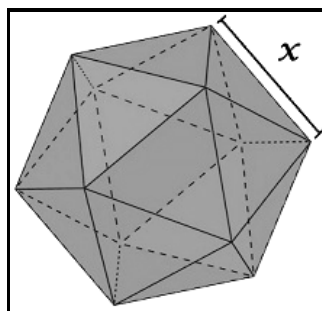
Voorbeeld van afvoer nadat die knoppie vir die tweede keer geklik is:



(4)

1.2 Knoppie [1.2 - Volume]

Die prent hieronder toon 'n figuur van 'n driedimensionele vorm wat sye van gelyke lengte het. Die lengte van elke sy word deur die letter x verteenwoordig.



Die formule om die volume van die figuur te bereken, is soos volg:

$$Volume = \frac{5(3+\sqrt{5})}{12} x^3$$

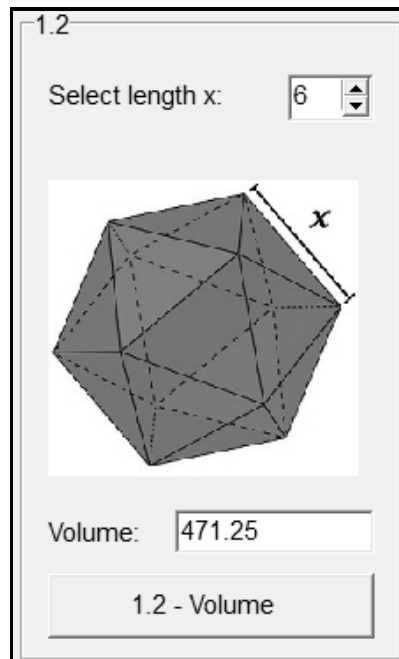
waar x die lengte van die sye van die figuur verteenwoordig.

Die gebruiker moet die lengte van x vanaf die spinredigeerder ('spin edit') **spnQ1_2** selekteer.

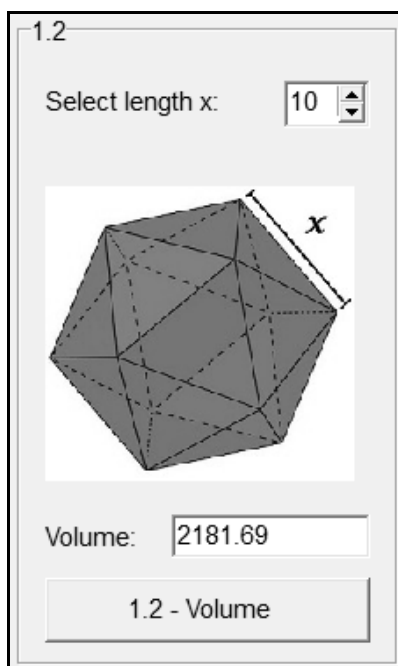
Skryf kode om die volgende te doen:

- Onttrek ('Extract') die lengte van die sy (x) uit die spinredigeerder **spnQ1_2**.
- Bereken die volume van die figuur deur die gegewe formule en ten minste TWEE VERSKILLENDE voorafgedefinieerde wiskundige funksies te gebruik.
- Vertoon die volume in die redigeerblokkie ('edit box') **edtQ1_2**, tot TWEE desimale plekke geformateer.

Voorbeeld van afvoer indien die waarde van 6 as die lengte van x geselekteer is:



Voorbeeld van afvoer indien die waarde van 10 as die lengte van x geselekteer is:



(6)

1.3 Knoppie [1.3 - Shopping aisle]

'n Plaaslike kruidenierswinkel gebruik 'n koderingstelsel om klante te help om gange ('aisles') met spesifieke voedselkategorieë te vind. Elke voedselkategorie word deur 'n enkele alfabetletter voorgestel. Die alfabetletters en die voedselkategorieë wat deur die letters voorgestel word, word in die tabel hieronder gegee.

ALFABETLETER	KATEGORIEBESKRYWING
F	Fruit and vegetables
D	Dairy
B	Butchery
S	Sauces
P	Pasta and rice

Die gebruiker moet 'n kode in die redigeerblokkie ('edit box') **edtQ1_3** in die volgende formaat insleutel:

<Aisle number>#<Alphabet letter>

Voorbeeld: 12#F verwys na gang ('aisle') 12 waar die vrugte- en groente-voorraad gehou word.

Skryf kode om die volgende te doen:

- Onttrek die kode wat ingesleutel is uit die redigeerblokkie **edtQ1_3**.
- Skei die gangnommer en die alfabetletter wat die voedselkategorie verteenwoordig van die kode wat ingesleutel is.
- Gebruik 'n **gevalstelling** ('case statement') en die alfabetletter van die kode en stoor die korrekte kategoriebeskrywing in 'n veranderlike.
- Vertoon 'Aisle', die nommer van die gang ('aisle'), 'n dubbelpunt-karakter (:), gevolg deur 'n spasie en die korrekte kategoriebeskrywing op byskrif ('label') **lblQ1_3** in die formaat:

Aisle <Aisle number>: <Category description>

Voorbeeld van afvoer indien die kode 12#F ingesleutel is:

The screenshot shows a window titled '1.3' with a 'Code:' label and a text box containing '12#F'. Below this, it displays 'Aisle 12: Fruit and vegetables'. At the bottom, there is a button labeled '1.3 - Shopping aisle'.

Voorbeeld van afvoer indien die kode 3#S ingesleutel is:

The screenshot shows a window titled '1.3' with a 'Code:' label and a text box containing '3#S'. Below this, it displays 'Aisle 3: Sauces'. At the bottom, there is a button labeled '1.3 - Shopping aisle'.

LET WEL: Jy kan aanneem dat die kode wat ingesleutel is, geldig is. (11)

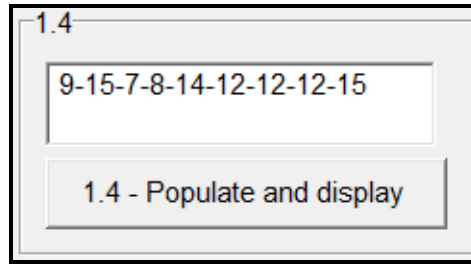
1.4 Knoppie [1.4 - Populate and display]

Die verklaring van 'n skikking met die naam **arrNumbers**, wat uit NEGE elemente van tipe heelgetal bestaan, is voorsien.

Skryf kode om die volgende te doen:

- Ken 'n ewekansige ('random') getal in die reeks van 1 tot 15 (ingesluit) aan elke element van skikking **arrNumbers** toe.
- Gebruik die memokomponent **memQ1_4** om die ewekansige getalle uit skikking **arrNumbers**, deur aandagstreep(-)karakters geskei, te vertoon.

Voorbeeld van afvoer:

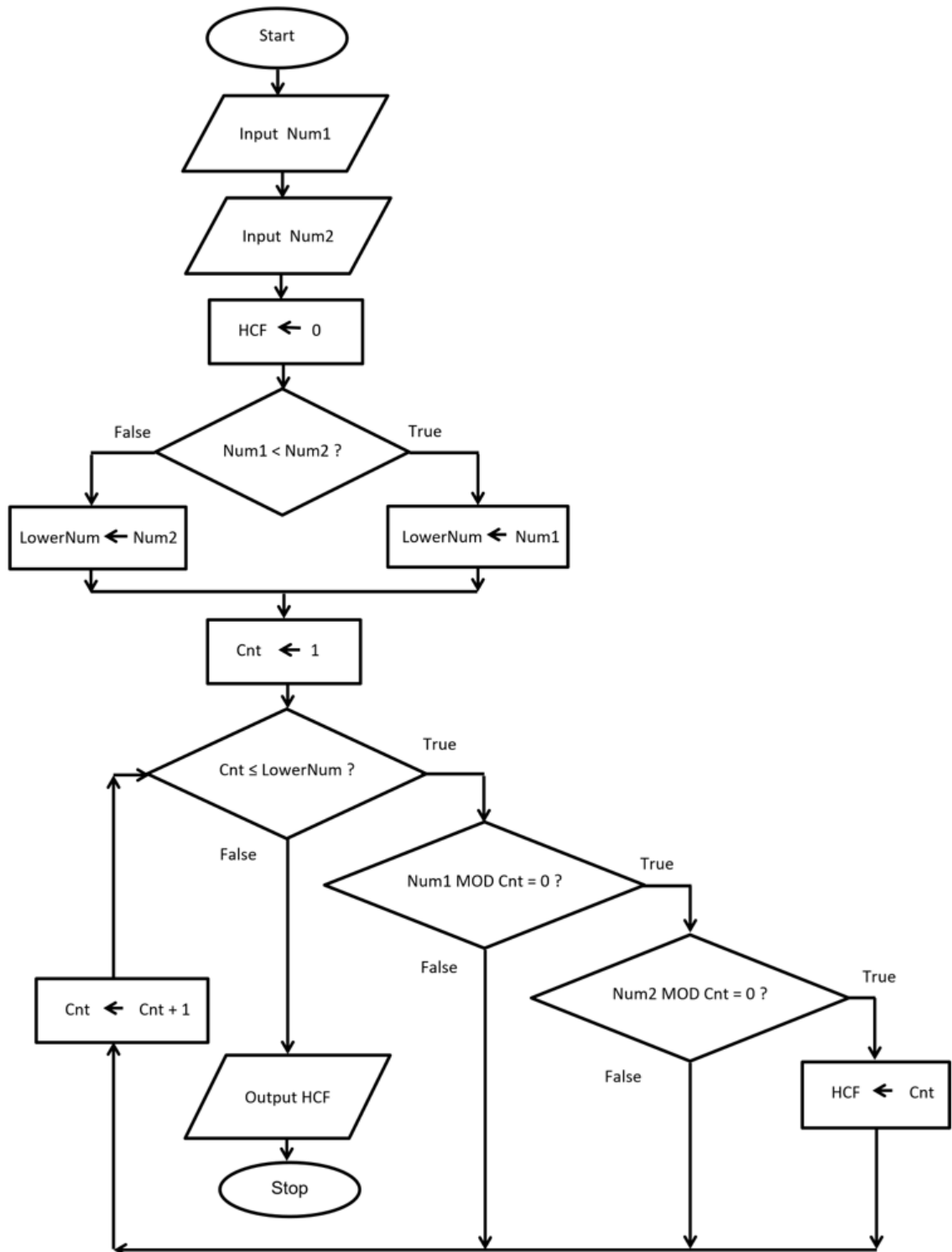


LET WEL: Die afvoer wat deur jou program vertoon word, kan van die afvoervoorbeeld verskil aangesien die getalle ewekansig gegenerer is.

(8)

1.5 **Knoppie [1.5 - Determine HCF]**

Die vloeiagram hieronder illustreer hoe om die grootste gemene faktor (GGF) ('highest common factor (HCF)') van enige TWEE positiewe heelgetalle te bepaal.



Die gebruiker moet heelgetalwaardes vir Num1 en Num2 insleutel.

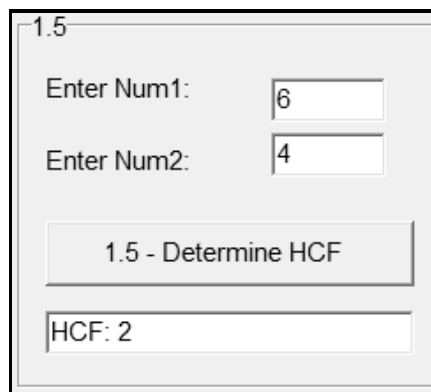
Kode is voorsien om:

- Die twee heelgetalwaardes Num1 en Num2 te onttrek en onderskeidelik in **iNum1** en **iNum2** te stoor, en
- **iHCF** na nul (0) te inisialiseer

Skryf kode om die volgende te doen:

- Kodeer die stappe wat in die vloediagram voorsien is om die GGF ('HCF') van die twee getalle wat ingesleutel is, te bepaal.
- Vertoon die GGF ('HCF') in redigeerblokkie ('edit box') **edtQ1_5**.

Voorbeeld van afvoer indien die getalle 6 en 4 ingesleutel is:

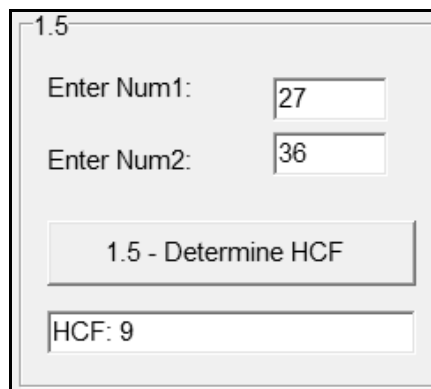


1.5

Enter Num1:

Enter Num2:

Voorbeeld van afvoer indien die getalle 27 en 36 ingesleutel is:



1.5

Enter Num1:

Enter Num2:

(11)

- 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 A: 40

AFDELING B**VRAAG 2: DATABASISPROGRAMMERING**

'n Diensverskaffer moet 'n versameling klein plase met verskillende boerdertipes, soos pluimvee ('Poultry'), lewende hawe ('Livestock') en suiwel ('Dairy'), bestuur. Elke plaas word deur slegs een ('n enkele) eienaar besit, terwyl een plaaseienaar baie plase kan besit.

'n Databasis met die naam **FarmManagementDB.mdb** is geskep en bevat twee tabelle met die name **tblFarmOwners** en **tblFarms**.

Die bladsye met data wat aan die einde van die vraestel aangeheg is, verskaf inligting oor die ontwerp en die 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 eksamennummer as kommentaar in die eerste reël van die **Question2_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit. Die inhoud van die tabelle word vertoon, soos wat hieronder op die seleksie van die oortjieblad ('tab sheet') **2.2 - Delphi code** getoon word.

OwnerID	FullName	ContactNumber	Email	DateOfBirth
101	John Smith	081 555 1201	john.smith@email.com	1/12/2002
102	Maria van Wyk	072 444 5622	maria.vw@email.com	5/20/1994
103	Themba Dlamini	083 678 9132	themba.d@email.com	2/14/1992
104	Sipho Mthembu	061 789 6512	sipho.m@email.com	9/3/1991
105	Fatima Khan	074 321 9043	fatima.k@email.com	1/1/2000
106	David Botha	065 543 2132		2/12/1992
107	Nkosi Zulu	082 111 7624	nkosi.z@email.com	2/18/1989
108	Peter van der Merwe	076 234 8132	peter.vdm@email.com	7/27/1995
109	Johann van Rensburg	071 899 1235	johann.vr@email.com	4/28/2003

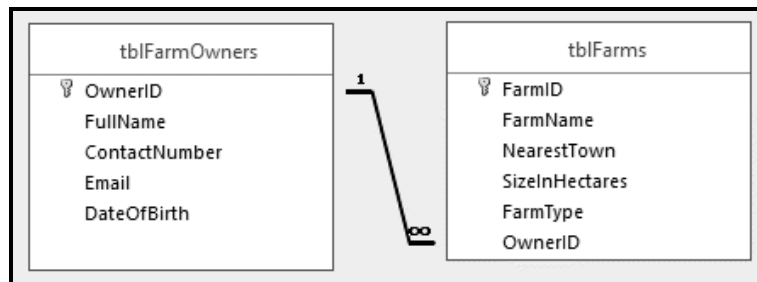
FarmID	FarmName	NearestTown	SizeInHectares	FarmType	OwnerID
201	Green Pastures	Pietermaritzburg	85	Dairy	101
202	Golden Fields	Stellenbosch	120	Poultry	102
203	Riverbank Farm	Pretoria	95	Livestock	103
204	Blue Horizon Farm	Polokwane	150	Poultry	104
205	Sunrise Dairy	Rustenburg	200	Dairy	105
206	Orchard View	Queenstown	75	Livestock	106
207	Highland Meadow	Bloemfontein	90	Livestock	114
208	Valley Creek Farm	Nelspruit	110	Livestock	108
209	Sunny Ridge	Durban	95	Poultry	109
210	Golden Valley	Paarl	130	Dairy	110

Farms on list that do not qualify:

2.2 - Mixed-farm type

Restore database Close

Die verwantskap tussen die twee tabelle **tblFarmOwners** en **tblFarms** word hieronder getoon.



- Volg die instruksies hieronder om die kode vir elke afdeling 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.

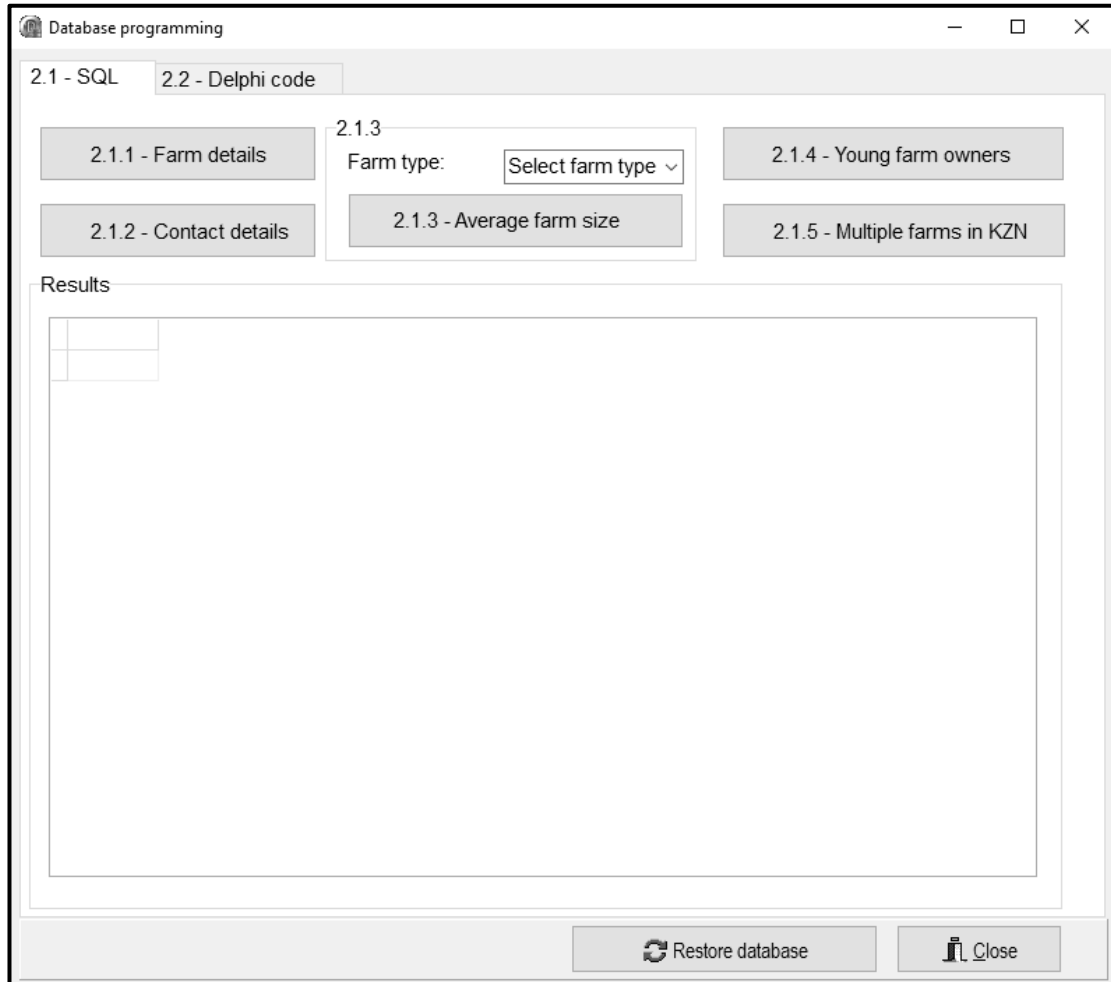
LET WEL:

- Die 'Restore database'-knoppie word voorsien om die data wat in die databasis voorkom, na die oorspronklike inhoud te herstel.
- Kode word voorsien om die GGK('GUI')-komponente met die databasis te koppel. MOENIE enige van die kode wat voorsien is, verander NIE.
- TWEE veranderlikes word as globale veranderlikes verklaar, soos in die tabel hieronder beskryf word.

Veranderlike	Datatype	Beskrywing
tblFarmOwners	TADOTable	Verwys na die tabel tblFarmOwners
tblFarms	TADOTable	Verwys na die tabel tblFarms

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

Voorbeeld van die grafiese gebruikerskoppelvlak (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 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 - Farm details]

Vertoon die **FarmName**, **NearestTown** en **SizeInHectares** van al die plase, wat in dalende orde volgens **SizeInHectares** gesorteer is.

Voorbeeld van afvoer van die eerste drie rekords:

FarmName	NearestTown	SizeInHectares
Mountain Dew	Rustenburg	312
Blue Crane	Pietermaritzburg	212
Green Pastures	Potchefstroom	211

(3)

2.1.2 Knoppie [2.1.2 - Contact details]

Vertoon die **FullName** en **ContactNumber** van alle plaaseienaars wie se e-posadres nie in die tabel **tblFarmOwners** gestoor is nie.

Voorbeeld van afvoer:

FullName	ContactNumber
David Botha	065 543 2132
Thabiso Tau	076 122 3458
Busi Nkosi	079 951 7536

(2)

2.1.3 Knoppie [2.1.3 - Average farm size]

Die gemiddelde grootte van al die plase van die geselekteerde boerderytype ('farm type') moet bereken en in 'n nuwe veld met die naam **AverageSize** gestoor word.

Kode is voorsien om die geselekteerde boerderytype ('farm type') uit die kombinasieblokkie ('combo box') **cmbQ2_1_3** te onttrek en in 'n veranderlike **sFarmType** te stoor.

Vertoon die **FarmType** en die **AverageSize**, tot TWEE desimale plekke geformateer.

Voorbeeld van afvoer indien Livestock as die boerderytype ('farm type') geselekteer is:

FarmType	AverageSize
Livestock	137.80

(6)

2.1.4 Knoppie [2.1.4 - Young farm owners]

Vertoon die **FullName**, **DateOfBirth** en **Age** (berekende veld) van alle plaaseienaars wat 25 jaar of jonger is. Die huidige stelseldatum ('current system date') moet gebruik word om die ouderdom as 'n heelgetalwaarde te bereken.

Gebruik die formule hieronder om die ouderdom van die plaaseienaars te bereken:

$$Age = \frac{Current\ system\ date - DateOf\ Birth}{365}$$

Voorbeeld van afvoer:

FullName	DateOfBirth	Age
John Smith	2002/01/12	23
Fatima Khan	2000/01/01	25
Johann van Rensburg	2003/04/28	22
Fred de Lange	2004/08/09	21

LET WEL: Die formaat van die geboortedatum mag van hierdie voorbeeld verskil as gevolg van jou rekenaar se instellings.

(7)

2.1.5 Knoppie [2.1.5 - Multiple farms in KZN]

Vertoon die **FullName** en die totale getal plase van elke plaaseienaar wat meer as een plaas in die Durban- of Pietermaritzburg-gebied besit. Die totale getal plase moet in 'n nuwe veld met die naam **TotalFarms** gestoor word.

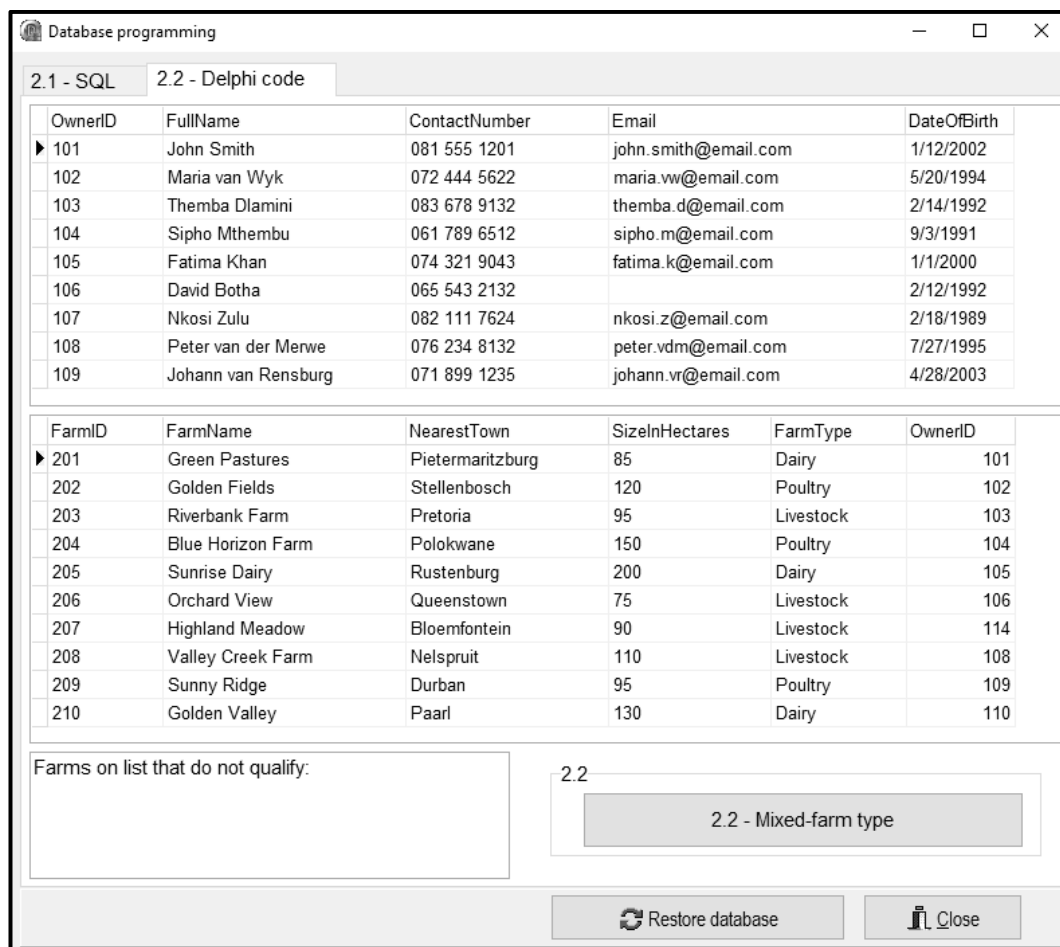
Voorbeeld van afvoer:

FullName	TotalFarms
Busi Nkosi	2
Johann van Rensburg	2
John Smith	3

(7)

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

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



LET WEL:

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

Knoppie [2.2 – Mixed-farm type]

'n Gemengdeboerdery-tipe verwys na 'n plaas met 'n kombinasie van verskillende tipes boerdery-aktiwiteite, soos pluimvee en suiwel. Boere moet versoek om na 'n gemengde boerdery te verander, indien hulle daarvoor kwalifiseer. Slegs plase groter as 100 hektaar kwalifiseer om 'n gemengdeboerdery-tipe te wees.

'n Tekslêer met die naam **MixedFarms.txt**, wat 'n lys bevat van **FarmID**'s van die plase wat moontlik as 'n gemengdeboerdery-tipe kan kwalifiseer, is voorsien.

Die eerste DRIE reëls met teks in die **MixedFarms.txt**-tekslêer word hieronder getoon.

201
204
206

Die program moet die boerderytype ('farm type') na 'Mixed' opdateer indien die plaas kwalifiseer om 'n gemengde boerdery te wees. Indien die plaas NIE kwalifiseer NIE, vertoon die **FarmID**, **FarmName** en **SizeInHectares** van die plaas in die 'rich edit' **redQ2_2**-komponent.

Kode is voorsien om die 'rich edit' **redQ2_2**-komponent skoon te maak en 'n geskikte opskrif te vertoon.

Skryf kode om die volgende te doen:

- Toets of die tekslêer **MixedFarms.txt** bestaan en vertoon 'n geskikte boodskap deur 'n 'ShowMessage'-dialoogblokkie ('dialog box') te gebruik indien die tekslêer nie bestaan nie.
- Indien die tekslêer bestaan, lees elke **FarmID** vanaf die tekslêer.
 - Identifiseer die plaas in die **tblFarms**-tabel.
 - Toets of die plaas groter as 100 hektaar is:
 - Indien waar, dateer die **FarmType**-veld van die plaas na 'Mixed' op.
 - Andersins, vertoon die **FarmID**, **FarmName** en **SizeInHectares** van die plaas in die 'rich edit' **redQ2_2**-komponent, in die formaat wat in die voorbeeld op die volgende bladsy getoon word.

Voorbeeld van rekords in die **tblFarms**-tabel voordat die **FarmType** verander is:

FarmID	FarmName	NearestTown	SizeInHectares	FarmType	OwnerID
201	Green Pastures	Pietermaritzburg	85	Dairy	101
202	Golden Fields	Stellenbosch	120	Poultry	102
203	Riverbank Farm	Pretoria	95	Livestock	103
204	Blue Horizon Farm	Polokwane	150	Poultry	104
205	Sunrise Dairy	Rustenburg	200	Dairy	105
206	Orchard View	Queenstown	75	Livestock	106
207	Highland Meadow	Bloemfontein	90	Livestock	114
208	Valley Creek Farm	Nelspruit	110	Livestock	108
209	Sunny Ridge	Durban	95	Poultry	109
210	Golden Valley	Paarl	130	Dairy	110

Voorbeeld van rekords in die **tblFarms**-tabel nadat die **FarmType** verander is:

FarmID	FarmName	NearestTown	SizeInHectares	FarmType	OwnerID
201	Green Pastures	Pietermaritzburg	85	Dairy	101
202	Golden Fields	Stellenbosch	120	Poultry	102
203	Riverbank Farm	Pretoria	95	Livestock	103
204	Blue Horizon Farm	Polokwane	150	Mixed	104
205	Sunrise Dairy	Rustenburg	200	Dairy	105
206	Orchard View	Queenstown	75	Livestock	106
207	Highland Meadow	Bloemfontein	90	Livestock	114
208	Valley Creek Farm	Nelspruit	110	Livestock	108
209	Sunny Ridge	Durban	95	Poultry	109
210	Golden Valley	Paarl	130	Dairy	110

Voorbeeld van afvoer van plase wat in die tekslêer gelys is en nie as 'n gemengdeboerdery-tipe ('mixed-farm type') kwalifiseer nie:

Farms on list that do not qualify: 201 - Green Pastures - 85 hectares 206 - Orchard View - 75 hectares 207 - Highland Meadow - 90 hectares 217 - Olifansberg - 99 hectares
--

(15)

- | |
|---|
| <ul style="list-style-type: none"> • 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

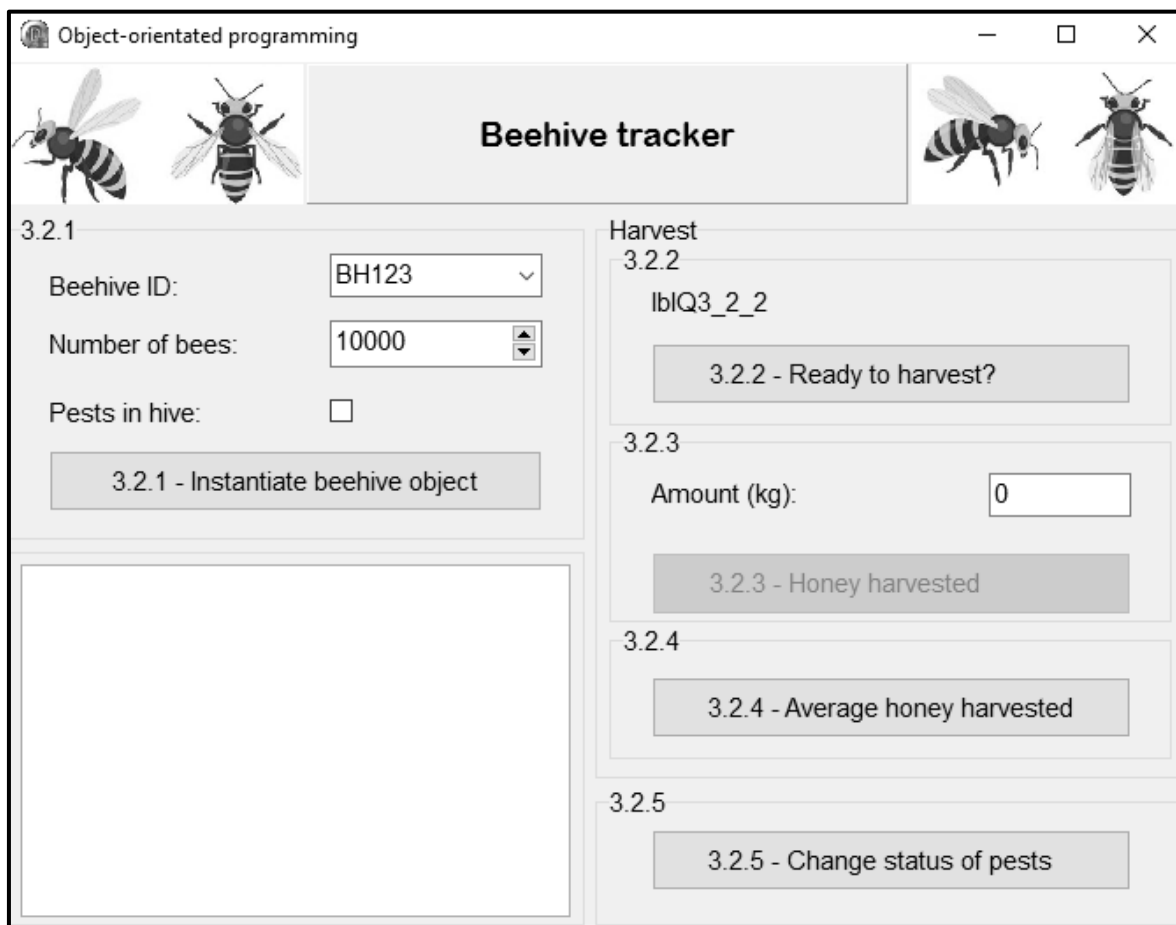
AFDELING C**VRAAG 3: OBJEK-GEORIËNTEERDE PROGRAMMERING**

Heuning word gedurende warm seisoene gereeld uit 'n byekorf geoes. 'n Byekorf-oesopspoorder word gebruik om die gesondheidstatus van 'n byekorf-kolonie te monitor en na te gaan.

Doen die volgende:

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

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



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

- 3.1 Die onvoltooide objekklas (**TBeehive**) wat voorsien is, bevat die verklaring van ses attribute wat 'n **Beehive**-objek beskryf.

Die attribute van 'n **Beehive**-objek is soos volg verklaar:

Attribuut	Tipe	Beskrywing
fBeehiveID	String	'n ID wat gebruik word om die byekorf te identifiseer
fBeeCount	Integer	Die geskatte aantal bye in die byekorf
fPests	Boolean	Dui aan of plaë tydens die laaste oes in die byekorf gevind is
fNoOfHarvests	Integer	Aantal voltooide oeste in die byekorf
fTotalHoneyHarvested	Real	Die totale hoeveelheid heuning, in kilogram, wat van die byekorf geoes is
fHarvestDates	String	'n Aaneengeskakelde string wat al die datums aandui waarop die byekorf tot dusver geoes is, elke datum op 'n nuwe reël

Die volgende metodes is voorsien:

- 'n Voltooide metode **setPests** wat die huidige status van die **fPests**-attribuut sal verander
- 'n Voltooide **toString**-metode

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

- 3.1.1 Skryf kode vir 'n konstruktor-metode om DRIE parameters vir die **fBeehiveID**-, **fBeeCount**- en **fPests**-attribute te ontvang.

Ken die parameterwaardes aan die onderskeie attribute toe en stel die oorblywende attribute op die volgende waardes:

$$\begin{aligned}
 \text{fNoOfHarvests} &= 2 \\
 \text{fTotalHoneyHarvested} &= 80 \\
 \text{fHarvestDates} &= '2025/01/05' + \#13 + '2025/06/23'
 \end{aligned}
 \tag{4}$$

- 3.1.2 Skryf kode vir 'n toegangsmetode ('accessor method') met die naam **getNoOfHarvests** om die **fNoOfHarvests**-attribuut terug te stuur. (2)

- 3.1.3 Skryf kode vir 'n metode met die naam **calcAverage**, om die gemiddelde hoeveelheid heuning wat in die korf geoes is, te bereken en terug te stuur, deur die formule hieronder te gebruik.

$$\text{Average} = \frac{\text{Total honey harvested}}{\text{Number of harvests}}
 \tag{4}$$

3.1.4 Skryf kode vir 'n metode met die naam **updateBeehiveDetails** wat 'n parameterwaarde vir die hoeveelheid heuning geoes as 'n reële waarde ontvang.

Skryf kode om die volgende te doen:

- Tel die parameterwaarde by die **fTotalHoneyHarvested**-attribuut.
- Inkrementeer ('increment') die **fNoOfHarvests**-attribuut met een.
- Voeg ('Join') die kode vir 'n nuwe reël (#13) en die stelseldatum ('system date') by die inhoud van die **fHarvestDates**-attribuut. (6)

3.1.5 Skryf kode vir 'n metode met die naam **checkHealthStatus** om 'n Boolese ('Boolean') waarde TRUE terug te stuur wat aandui dat die byekorf gesond ('healthy') is, of FALSE indien die byekorf nie gesond is nie.

Daar moet aan die volgende kriteria voldoen word vir die status van die byekorf om gesond te wees:

- Die aantal bye moet meer as 7 000 wees OF die aantal oeste moet gelyk aan of minder as 3 wees.
- Daar moet geen plaë in die byekorf wees nie. (6)

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 **objBeehive**.

Kode is in die **FormCreate**-metode voorsien om knoppie **btnQ3_2_3** te deaktiveer ('disable') en die 'rich edit' **redQ3_2**-komponent te formateer.

Skryf kode om die take wat in VRAAG 3.2.1 tot VRAAG 3.2.5 beskryf word, uit te voer.

3.2.1 **Knoppie [3.2.1 - Instantiate beehive object]**

Voordat heuning uit 'n byekorf geoes word, moet die gebruiker die besonderhede van die byekorf insleutel.

Kode is voorsien om die volgende te doen:

- Maak die 'rich edit' **redQ3_2**-komponent skoon.
- Onttrek die 'beehive ID' uit kombinasieblokkie ('combo box') **cmbQ3**.
- Onttrek die aantal bye uit spinredigeerder ('spin edit') **spnQ3**.
- Onttrek die plaagstatus uit kontroleblokkie ('check box') **chbQ3**.

Skryf kode om die volgende te doen:

- Gebruik die inligting wat onttrek ('extracted') is om die **objBeehive**-objek te instansieer.
- Vertoon die besonderhede van die objek in die 'rich edit' **redQ3_2** deur die gegewe **toString**-metode te gebruik.

Voorbeeld van toevoer en afvoer:

3.2.1

Beehive ID:

Number of bees:

Pests in hive:

Beehive ID: BH123
Number of bees in beehive: 10000
Honey harvested in kg: 80.0
Number of harvests: 2

Harvest dates:
2025/01/05
2025/06/23
No pests in beehive.

(5)

3.2.2 Knoppie [3.2.2 - Ready to harvest?]

Die gesondheidstatus van 'n byekorf moet nagegaan word voordat 'n nuwe oes plaasvind.

Skryf kode om die volgende te doen:

- Roep die **checkHealthStatus**-metode en vertoon 'n geskikte boodskap in die byskrif ('label') **lblQ3_2_2** om aan te dui of die byekorf gesond is ('ready to harvest') of nie gesond is nie ('not ready to harvest').
- Indien die byekorf gereed is om geoes te word, aktiveer ('enable') knoppie **btnQ3_2_3**.
- Indien die byekorf NIE gereed is om geoes te word NIE, deaktiveer ('disable') knoppie **btnQ3_2_3**.

Voorbeeld van afvoer indien die byekorf gereed is om geoes te word:

Voorbeeld van afvoer indien die byekorf NIE gereed is om geoes te word NIE:

(4)

3.2.3 **Knoppie [3.2.3 - Honey harvested]**

Die hoeveelheid heuning wat tydens 'n nuwe oes geoes is, moet by die totale hoeveelheid heuning wat uit die byekorf geoes is, getel word.

Kode is voorsien om die 'rich edit' **redQ3_2**-komponent skoon te maak.

Skryf kode om die volgende te doen:

- Onttrek die hoeveelheid heuning wat geoes is uit die redigeerblokkie ('edit box') **edtQ3_2_3**.
- Roep die **updateBeehiveDetails**-metode deur die hoeveelheid heuning as 'n argument te gebruik.
- Vertoon die volgende in die **redQ3_2**-komponent:
 - Die getal oeste in die formaat:
'Harvest number: ' <Number of harvests>
 - Die opgedateerde besonderhede van die byekorf deur die **toString**-metode te gebruik

Voorbeeld van afvoer indien die heuning wat tydens die nuwe oes ingesamel is 67,1 kg was en die getal voltooide oeste is drie:

```
Harvest number:3
Beehive ID:      BH123
Number of bees in beehive: 10000
Honey harvested in kg: 147.1
Number of harvests: 3

Harvest dates:
2025/01/05
2025/06/23
2025/10/22
No pests in beehive.
```

(4)

3.2.4 Knoppie [3.2.4 - Average honey harvested]

Skryf kode om die relevante metode te roep om die gemiddelde hoeveelheid heuning wat uit die byekorf geoes is, in 'n ShowMessage-dialogblokkie ('dialog box') te vertoon, tot TWEE desimale plekke geformateer.

Voorbeeld van afvoer indien die totale hoeveelheid heuning ná drie oeste 147,1 kg was:

```
Average honey harvested = 49.03
```

OK

(3)

3.2.5 Knoppie [3.2.5 - Change status of pests]

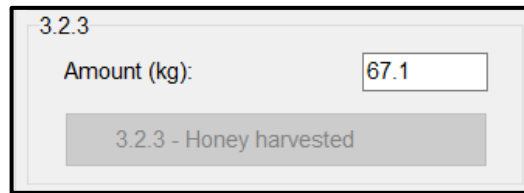
Die status van plaë in die byekorf moet verander word, afhangende daarvan of plaë opgespoor is of nie.

'n Metode met die naam **setPests**, wat die huidige status van plaë in die byekorf sal verander, is in die objekklas voorsien.

Skryf kode om die volgende te doen:

- Roep die **setPests**-metode.
- Deaktiveer ('Disable') die **btnQ3_2_3**-knoppie.

Voorbeeld van die 'Honey harvested'-knoppie wat gedeaktiveer is wanneer die gebruiker een keer op die 'Change status of pests'-knoppie geklik het:



The screenshot shows a window with a title bar containing the text '3.2.3'. Inside the window, there is a label 'Amount (kg):' followed by a text input field containing the value '67.1'. Below this, there is a greyed-out button with the text '3.2.3 - Honey harvested'.

(2)

- 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

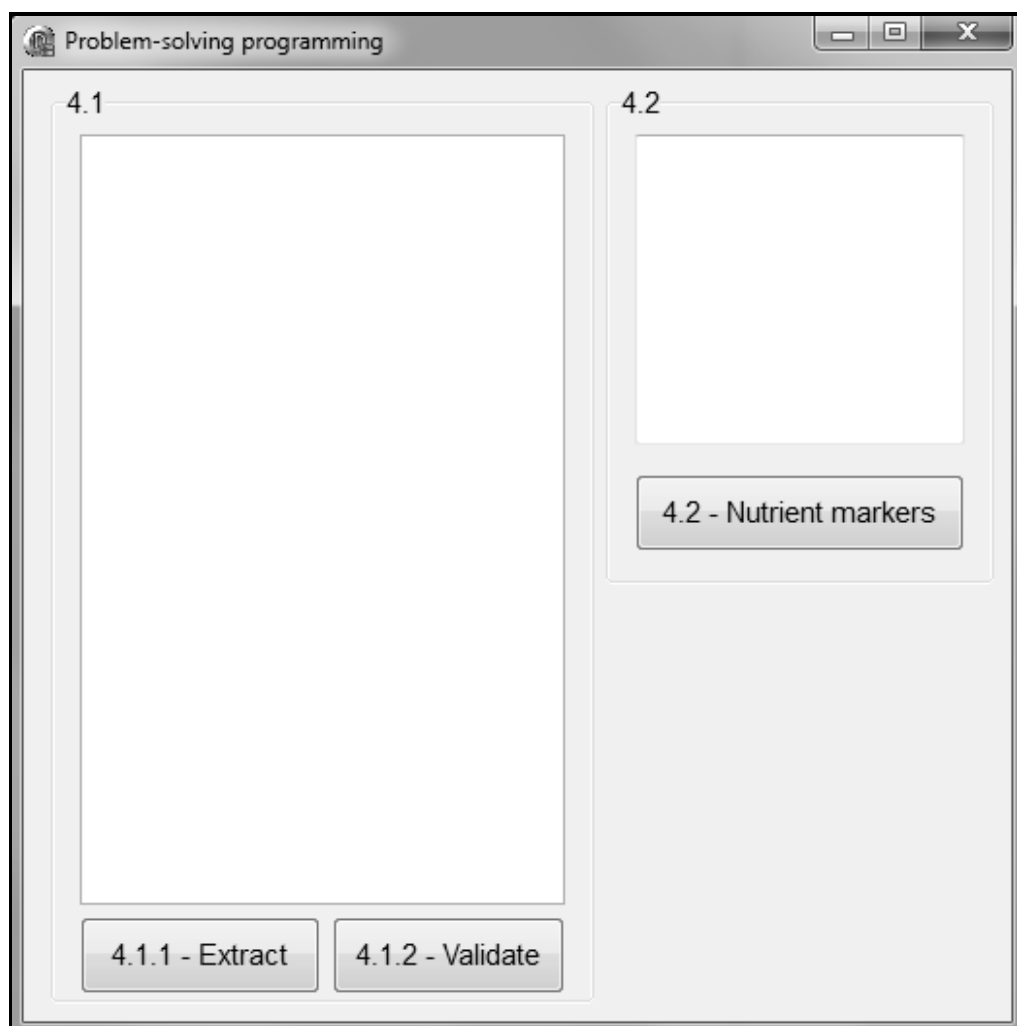
AFDELING D**VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING**

SustainaFarm Collective is 'n maatskappy wat navorsing doen oor volhoubare boerderypraktyke in gebiede waar verskillende tipes gewasse geplant word.

Doen die volgende:

- Maak die onvolledige program in die **Question 4**-lêergids oop.
- Sleutel jou eksamennommer as kommentaar in die eerste reël van die **Question4_U.pas**-lêer in.
- Kompileer en voer die program uit. Die program het tans beperkte funksionaliteit.

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



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

- 4.1 Die SustainaFarm Collective monitor die samestelling van grond in tien hektaar boerderygrond deur die persentasiewaardes per hektaar vir klei ('Clay'), sand ('Sand') en slik ('Silt') in die grond te meet.

Jy is van die volgende voorsien:

- 'n Voltooide **Display**-prosedure
- Verklarings vir die volgende skikkings:
 - 'n Gevulde eendimensionele skikking van tipe string waar elke element van die skikking die hoeveelheid klei, sand en slik in die grond verteenwoordig:


```
arrSoil: array[1..10] of String =
('20:50:10', '40:30:30', '30:28:30', '60:20:20',
'25:30:45', '50:40:10', '30:55:15', '45:35:20',
'35:0:55', '55:25:20');
```
 - 'n Tweedimensionele skikking van reële tipe ('type real'):


```
arr2DSoil: array[1..10, 1..3] of Real;
```

4.1.1 Knoppie [4.1.1 - Extract]

Skryf kode om die Clay-, Sand- en Silt-waardes vir elke hektaar uit die **arrSoil**-skikking te onttrek en dit in die gegewe tweedimensionele skikking **arr2DSoil** te stoor.

Kode is voorsien om die volgende te doen:

- Maak die 'rich edit'-komponent **redQ4_1** skoon.
- Roep die **Display**-metode om die data in die **arr2DSoil** tweedimensionele skikking in die 'rich edit'-komponent **redQ4_1** af te voer.

Voorbeeld van afvoer:

Hectare	Clay	Sand	Silt
1	20.0	50.0	10.0
2	40.0	30.0	30.0
3	30.0	28.0	30.0
4	60.0	20.0	20.0
5	25.0	30.0	45.0
6	50.0	40.0	10.0
7	30.0	55.0	15.0
8	45.0	35.0	20.0
9	35.0	0.0	55.0
10	55.0	25.0	20.0

(8)

4.1.2 Knoppie [4.1.2 - Validate]

Dit is vir die SustainaFarm Collective belangrik dat die som van die Clay-, Sand- en Silt-waardes vir elke hektaar aan die waarde van 100 gelyk is.

Kode is voorsien om die **redQ4_1**-komponent skoon te maak en die opskrif 'Hectares adjusted' te vertoon.

Skryf kode om die volgende te doen:

- Bereken die som van die Clay-, Sand- en Silt-waardes vir elke hektaar.
- Indien die som van enige hektaar nie die waarde van 100 het as dit opgetel word nie, doen die volgende:
 - Pas die Clay-, Sand- en Silt-waardes vir daardie hektaar in dieselfde verhouding aan om seker te maak dat die som van die drie waardes, as dit opgetel word, 100 is.
 - Vertoon die hektaarnommer in die **redQ4_1**-komponent in die formaat:

Hectare: <hectare number>

LET WEL: Neem aan dat die totaal vir elke hektaar nie meer as 100 sal wees nie.

Kode is voorsien om die volgende te doen:

- Vertoon die opskrif 'Updated data:'.
- Roep die **Display**-metode om die data van die **arr2DSoil** tweedimensionele skikking in die 'rich edit'-komponent **redQ4_1** af te voer.

Voorbeeld van afvoer:

Hectares adjusted:			
Hectare 1			
Hectare 3			
Hectare 9			
Updated data:			
=====			
Hectare	Clay	Sand	Silt
1	25.0	62.5	12.5
2	40.0	30.0	30.0
3	34.1	31.8	34.1
4	60.0	20.0	20.0
5	25.0	30.0	45.0
6	50.0	40.0	10.0
7	30.0	55.0	15.0
8	45.0	35.0	20.0
9	38.9	0.0	61.1
10	55.0	25.0	20.0

(9)

4.2 Knoppie [4.2 - Nutrient markers]

Die SustainaFarm Collective doen navorsing oor die groeipotensiaal van gewasse deur unieke voedingstofmerkers ('nutrient markers') te gebruik. Voedingstofmerkers is spesiale getalle waar die som van die fakulteite ('factorials') van hulle syfers aan die getal self gelyk is.

Voorbeeld 1:

Die getal 145 is 'n voedingstofmerker ('nutrient marker') aangesien die som van die fakulteite ('factorials') van die syfers aan die oorspronklike getal 145 gelyk is.

$$1! = 1 \times 1 = 1$$

$$4! = 4 \times 3 \times 2 \times 1 = 24$$

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

$$1 + 24 + 120 = 145$$

Voorbeeld 2:

Die getal 32 is NIE 'n voedingstofmerker ('nutrient marker') NIE.

$$3! = 3 \times 2 \times 1 = 6$$

$$2! = 2 \times 1 = 2$$

$$6 + 2 = 8 \neq 32$$

Kode is voorsien om die volgende te doen:

- Maak die memokomponent **memQ4_2** skoon.
- Vertoon die onderstreepte opskrif 'Nutrient markers:'.

Skryf kode om die volgende te doen:

- Bepaal al die voedingstofmerkers ('nutrient markers') in die reeks van 1 tot 50 000 (ingesluit).
- Vertoon elke voedingstofmerker ('nutrient marker') in die memokomponent **memQ4_2**.

Voorbeeld van afvoer:

```
4.2
Nutrient markers:
=====
1
2
145
40585

4.2 - Nutrient markers
```

(13)

- 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 D: 30
GROOTTOTAAL: 150

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

Die ontwerp van die databasistabelle is soos volg:

Tabel: **tblFarmOwners**

Hierdie tabel bevat die besonderhede van plaaseienaars.

Veldnaam	Datatype	Beskrywing
OwnerID (PK)	Number	Unieke identifiseerder vir die plaaseenaar
FullName	Text (20)	Volle naam van die plaaseenaar
ContactNumber	Text (12)	Kontaknommer van die plaaseenaar
Email	Text (25)	E-posadres van die plaaseenaar
DateOfBirth	Date/Time	Geboortedatum van die plaaseenaar

Voorbeeld van die eerste nege rekords van die **tblFarmOwners**-tabel:

OwnerID	FullName	ContactNumber	Email	DateOfBirth
101	John Smith	081 555 1201	john.smith@email.com	2002/01/12
102	Maria van Wyk	072 444 5622	maria.vw@email.com	1994/05/20
103	Themba Dlamini	083 678 9132	themba.d@email.com	1992/02/14
104	Sipho Mthembu	061 789 6512	sipho.m@email.com	1991/09/03
105	Fatima Khan	074 321 9043	fatima.k@email.com	2000/01/01
106	David Botha	065 543 2132		1992/02/12
107	Nkosi Zulu	082 111 7624	nkosi.z@email.com	1989/02/18
108	Peter van der Merwe	076 234 8132	peter.vdm@email.com	1995/07/27
109	Johann van Rensburg	071 899 1235	johann.vr@email.com	2003/04/28

Tabel: **tblFarms**

Hierdie tabel bevat die besonderhede van elke plaas.

Veldnaam	Datatype	Beskrywing
FarmID (PK)	Number	Unieke identifiseerder vir elke plaas
FarmName	Text (20)	Naam van die plaas
NearestTown	Text (20)	Die dorp naaste aan die plaas
SizeInHectares	Number	Grootte van die plaas in hektaar
FarmType	Text (10)	Tipe plaas (bv. Dairy, Poultry, Mixed)
OwnerID (FK)	Number	Die eienaar-ID ('owner ID') van die plaaseenaar

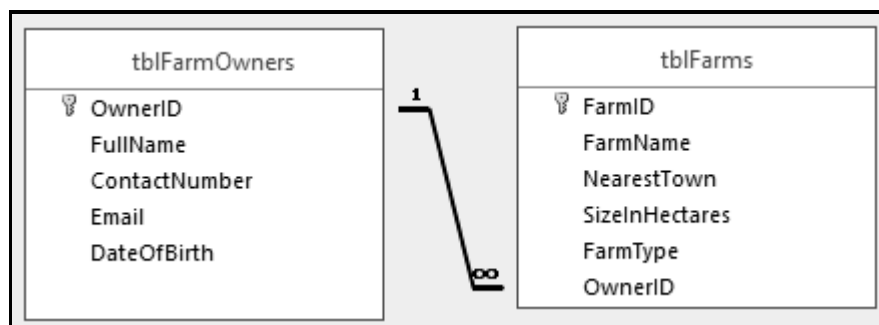
Voorbeeld van die eerste elf rekords van die **tblFarms**-tabel:

FarmID	FarmName	NearestTown	SizeInHectares	FarmType	OwnerID
201	Green Pastures	Pietermaritzburg	85	Dairy	101
202	Golden Fields	Stellenbosch	120	Poultry	102
203	Riverbank Farm	Pretoria	95	Livestock	103
204	Blue Horizon Farm	Polokwane	150	Mixed	104
205	Sunrise Dairy	Rustenburg	200	Dairy	105
206	Orchard View	Queenstown	75	Livestock	106
207	Highland Meadow	Bloemfontein	90	Livestock	114
208	Valley Creek Farm	Nelspruit	110	Livestock	108
209	Sunny Ridge	Durban	95	Poultry	109
210	Golden Valley	Paarl	130	Dairy	110
211	Riverside Ranch	Pretoria	100	Livestock	108

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

Eksamenplakker

150**INLIGTINGSTEGNOLOGIE V1 – NOVEMBER 2025****INLIGTINGSBLAD** (moet NA AFLOOP VAN die 3 uur-sessie deur die kandidaat ingevul word)

SENTRUMNOMMER: _____

EKSAMENNOMMER: _____

WERKSTASIENOMMER: _____

Weergawe van Delphi wat tydens die INLIGTINGSTEGNOLOGIE NSS NOV. 2025-eksamen gebruik is:

Maak 'n kruisie in die toepaslike blokkie (X)	Delphi 2010	Delphi XE	Delphi 10.3	Delphi Community	Delphi 11	Delphi 12	Ander: (Spesifiseer) _____
---	-------------	-----------	-------------	------------------	-----------	-----------	-------------------------------

NAAM VAN LÊERGIDS ('FOLDER'): _____

Kandidaat moet afmerk as die lêernaam (lêername) wat vir elke antwoord gebruik is, gestoor is en/of 'n poging aangewend is.

Vraag-nummer	Lêernaam	Gestoor (✓)	Gepoog (✓)	Maksimum Punt	Punt Toegeken	Nasiener se Kode
1	Question1_P.dproj			40		
2	Question2_P.dproj			40		
3	Beehive_U.pas			22		
	Question3_P.dproj			18		
4	Question4_P.dproj			30		
TOTAAL				150		

Kommentaar: (slegs vir kantoor/nasiener se gebruik)



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2025

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 27 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as die basis vir die nasiensessie gebruik word. Dit is voorberei om deur nasieners gebruik te word. Daar word van alle nasieners verwag om 'n deeglike standaardiseringsvergadering by te woon om seker te maak dat die riglyne konsekwent geïnterpreteer en tydens die nasien van die leerders se werk toegepas word.
- Let op dat leerders wat 'n alternatiewe korrekte oplossing as wat as voorbeeld van 'n oplossing in die nasienriglyne gegee word verskaf, volle krediet vir die relevante oplossing moet kry tensy die spesifieke instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom is nie.
- **Bylae A, B, C en D** (bladsy 3 tot 12) sluit die nasienriglyne vir elke vraag in.
- **Bylae E, F, G en H** (bladsy 13 tot 27) bevat voorbeelde van oplossings vir Vrae 1 tot 4 in programmeringskode.
- Kopieë van **Bylae A, B, C, D en die opsomming van die leerder se punte** (bladsy 3 tot 12) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

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

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

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

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1	SQL-stellings		
2.1.1	Knoppie [2.1.1 - Farm details] SELECT FarmName, NearestTown, SizeInHectares ✓ FROM tblFarms ✓ ORDER BY SizeInHectares DESC ✓	3	
2.1.2	Knoppie [2.1.2 - Contact details] SELECT FullName, ContactNumber FROM tblFarmOwners ✓ WHERE Email IS NULL ✓ Alternatief: ISNULL(Email)	2	
2.1.3	Knoppie [2.1.3 - Average farm size] SELECT FarmType, FORMAT(AVG(SizeInHectares) ✓, "0.00" ✓) AS AverageSize ✓ FROM tblFarms WHERE FarmType = ✓ ' ' + sFarmType + ' ' ✓ GROUP BY FarmType ✓ Alternatief: FarmType = ' + QuotedStr(sFarmType) + ' FarmType LIKE ' ' + sFarmType + ' '	6	
2.1.4	Knoppie [2.1.4 - Young farm owners] SELECT Fullname, DateOfBirth, ✓ INT ✓ ((NOW ✓ - DateOfBirth) /365 ✓) AS Age FROM tblFarmOwners ✓ WHERE INT((NOW - DateOfBirth)/365) ✓ <= 25 ✓ Alternatief: (Date() - DateOfBirth)	7	

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

VRAAG 2: NASIENRUBRIEK (VERVOLG)

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

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

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

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

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

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

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

OPSOMMING VAN LEERDER SE PUNTE:

SENTRUMNOMMER:		LEERDER SE EKSAMENNUMMER:			
	AFDELING A	AFDELING B	AFDELING C	AFDELING D	
	VRAAG 1	VRAAG 2	VRAAG 3	VRAAG 4	GROOT-TOTAAL
MAKS. PUNTE	40	40	40	30	150
LEERDER SE PUNTE					

BYLAE E: OPLOSSING VIR VRAAG 1

```
unit Question1_U;

interface

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

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

public
    { Public declarations }

end;

var
    frmQuestion1: TfrmQuestion1;

implementation
{$R *.dfm}
```

```
// =====  
// Vraag 1.1                   4 punte  
// =====  
procedure TfrmQuestion1.btnQ1_1Click(Sender: TObject);  
var  
    iNum: Integer;  
begin  
  
    // Question 1.1  
    iNum := StrToInt(pnlQ1_1.Caption);  
    Inc(iNum);  
    pnlQ1_1.Caption := IntToStr(iNum);  
end;  
  
// =====  
// Vraag 1.2                   6 punte  
// =====  
procedure TfrmQuestion1.btnQ1_2Click(Sender: TObject);  
var  
    iSide: Integer;  
    rVolume: Real;  
begin  
    // Question 1.2  
    iSide := spnQ1_2.Value;  
    rVolume := 5 * (3 + Sqrt(5)) / 12 * Power(iSide,3);  
    edtQ1_2.Text := FloatToStrF(rVolume, ffFixed, 8, 2);  
end;  
  
// =====  
// Vraag 1.3                   11 punte  
// =====  
procedure TfrmQuestion1.btnQ1_3Click(Sender: TObject);  
var  
    sCode, sOutput: string;  
    sAisle: string;  
    cCategory : Char;  
    iPos : Integer;  
begin  
    // Question 1.3  
    sCode := edtQ1_3.Text;  
    iPos := Pos('#', sCode);  
    cCategory := sCode[iPos+1];  
    sAisle := Copy(sCode, 1, iPos-1);  
    case cCategory of  
        'F': sOutput := 'Fruit and vegetables';  
        'D': sOutput := 'Dairy';  
        'B': sOutput := 'Butchery';  
        'S': sOutput := 'Sauces';  
        'P': sOutput := 'Pasta and Rice';  
    end;  
    sOutput := 'Aisle ' + sAisle + ': ' + sOutput;  
    lblQ1_3.Caption := sOutput;  
end;
```

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

Code using a Repeat Until loop

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

//Code using a For loop

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

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

```
end;
```

```
end.
```

BYLAE F: OPLOSSING VIR VRAAG 2

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

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

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

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

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

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

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

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

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

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

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

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

end;

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

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

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

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

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

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

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

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

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

```

unit Beehive_U;

interface

Uses SysUtils, DateUtils, Math;

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

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

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

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

```

```

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

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

```

Hoofvormeenheid:

```

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

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

```

implementation

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

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

BYLAE H: OPLOSSING VIR VRAAG 4

```
unit Question4_U;

interface

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

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

  private
    { Private declarations }
  public
    procedure Display;
  end;

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

implementation

{$R *.dfm}
```

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

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

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

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

    if (rSum < 100) then
    begin
      for A := 1 to 3 do
        arr2DSoil[iRow, A] := (arr2DSoil[iRow, A] / rSum * 100);
        redQ4_1.Lines.Add('Hectare ' + IntToStr(iRow));
      end;
    end;
  // Provided code
  redQ4_1.Lines.Add(#13 + 'Updated data: ');
  redQ4_1.Lines.Add('=====');
  Display;
end;
```

```

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

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

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

```