



Province of the
EASTERN CAPE
EDUCATION

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

SEPTEMBER 2021

INLIGTINGSTEGNOLOGIE V1

PUNTE: 150

TYD: 3 uur

Hierdie vraestel bestaan uit 19 bladsye.

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 verskaf is of enige data-uittreksels wat in die vraestel verskyn nie.
7. Roetines, soos vind, 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 verskaf word.
9. 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.
10. Maak seker dat jou naam 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 wagwoordbeskermdes uitvoerbare lêers verskaf.

Doen die volgende:

- Dubbelklik op die wagwoordbeskermdes uitvoerbare lêer.
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **gR12#21sEpt**

Nadat dit onttrek ('extracted') is, sal die volgende lys lêers in die lêergids ('folder') **DataSept2021** beskikbaar wees:

Vraag 1:

Vraag1_u.pas
Vraag1_u.dfm
Vraag1_p.dpr
Vraag1_p.res
reën.jpg

Vraag 2:

dbConnection_u.pas
Voëls.mdb
VoëlsRugsteun.mdb
Vraag2_u.pas
Vraag2_u.dfm
Vraag2_p.dpr
Vraag2_p.res

Vraag 3:

Vraag3KlasDefinisie.pas
Vraag3_u.pas
Vraag3_u.dfm
Vraag3_p.dpr
Vraag3_p.res

Vraag 4:

water.txt
Vraag4_u.pas
Vraag4_u.dfm
Vraag4_p.dpr
Vraag4_p.res

VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE

Doen die volgende:

- Maak die onvoltooide program in die **Vraag 1** lêergids oop.
- Sleutel jou naam en van as kommentaar in die eerste reël van die **Vraag1_u.pas** lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.
- Volg die instruksies hieronder om die kode vir elke afdeling van VRAAG 1, soos beskryf in VRAAG 1.1, VRAAG 1.2, VRAAG 1.3 en VRAAG 1.4, te voltooi.

Jou skool het 'n klub gestig wat inligting verskaf en navorsing oor die water toevoer en reënval kwessies doen. Voltooi die program vir die klub deur die instruksies hieronder te volg.

1.1 Knoppie [1.1 Vertoon]

Skryf kode om die volgende te doen:

Laai die prentjie, **reën.jpg**, in die prentjie-komponent met die naam **imgReën**.

Maak die paneel, **pnlBad**, aktief.

Verander die agtergrondkleur van die paneel, **pnlReën**, na aqua.

(3)

1.2 Knoppie [1.2 Verwerk]

Reënval verwerker.

Hoeveel 150-liter-baddens kan volgemaak word met die reënval wat in 'n sekere area in jou tuin of op jou dak geval het?

Skryf kode om die volgende te doen:

Die gebruiker moet die hoogte en wydte in die 'spinedit' komponente met die name **sedWydte** en **sedHoogte** in lees. Die aantal millimeter moet in die **cmbReën**-kombinasielys gekies word.

Die aantal 150-liter-baddens is gelyk aan die aantal vierkante meter vermenigvuldig met die hoeveelheid water in millimeter gedeel deur 150. Rond die antwoord na 1 desimale plek af.

Voorbeeld: 'n Area van 20 m by 35 m ontvang 1/2 millimeter (0.5 mm) reën.

$$20 \times 35 \times 0.5 / 150 = 2.3 \text{ baddens (150 liters elk)}$$

Gebruik die volgende data om jou oplossing te toets:

Reënval in mm	Hoogte in meters	Wydte in meters	Aantal 150 liter-baddens
15	9	10	9
1/10	20	20	0.3
1/2	20	35	2.3
30	9	12	22

Voorbeeld van afvoer:

(11)

1.3 Knoppie [1.3 Vind reënvaltype]

Ons wil probeer uitvind watter tipe reënval in jou area geval het.

Die volgende beskrywings is van toepassing op die verskillende hoeveelhede reën wat binne een uur geval het.

Geen reën: Minder as 1 mm per uur

Matige reën: Groter as of gelyk aan 1 mm per uur, maar minder as 4 mm per uur.

Swaar reën: Groter as of gelyk aan 4 mm per uur, maar minder as 8 mm per uur.

Baie swaar reën: Groter as of gelyk aan 8 mm per uur, maar minder as 10 mm per uur.

Stortreën: Groter as of gelyk aan 10 mm per uur, maar minder of gelyk aan 50 mm per uur.

Vloed: Meer as 50 mm per uur.

Skryf kode vir die knoppie om te bepaal watter tipe reën geval het. Die gebruiker moet die hoeveelheid reënval in millimeter per uur in die redigeerblokkie, **edtReën**, in tik. Die program moet dan die tipe reënval bereken en in die **lblReënvalTipe** vertoon soos wat hierbo beskryf is.

Jy moet ook toets of 'n heelgetal ('integer') in die redigeerblokkie, **edtReën**, ingelees is. As dit nie 'n heelgetal is nie, moet 'n foutboodskap vertoon word en die prosedure moet ge-'exit' word.

Voorbeelde van afvoer:

Tik die reenval per uur in (mm): 0 V1.3 Vind reënvaltype Geen reën	Tik die reenval per uur in (mm): 3 V1.3 Vind reënvaltype Matige reën	Tik die reenval per uur in (mm): 6 V1.3 Vind reënvaltype Swaar reën
Tik die reenval per uur in (mm): 9 V1.3 Vind reënvaltype Baie swaar reën	Tik die reenval per uur in (mm): 25 V1.3 Vind reënvaltype Stortreën	Tik die reenval per uur in (mm): 51 V1.3 Vind reënvaltype Vloed

(13)

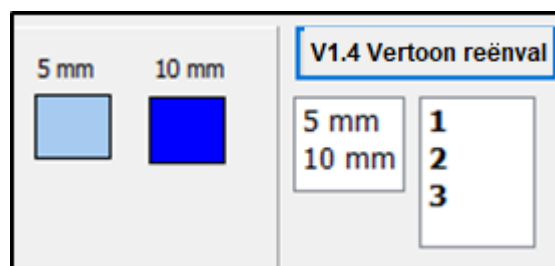
1.4 Knoppie [1.4 Vertoon Reënval]

Jy word vereis om die aantal kere wat 5 mm of 10 mm reënval in 8 verskillende areas geval het, te bereken en te vertoon.

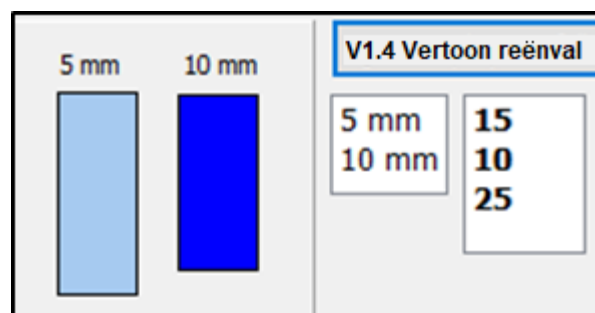
Skryf kode om die volgende instruksies 8 keer te herhaal:

- Genereer 2 verskillende ewekansige getalle tussen 5 en 10 (albei ingesluit).
- As die ewekansige getalle 'n 5 of 'n 10 is, dan moet 1 by die hoeveelheid in die 'listbox', **LstHoeveelheid** bygetel word. Dit moet op die korrekte reël bygetel word sodat dit met die 5 mm of 10 mm in die **LstReën** 'listbox' ooreenstem.
- Bereken die som van die twee hoeveelhede en vertoon dit op die derde reël van die **LstHoeveelheid** 'listbox'.
Byvoorbeeld: $1 + 2 = 3$ en $15 + 10 = 25$
- Verander die hoogtes van die vorms ('shapes') met die name **shp5mm** en **shp10mm**, deur die waardes van die ooreenstemmende getalle in **LstHoeveelheid** by die huidige hoogte van elke vorm by te tel.

Voorbeeld van afvoer nadat die knoppie een keer uitgevoer is. Dit wys dat 1 area 5 mm reën en 2 areas 10 mm reën gehad het:



Voorbeeld van afvoer nadat die knoppie meer as een keer geklik is:



(13)

- Sleutel jou naam en van as kommentaar in die eerste reël van die programlêer.
- Stoor jou program.
- 'n Drukstuk van jou kode mag vereis word.

[40]

VRAAG 2: DATABASISPROGRAMMERING

Die databasis **Voëls.mdb** bevat inligting van voëls en hul status in Suid-Afrika. Die databasis bestaan uit twee tabelle, naamlik **Voël** en **Status**.

Tabel: **Status**

Hierdie tabel bevat die beskrywings van die moontlike status van 'n voël.

Veldnaam	Datatype	Beskrywing
StatusID	Number	'n Unieke nommer is aan elke status toegeken
StatusNaam	Text (50)	Die beskrywing van die status van 'n voël

Voorbeeld van data in die **Status** tabel:

StatusID	StatusNaam
1	Critically Endangered
2	Endangered
3	Vulnerable
4	Near Threatened
5	Least Concern
6	Not Assessed

Tabel: **Voël**

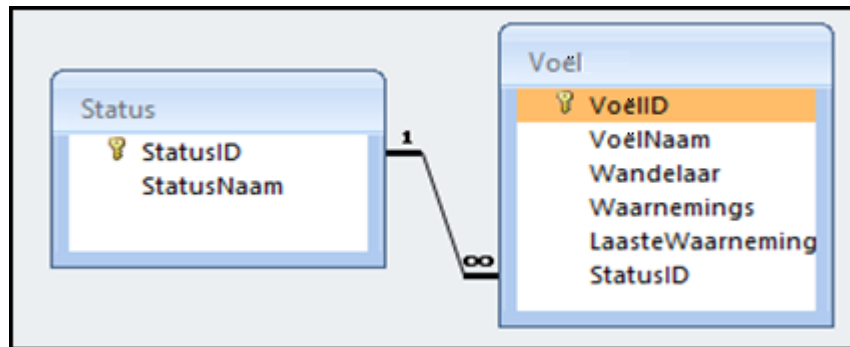
Hierdie tabel bevat inligting van voëls en hul waarnemings in Suid-Afrika.

Veldnaam	Datatype	Beskrywing
VoëlID	Number	'n Unieke nommer wat aan 'n voël toegeken is
VoëlNaam	Text (50)	Die algemene naam van 'n voël
Wandelaar	Yes/No	Voëls dwaal buite die perke van hul natuurlike gebied — kan enige plek gevind word
Waarnemings	Number	Totale aantal waarnemings van 'n voël
Laaste Waarneming	Date (short date)	Datum wat die voël laas gesien is
StatusID	Number	Vreemde sleutel om met die Bird-tabel te verbind

Voorbeeld van data van die eerste veertien rekords van die **Voëls** tabel:

VoelID	VoelNaam	Wandelaar	Waarneming	LaasteWaarneming	StatusID
1	African Broadbill	☐	533	2005/01/23	3
2	African Finfoot	☐	208	2007/07/12	3
3	African Grass Owl	☐	161	2012/05/27	3
4	African Marsh Harrier	☐	366	2010/10/08	2
5	African Rock Pipit	☐	918	2005/12/06	4
6	Agulhas Long-billed Lark	☐	726	2006/07/23	4
7	Antarctic Tern	☐	333	2008/07/25	2
8	Barlow's Lark	☐	720	2009/03/19	4
9	Bat Hawk	☐	347	2007/01/17	2
10	Bateleur	☐	364	2011/03/11	2
11	Bearded Vulture	☐	352	2007/12/08	1
12	Blue Crane	☐	206	2012/05/23	4
13	Blue Korhaan	☐	202	2010/10/11	5
14	Blue Petrel	☒	511	2011/01/18	4

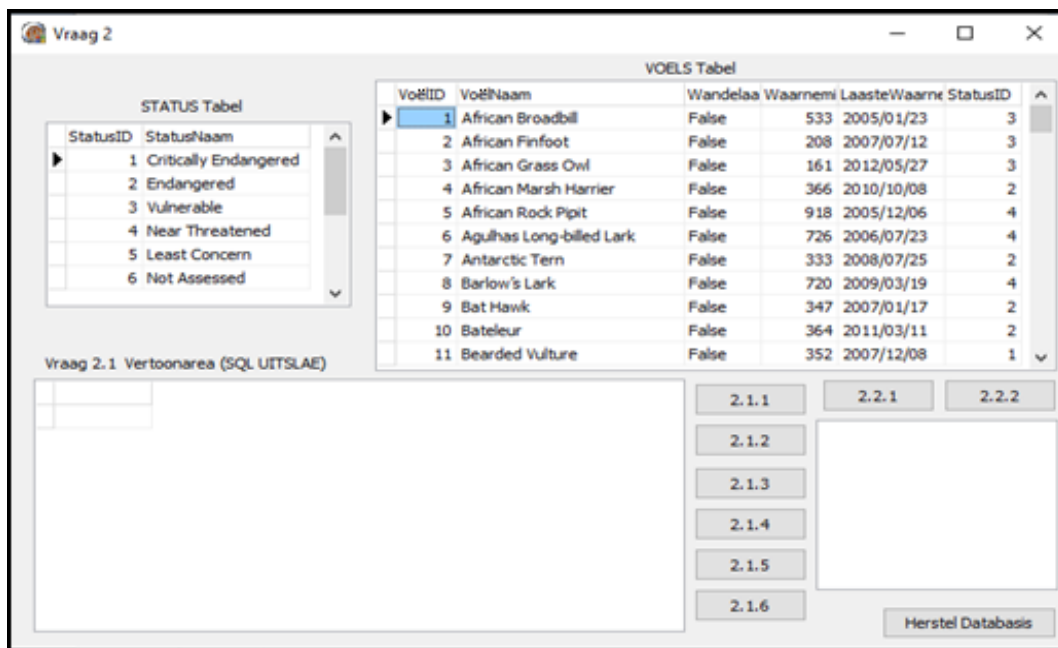
Die verwantskap tussen die twee tabelle word hieronder gewys:



Doen die volgende:

- Maak die onvoltooide projeklêer, **Vraag2_p.dpr** in die Vraag 2 gidslêer oop.
- Tik jou naam en van as kommentaar in die eerste reël van die **Vraag2_u.pas** lêer in.
- Kompileer en voer die program uit. Die program het tans geen funksionaliteit nie.

Die gebruikerskoppelvlak word hieronder vertoon:



- Volg die instruksies om die kode vir elke vraag te voltooi, soos in VRAAG 2.1 en VRAAG 2.2 beskryf is.
- **Gebruik SQL-kode om VRAAG 2.1 te beantwoord en Delphi-kode om VRAAG 2.2 te beantwoord.**

LET WEL:

- Die [**Herstel Databasis**] knoppie word verskaf om die data wat in die databasis voorkom, na die oorspronklike inhoud te herstel. As jy jou kode op die oorspronklike data moet toets, mag jy die knoppie klik om die data te herstel.
- MOENIE gegewe kode verander NIE.
- Kode word gegee om die GGK-komponente met die databasis te verbind.
- **TWEE** veranderlikes is as globale veranderlikes verklaar, soos in die onderstaande tabel beskryf word.
- **Gebruik tblStatus en tblVoëls komponente net in Vraag 2.2.**

Veranderlike	Datatype	Beskrywing
tblStatus	TADOTable	Verwys na die tabel met die naam Status
tblVoëls	TADOTable	Verwys na die tabel met die naam Voëls

- 2.1 In hierdie afdeling mag jy SLEGS **SQL-stellings** gebruik om VRAAG 2.1.1 tot VRAAG 2.1.5. te beantwoord.

Kode word voorsien om die **SQL-stellings** uit te voer en die resultate van die navrae te vertoon. Die SQL-stellings is onvolledig.

Doen die volgende om die onvoltooide **SQL-stellings** wat respektiewelik per vraag aan die veranderlikes sSQL1, sSQL2, sSQL3, sSQL4 en sSQL5 toegeken is, te voltooi.

2.1.1 Knoppie [2.1.1]

Skryf SQL-kode wat al die **Status** inligting in omgekeerde alfabetiese volgorde volgens die **StatusNaam**, sal vertoon.

Voorbeeld van afvoer:

StatusID	StatusNaam
3	Vulnerable
6	Not Assessed
4	Near Threatened
5	Least Concern
2	Endangered
1	Critically Endangered

(3)

2.1.2 Knoppie [2.1.2]

Skryf SQL-kode om die **VoëlNaam** van al die voëls wat minder as 200 **Waarnemings** gehad het, te vertoon.

Voorbeeld van afvoer:

VoëlNaam
African Grass Owl
Cape Parrot
Eastern Bronze-naped Pigeon
Green Barbet
Ground Woodpecker
Maccoa Duck
Mangrove Kingfisher
Pel's Fishing Owl
Swamp Nightjar

(3)

2.1.3 Knoppie [2.1.3]

Die gebruiker moet die naam van 'n voël intik. Kode word verskaf wanneer “**vulture**” in 'n 'inputbox' ingelees word en in 'n veranderlike **sToevoer**, gestoor word.

Skryf SQL-kode om die **VoëlNaam** en **LaasteWaarneming** datum van al die voëls, wat die woord wat in die **sline** veranderlike, gestoor is, sal vertoon.

Voorbeeld van afvoer:

VoëlNaam	LaasteWaarneming
Bearded Vulture	2007/12/08
Egyptian Vulture	2006/08/09
Rüppell's Vulture	2011/05/19

(4)

2.1.4 Knoppie [2.1.4]

Skryf SQL-kode om al die voëls wat nog nie geassesseer is nie of wie se status 'Least concern' is, te verwyder. Die **StatusID** vir 'Not Assessed' is nommer **6** en die **StatusID** vir 'Least Concern' is nommer **5**.

(Kode is geskryf om die opdateerde tabel te vertoon nadat die gemaak is.)

Voorbeeld van afvoer van die eerste 9 rekords:

VoëlNaam	StatusID
African Broadbill	3
African Finfoot	3
African Grass Owl	3
African Marsh Harrier	2
African Rock Pipit	4
Agulhas Long-billed Lark	4
Antarctic Tern	2
Barlow's Lark	4
Bat Hawk	2

(4)

2.1.5 Knoppie [2.1.5]

Skryf SQL-kode om al die voëls se **StatusNaam** en **VoëlNaam** te vertoon.

Voorbeeld van afvoer van die eerste 8 rekords:

StatusNaam	VoëlNaam
Critically Endangered	Bearded Vulture
Critically Endangered	Blue Swallow
Critically Endangered	Damara Tern
Critically Endangered	Leach's Storm Petrel
Critically Endangered	Southern Banded S
Critically Endangered	Taita Falcon
Critically Endangered	Wattled Crane
Endangered	African Marsh Harri

(3)

2.1.6 Knoppie [2.1.6]

Skryf SQL-kode om die gemiddelde aantal waarnemings vir elke **StatusID** van al die voëls wat laas in die jaar 2007 **LaasteWaarneming** waargeneem is, sal vertoon. Die beskrywing moet 'GemiddeldeWaarnemings' wees. Rond die gemiddeld na 2 desimale plekke af.

Voorbeeld van afvoer:

StatusID	GemiddeldeWaarnemings
1	357.00
2	347.00
3	283.25
4	730.00

(8)

2.2 In hierdie afdeling mag slegs Delphi-programmeringskode gebruik word om VRAAG 2.2.1 en VRAAG 2.2.2 te beantwoord.

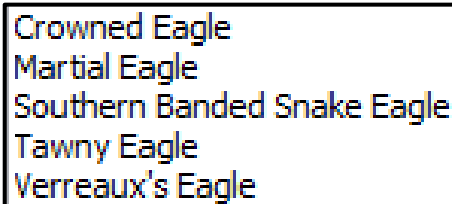
Gebruik die globale veranderlikes, **tblStatus** en **tblVoëls**, wat verskaf is.

GEEN punte sal vir SQL-stellings in VRAAG 2.2 toegeken word nie.

2.2.1 Knoppie [2.2.1]

Skryf kode wat al die voël name (**VoëlNaam**) wat die woord 'EAGLE' bevat in die 'richedit'-komponent, **redVertoon**, sal vertoon.

Voorbeeld van afvoer:



Crowned Eagle
Martial Eagle
Southern Banded Snake Eagle
Tawny Eagle
Verreaux's Eagle

(6)

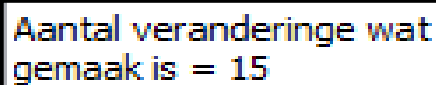
2.2.2 Knoppie [2.2.2]

Alle voëls wat as kwesbaar ('vulnerable') geklassifiseer is, se status moet na bedreig verander word.

Skryf kode om al die voëls wat 'n **StatusID** van 3 het, te verander na 'n **StatusID** van 2.

Tel hoeveel veranderinge gemaak is en vertoon die hoeveelheid in die **redVertoon** 'richedit'.

Voorbeeld van afvoer:



Aantal veranderinge wat
gemaak is = 15

(9)

- Sleutel jou naam en van as kommentaar in die eerste reël van die programlêer.
- Stoor jou program.
- 'n Drukstuk van jou kode mag vereis word.

[40]

VRAAG 3: OBJEK-GEÏORIENTEERDE PROGRAMMERING

Voëlkyk word gedefinieer as 'n ontspanningsaktiwiteit, deur mense wat voëlkykers genoem word, om wilde voëls in hul natuurlike habitat te identifiseer en waar te neem. Daar is baie wilde voëls wat deur voëlkykers in verskillende biologiese gemeenskappe in die Oos-Kaap gesien kan word.

Doen die volgende:

- Maak die onvoltooide program in die Vraag 3-lêergids oop.
- Maak die onvoltooide objekklas **Vraag3KlasDefinisie.pas** oop.
- Sleutel jou naam en van as kommentaar in beide die **Vraag3KlasDefinisie.pas** en **Vraag3_u.pas**.
- Kompileer en voer die program uit. Tans het die program geen funksionaliteit nie.
- MOENIE gegewe kode verander or verwyder NIE.

Die volgende gebruikerskoppelvlak word vertoon:

Voltooi die kode vir hierdie program, soos wat dit in VRAAG 3.1 en VRAAG 3.2 gespesifiseer is.

3.1 Die onvolledige klas (**TWaarneming**) bevat vyf attribute wat die **objVoëlkyker**-objek beskryf.

NAME VAN ATTRIBUTE	BESKRYWING
fNaam	Die naam van die persoon wat 'n voëlkyker is
fArea	Die naam van die area waar die voël die <u>eerste</u> keer gesien is
fVoëlNaam	Die naam van die voël wat gesien is
fDatum	Die datum wanneer die voël die eerste keer gesien is
fWaarnemings	Die totale waarnemings tot dusver

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

- 3.1.1 Skryf kode vir 'n **konstrukter** metode met die naam **Create** wat die naam van die voëlkyker, area, voël naam en die waarnemingsdatum as parameters ontvang.

Ken die parameterwaardes aan die korrekte attribute toe en inisialiseer die attribuut vir die waarnemings na EEN.

(4)

- 3.1.2 Skryf kode vir 'n metode, **getWaarnemings**, wat die aantal waarnemings terug stuur.

(2)

- 3.1.3 Skryf kode vir die **VermeerderHoeveelheid**-metode wat 'n een heelgetal as parameter sal ontvang en dit by die attribuut, **fHoeveelheid**, sal by tel.

(4)

- 3.1.4 Skryf kode vir 'n metode met die naam **WaarnemingsGaping** wat die aantal jare moet bereken sedert die eerste waarneming van die voël plaasgevind het. Gebruik die huidige datum en **fDatum**-attribuut om die aantal jare as deel van 'n string met die woorde 'jaar sedert eerste waarneming' aan die einde by te voeg.

Voorbeeld van afvoer: '23 jaar sedert eerste waarneming.'

(7)

- 3.1.5 Skryf kode vir 'n metode, **toString**, wat 'n string moet terugstuur.

Die string moet die attribute van die klas bymekaar voeg in die volgende formaat:

<persoon se naam> het 'n <voël naam>
<aantal jaar sedert eerste waarneming> by <area> op <datum van eerste waarneming> gesien

Voorbeeld van afvoer:

Joseph Mcina het 'n Cardinal Woodpecker
11 jaar sedert eerste waarneming by Cape Recife op 4/11/2010
gesien

(5)

3.2 'n Onvoltooide eenheid **Vraag3_u.pas** is verskaf.

Dit bevat kode vir die objekklas om toeganklik te wees. 'n Globale objekveranderlike, **objVoëlkyker**, is verklaar.

Moet Geen gegewe kode verander of verwyder NIE.

Volg die instruksies hieronder om die oplossing te kodeer:

3.2.1 Knoppie V3.2.1

Die gebruiker moet die voëlkyker se naam in tik, 'n waarnemingsarea kies, die voël naam en die datum wanneer die voël vir die eerste keer gesien is in die gegewe komponente, **edtVoëlkykerNaam**, **cmbArea**, **rgpVoëls**, **sedDag**, **sedMaand** en **sedJaar**, in tik.

Skryf kode om die volgende te doen:

- Die datum moet geskep word deur die jaar, maand en datum in die formaat 'Dag/Maand/Jaar' bymekaar te voeg. (Byvoorbeeld: 23/9/2012)
- Instansieer die objek, **objVoëlkyker**, deur die voëlkyker se naam, die waarnemingsarea, voël naam en die datum wanneer dit vir die eerste keer waargeneem is, te gebruik.
- Maak die 'richedit' skoon.

Gebruik 'n metode van die klas om die voëlkyker se naam, die voël se naam, die aantal jare sedert die eerste waarneming, die area van die waarneming en die datum van die eerste waarneming, te vertoon.

Voorbeeld van afvoer:

(11)

3.2.2 Knoppie V3.2.2

Skryf kode om die toevoer van die spin-redigeerblokkies ('spinedit'), **sedWaarnemings**, te kry.

Gebruik die metodes van die klas om die volgende te doen:

- Vermeerder die aantal waarnemings deur die waarde van die spin-redigeerblokkies te gebruik.
- Vertoon die woorde 'Totale waarnemings tot dusver', asook die aantal waarnemings in die richedit, **redVertoon**.

Voorbeeld van afvoer as 3 waarnemings by die hoeveelheid gevoeg is:

(5)

- Sleutel jou naam en van as kommentaar in die eerste reël van die programlêer. (In beide die klas en hoofprogram wat die klas gebruik.)
- Stoor jou programme.
- 'n Drukstuk van die kode van albei eenhede mag vereis word.

[38]

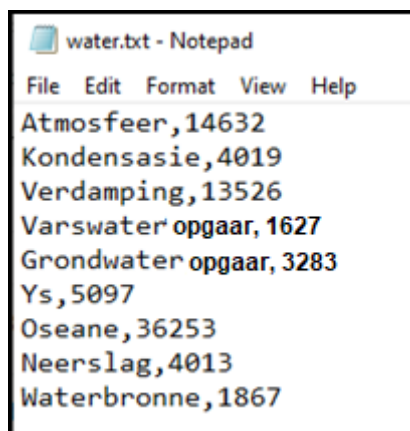
VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING

Tegnies is daar geen beginpunt van die watersiklus nie, maar op die vraag 'Waar begin die watersiklus?', het mense verskillende antwoorde. 'n Opname is gedoen met baie mense wat hierdie vraag beantwoord het. Gebruik die data uit hierdie opname om berekeninge uit te voer en die antwoorde te vertoon.

Doen die volgende:

- Maak die onvolledige program in die **Vraag 4** lêergids oop.
- Sleutel jou naam en van 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 tekslêer met die naam **water.txt**:



```
water.txt - Notepad
File Edit Format View Help
Atmosfeer,14632
Kondensasie,4019
Verdamping,13526
Varswater opgaar,1627
Grondwater opgaar,3283
Ys,5097
Oseane,36253
Neerslag,4013
Waterbronne,1867
```

Voltooi die kode vir elke vraag, VRAAG 4.1 en VRAAG 4.2.

LET WEL:

- Goeie programmeringsbeginsels moet in die ontwerp en kodering van jou oplossing toegepas word.
- MOENIE die kode wat verskaf is, verander NIE.

Gegee:

'n Globale skikking, **arrTipies**, is verklaar.

'n Globale skikking, **arrHoeveelheid**, is verklaar.

'n Globale teller, **iTeller**, is verklaar.

'n Globale veranderlike vir die totaal, **iTotaal**, is verklaar.

4.1 Knoppie V4.1

Skryf kode om die volgende te doen:

- Lees die reëls van die tekslêer, **water.txt** in die twee skikkings **arrTipes** en **arrHoeveelheid** deur die tipe metode te skei en in **arrTipes** te voeg en die aantal mense wat die metode gebruik, in **arrHoeveelheid** te stoor.
- Bereken die totale aantal mense wat 'n keuse gemaak het en stoor dit in die globale veranderlike, **iTotaal**.
- Sorteer die twee skikkings volgens die getalle in **arrHoeveelheid** van laag na hoog.
- Vertoon albei skikkings, in netjiese kolomme, in **redVertoon** asook die totale aantal mense wat deelgeneem het.

Voorbeeld van afvoer:

Varswater	1627
Waterbronne	1867
Grondwater	3283
Neerslag	4013
Kondensasie	4019
Ys	5097
Verdamping	13526
Atmosfeer	14632
Oseane	36253
Aantal mense = 84317	

(19)

4.2 Knoppie V4.2

Skryf kode om die persentasie van elke hoeveelheid in **arrHoeveelheid** te bereken deur die globale veranderlike **iTotaal** te gebruik. Rond die antwoord na geen desimale plekke af.

Voorbeeld: $\text{persentasie} = \text{hoeveelheid van 'n gekose tipe} / \text{totaal van al die keuses} \times 100$

Vertoon die persentasies as 'n simbool vir elk van die vier kategorieë.

Die VIER kategorieë bestaan uit die volgende:

- Persentasies tussen en gelyk aan 1 en 5.
- Persentasies tussen en gelyk aan 6 en 10.
- Persentasies tussen en gelyk aan 11 en 20.
- Persentasies tussen en gelyk aan 21 en 50.

Vertoon 'n simbool (x) in 'n netjiese tabel om die verskillende persentasie vir elke tipe van **arrTipes** voor te stel.

Voorbeeld van afvoer:

Tipe	Persentasies			
	1-5	6-10	11-20	21-50
Varswater	X			
Waterbronne	X			
Grondwater	X			
Neerslag	X			
Kondensasie	X			
Ys		X		
Verdamping			X	
Atmosfeer			X	
Oseane				X

(13)

- Sleutel jou naam en van as kommentaar in die eerste reël van die programlêer.
- Stoor jou program.
- 'n Drukstuk van jou kode mag vereis word.

[32]

TOTAAL: 150



Province of the
EASTERN CAPE
EDUCATION

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

SEPTEMBER 2021

**INLIGTINGSTEGNOLOGIE V1
NASIENRIGLYN**

PUNTE: 150

Hierdie nasienriglyn bestaan uit 16 bladsye.

NAAM VAN LEERDER:				
TOTAAL VRAAG 1:	TOTAAL VRAAG 2:	TOTAAL VRAAG 3:	TOTAAL VRAAG 4:	TOTAAL
/40	/40	/38	/32	/150

VRAAG 1: ALGEMENE PROGRAMMERINGSBEGINSELS		MAKS. PUNTE	PUNTE BEHAAL
1.1	Knoppie [1.1] Laai prentjie na prentjie lêer. ✓ Maak pnlBad sigbaar ✓ Verander die kleur van pnlReen na aqua ✓	3	
1.2	Knoppie [1.2 Verwerk] Kry reën toevoer van cmbReën of gebruik itemindeks ✓ Kry die bedrag in millimeter: Toets of reën toevoer 'n '/' bevat ✓ Indien nie 'n '/' in toevoer ✓ dan kry die getal en skakel om na real ✓ Else ✓ Verwerk die aantal millimeter korrek na 1/10 ✓ en 1/5 ✓ <i>Alternatiewe oplossing:</i> <i>Gebruik case met cmbReen itemindex om mm te verkry ✓</i> <i>0 : stel mm to 0.1 ✓</i> <i>1 : stel mm to 0.5 ✓</i> <i>Alle ander indekse is na korrekte mm's gestel ✓✓</i> <i>End vir case ✓</i> Kry toevoer van spinedits ✓ Bereken: wydte * hoogte * reentoever / 150 ✓ Vertoon in pnlBaddens ✓ afgerond na 1 desimale plek ✓	11	
1.3	Knoppie [1.3 Vind Reënvaltype] Toets edtReen vir heelgetal toevoer ✓ Vertoon 'n boodskap ✓ en exit ✓ as dit nie 'n heelgetal is nie ✓ <i>Alternatiewe oplossing:</i> <i>Gebruik val ✓</i> <i>As icode <> 0 ✓ dan showmessage ✓ en exit ✓</i> Kry toevoer van edtReën ✓ Gebruik case of nested if stellings ✓ 0 : lblReenTipe.Caption := 'Geen reën'; ✓ 1 .. 3 : lblReenTipe.Caption := 'Matige reën'; ✓ 4 .. 7 : lblReenTipe.Caption := 'Swaar reën'; ✓ 8 .. 9 : lblReenTipe.Caption := 'Baie swaar reën'; ✓ 10 .. 50 : lblReenTipe.Caption := 'Stortreën'; ✓ else ✓ lblReenTipe.caption := 'Vloed'; ✓	13	

1.4	Knoppie [1.4 Vertoon Reënval] Lus 8 keer ✓ Gebruik 'n lus ✓ om twee verskillende ✓ ewekansige getalle ✓ met omvang van 5 tot 10 ✓ te genereer As die ewekansige getal = 5 ✓ Voeg 1 by LstHoeveelheid indeks 0 ✓ As die ewekansige getal = 10 ✓ Voeg 1 by LstHoeveelheid indeks 1 ✓ Bereken totale hoeveelheid deur LstHoeveelheid items na integer om te skakel ✓ Vertoon die totaal in LstHoeveelheid indeks 2 omgeskakel na string ✓ Voeg LstHoeveelheid waardes by hoogtes van die twee Vorms ✓ Korrekte vorms met die korrekte items omgeskakel na Integer ✓	13	
TOTAL VRAAG 1		40	

VRAAG 2: DATABASISPROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
2.1.1	Knoppie: [2.1.1]	3	
	SQL: 'select * from Status order by StatusNaam DESC'		
	Konsepte: SELECT korrekte velde ✓ FROM korrekte tabel ✓ ORDER BY korrekte veld DESC ✓		
2.1.2	Knoppie: [2.1.2]	3	
	SQL: 'Select VoëlNaam from Voëls where Waarnemings < 200'		
	Konsepte: SELECT korrekte veld ✓ FROM korrekte tabel ✓ WHERE Waarnemings <200 ✓		
2.1.3	Knoppie: [2.1.3]	4	
	SQL: 'Select VoëlNaam, LaasteWaarneming from Voëls where VoëlNaam like ' + quotedstr('%' + sline + '%')		
	Konsepte: SELECT twee korrekte velde ✓ FROM korrekte tabel ✓ WHERE VoëlNaam LIKE ✓ quotedstr('%' + sline + '%') ✓		
2.1.4	Knoppie: [2.1.4]	4	
	SQL: 'Delete from Voëls where (StatusID = 6) OR (StatusID = 5)'		
	Konsepte: DELETE from Voels ✓ WHERE StatusID = 6 ✓ OR ✓ StatusID = 5 ✓		
2.1.5	Knoppie: [2.1.5]	3	
	SQL: 'select StatusNaam, VoëlNaam from Voëls, Status where Voëls.StatusID = Status.StatusID'		
	Konsepte: SELECT twee korrekte velde ✓ FROM twee korrekte tabelle ✓ WHERE om die tabelle te verbind ✓		
2.1.6	Knoppie: [2.1.6]	8	
	'Select StatusID,format(avg(Waarnemings),"fixed",1) as AverageWaarnemings from Voëls where year(LaasteWaarneming) = 2007 group by StatusID'		
	SELECT StatusID, ✓ format(avg(Waarnemings),"fixed",2) ✓ AS GemiddeldeWaarnemings ✓ FROM Voëls ✓ WHERE year(LaasteWaarneming) ✓ = 2007 ✓ GROUP by StatusID ✓		
2.1 Subtotaal: SQL		25	

2.2.1	Knoppie: [2.2.1] tblVoëls.First; ✓ while not tblVoëls.eof do ✓ if pos('EAGLE',uppercase(tblVoëls['VoëlNaam'] ✓)) <> 0 then ✓ redVertoon.Lines.Add(tblVoëls['VoëlNaam']); ✓ tblVoëls.Next; ✓	6	
2.2.2	Knoppie: [2.2.2] iTeller := 0; ✓ tblVoëls.First; while not tblVoëls.eof do ✓ begin if tblVoëls['StatusID'] = 3 then ✓ begin tblVoëls.Edit; ✓ tblVoëls['StatusID'] := 2; ✓ tblVoëls.Post; ✓ inc(iTeller) ✓ end; tblVoëls.Next; ✓ end; redVertoon.Lines.Add('Aantal veranderinge gemaak = ' + inttostr(iTeller)); ✓	9	
2.2 Subtotaal: Kode-konstruktor		15	
TOTAL VRAAG 2		40	

VRAAG 3: OBJEK-GEÖRIENTEERDE PROGRAMMERING		MAKS. PUNTE	PUNTE BEHAAL
3.1.1	Constructor Create: Korrekte naam ✓ met vier string parameters ✓ Stel attribute na korrekte parameter waardes ✓ Stel fWaarnemings na 1 ✓	4	
3.1.2	function getHoeveelheid: integer; Korrekte metode – integer funksie ✓ Stuur korrekte attribuut fHoeveelheid terug ✓	2	
3.1.3	procedure VermeerderHoeveelheid(iqty: integer); Korrekte metode – prosedure ✓ een integer parameter ✓ Voeg parameter ✓ by fHoeveelheid ✓	4	
3.1.4	function WaarnemingsGaping: string; Korrekte metode – string funksie ✓ Kry jaar ✓ van vandag se jaar ✓ Kry jaar van datum attribuut ✓ Trek attribuut jaar van huidige jaar af ✓ as integers ✓ Stuur resultaat met korrekte woorde bygevoeg ✓	7	
3.1.5	function toString: string; korrekte string metode ✓ Stuur kompilleerde resultaat terug met korrekte attribuut ✓ gebruik #13, ✓ korrekte woorde. ✓ Gebruik WaarnemingsGaping funksie ✓	5	
	3.1 Subtotaal: Objekklas	22	
3.2.1	Knoppie [Q3.2.1] Kry toevoere van twee edits ✓ Kry toevoer van kombinasieblokkie ✓ en radiogroupbox ✓ Kry toevoer van spinedits ✓ Instansieer die objek Objeknaam = ✓ tWaarneming.create ✓ met vier string parameters ✓ In korrekte volgorde ✓ Vertoon leë reël in richedit ✓ Gebruik metode van klas ✓ en objeknaam om te vertoon ✓	11	
3.2.2	Knoppie [Q3.2.2] Gebruik objeknaam ✓ en metode van die klas ✓ om waarnemings te vermeerder deur sedWaarnemings te gebruik ✓ Vertoon in richedit deur objeknaam ✓ en korrekte metode te gebruik ✓	5	
	3.2 Subtotaal: Vormklas	16	
TOTAAL VRAAG 3		38	

VRAAG 4: PROBLEEMOPLOSSINGPROGRAMMERING		MAKS. MARKS	PUNTE BEHAAL
4.1	<pre> iTeller := 0; iTotaal := 0; ✓ assignfile(myfile, 'water.txt'); ✓ reset(myfile); ✓ while not eof(myfile) do ✓ readln(myfile, soneline); ✓ inc(iTeller); ✓ Voeg tipe ✓ by arrTipes ✓ Voeg hoeveelheid by arrHoeveelheid ✓ iTotaal := iTotaal + arrHoeveelheid[iTeller]; ✓ for k := 1 to iTeller - 1 do ✓ for l := k + 1 to iTeller do ✓ if arrHoeveelheid[k] > arrHoeveelheid[l] then ✓ itemp := arrHoeveelheid[k]; arrHoeveelheid[k] := arrHoeveelheid[l]; } ✓ arrHoeveelheid[l] := itemp; stemp := arrTipes[k]; arrTipes[k] := arrTipes[l]; } ✓ arrTipes[l] := stemp; for k := 1 to iTeller do ✓ redVertoon.Lines.Add(arrTipes[k] ✓ + #9 + inttostr(arrHoeveelheid[k])); ✓ redVertoon.Lines.Add('Totale aantal mense = ' + inttostr(iTotaal)) ✓ </pre>	19	
4.2	<pre> for k := 1 to iTeller do ✓ iperc := round ✓ (arrHoeveelheid[k] ✓ / iTotaal * 100 ✓); arrperc[k] := iperc; ✓ for k := 1 to iTeller do ✓ sline := arrTipes[k] + #9; ✓ case arrperc[k] of ✓ (case korrek gebruik) 1..5 : sline := sline + 'X'; ✓ 6..10 : sline := sline + #9 + 'X'; ✓ 11..20 : sline := sline + #9 + #9 + 'X'; ✓ 21..50 : sline := sline + #9 + #9 + #9 + 'X'; ✓ end; redVertoon.lines.add(sline); binne die lus ✓ </pre>	13	
TOTAAL VRAAG 4		32	

MOONTLIKE OPLOSSINGS**VRAAG 1**

```
procedure TfrmVraag1.btnV1_1Click(Sender: TObject);
begin
imgReen.picture.LoadFromFile('reen.jpg');
pnlBad.enabled := true;
pnlrain.Color := claqua;
end;

procedure TfrmVraag1.btnV1_2Click(Sender: TObject);
var
rrain : real;
srain : string;
begin
srain := cmbReen.Text;
if copy(srain,2,1) <> '/' then
begin
rrain := strtofloat(copy(srain,1, pos(' ',srain) - 1));
end
else
if copy(srain,3,1) = '1' then
rrain := 0.1
else
rrain := 0.5;

pnlBaddens.Caption := floattostrf(sedHoogte.value * sedWydte.value * rrain / 150,
ffixed,8,1);

end;

procedure TfrmVraag1.btnV1_3Click(Sender: TObject);
var irain, inum, icode : integer;
begin
val(edtReen.Text, inum, icode);
if icode <> 0 then
begin
showmessage('Tik slegs "n heelgetal in');
exit;
end;
irain := strtoint(edtReen.Text);
case irain of
0 : lblReentipe.Caption := 'Geen reen';
1 .. 3 : lblReentipe.Caption := 'Matige reen';
4 .. 7 : lblReentipe.Caption := 'Swaar reen';
8 .. 9 : lblReentipe.Caption := 'Baie swaar reen';
10 .. 50 : lblReentipe.Caption := 'Stortreen';
else
lblReentipe.caption := 'Vloed';
end;

end;
```

```
procedure TfrmVraag1.btnV1_4Click(Sender: TObject);
var iran1, iran2, k, itotal: integer;
begin
for k := 1 to 8 do
  begin
  repeat
  iran1 := randomrange(5,11);
  iran2 := randomrange(5,11);
  until (iran1 <> iran2);
  if iran1 = 5 then
    LstHoeveelheid.Items[0] := inttostr(strtoint(LstHoeveelheid.Items[0])+1);
  if iran2 = 10 then
    LstHoeveelheid.Items[1] := inttostr(strtoint(LstHoeveelheid.Items[1])+1);
  end;
  itotal := strtoint(LstHoeveelheid.Items[0]) + strtoint(LstHoeveelheid.Items[1]);
  LstHoeveelheid.Items[2] := inttostr(itotal);
  shp5mm.Height := shp5mm.Height + strtoint(LstHoeveelheid.Items[0]);
  shp10mm.Height := shp10mm.Height + strtoint(LstHoeveelheid.Items[1]);
end;

//gegeve kode - moenie verwyder nie//////////
procedure TfrmVraag1.FormActivate(Sender: TObject);
begin
LstHoeveelheid.Items[0] := inttostr(0);
LstHoeveelheid.Items[1] := inttostr(0);
LstHoeveelheid.Items[2] := inttostr(0);
end;
//////////
end.
```

VRAAG 2

```
//=====
// Vraag 2.1.1 3 punte
//=====
procedure TfrmVraag2.btnVraag2_1_1Click(Sender: TObject);
var
  sSQL1: String;
begin
  sSQL1 := 'select * from Status order by StatusNaam DESC';
  // Gegewe kode - moenie verander nie/
  dbCONN.runSQL(sSQL1);

end;

//=====
// Vraag 2.1.2 3 punte
//=====
procedure TfrmVraag2.btnVraag2_1_2Click(Sender: TObject);
// Gegewe kode - moenie verander nie/
var
  sSQL2: String;
begin
  // Gegewe kode - moenie verander nie/

  sSQL2 := 'Select VoëlNaam from Voëls where Waarnemings < 200';

  // Gegewe kode - moenie verander nie/
  dbCONN.runSQL(sSQL2);
end;

//=====
// Vraag 2.1.3 4 punte
//=====
procedure TfrmVraag2.btnVraag2_1_3Click(Sender: TObject);
// Gegewe kode - moenie verander nie/
var
  sline : string;
  sSQL3: String;
begin
  sline := inputbox('Tik "voël naam in",', 'vulture');

  sSQL3 := 'Select VoëlNaam, LaasteWaarneming from Voëls where VoëlNaam like ' +
quotedstr('%' + sline + '%)';

  // Gegewe kode - moenie verander nie/
  dbCONN.runSQL(sSQL3);
end;
```

```
//=====
//  Vraag 2.1.4  4 punte
//=====
procedure TfrmVraag2.btnVraag2_1_4Click(Sender: TObject);
// Gegewe kode - moenie verander nie/

var
    sSQL4: String;
begin

    sSQL4 := 'Delete from Voëls where (StatusID = 6) OR (StatusID = 5)';

// Gegewe kode - moenie verander nie/

    dbCONN.executeSQL(sSQL4,dbgstatus,dbgVoël,dbgQryVoël);
end;

//=====
//  Vraag 2.1.5  3 punte
//=====
procedure TfrmVraag2.btnVraag2_1_5Click(Sender: TObject);
// Gegewe kode - moenie verander nie/

var
    sSQL5: String;
begin

    sSQL5 := 'select StatusNaam, VoëlNaam from Voëls, Status where Voëls.StatusID =
Status.StatusID' ;

// Gegewe kode - moenie verander nie/
    dbCONN.runSQL(sSQL5);
end;

//=====
//  Vraag 2.1.6  8 punte
//=====
procedure TfrmVraag2.btnVraag2_1_6Click(Sender: TObject);
// Gegewe kode - moenie verander nie/

var
    sSQL6: String;
begin
    sSQL6 := 'Select StatusID,format(avg(Waarnemings),"fixed",1) as
GemiddeldeWaarnemings from Voëls where year(LaasteWaarneming) = 2007 group by
StatusID';
// Gegewe kode - moenie verander nie/

    dbCONN.runSQL(sSQL6);

end;
```

```
//=====
//  Vraag 2.2.1  6 punte
//=====
procedure TfrmVraag2.btnVraag2_2_1Click(Sender: TObject);
begin
redVertoon.Clear;
/// tik jou kode hier//
tblVoëls.First;
while not tblVoëls.eof do
begin
if pos('EAGLE',uppercase(tblVoëls['VoëlNaam'])) <> 0 then
redVertoon.Lines.Add(tblVoëls['VoëlNaam']);
tblVoëls.Next;
end;

end;

//=====
//  Vraag 2.2.2  9 punte
//=====
procedure TfrmVraag2.btnVraag2_2_2Click(Sender: TObject);
var icount : integer;
begin
redVertoon.Clear;
/// tik jou kode hier//
icount := 0;
tblVoëls.First;
while not tblVoëls.eof do
begin
if tblVoëls['StatusID'] = 3 then
begin
tblVoëls.Edit;
tblVoëls['StatusID'] := 2;
tblVoëls.Post;
inc(icount)
end;
tblVoëls.Next;
end;
redVertoon.Lines.Add('Aantal veranderinge wat gemaak is = ' + inttostr(icount));

end;
```

VRAAG 3**Klaseenheid:**

```
unit Vraag3KlasDefinisie;
interface
uses sysutils, dialogs, math;
type
TWaarneming = class (tobject)
private
  fNaam : string;
  fArea : string;
  fVoëlNaam : string;
  fDatum : string;
  fWaarnemings : integer;
public
  constructor create(sNaam,sArea,sVoëlNaam,sDatum : string);
  procedure VermeerderWaarnemings(ihoeveel : integer);
  function WaarnemingsGaping : string;
  function toString : string;
  function getWaarnemings : integer;
end;

implementation

constructor TWaarneming.create(sNaam,sArea,sVoëlNaam,sDatum: string);
begin
  fNaam := sVoëlkyker;
  fArea := sArea;
  fVoëlNaam := sVoëlNaam;
  fDatum := sDatum;
  fHoeveelheid := 1;
end;

function TWaarneming.getWaarnemings : integer;
begin
  result := fHoeveelheid;
end;

procedure TWaarneming.VermeerderfHoeveelheid(ihoeveel: integer);
begin
  fhoeveelheid := fHoeveelheid + ihoeveel;
end;
```

```

function TWaarneming.WaarnemingsGaping: string;
var svandag, sjaar : string;
begin
svandag := datetostr(date);
svandag := copy(svandag,1,4);

sjaar := fDatum;
delete(sjaar,1,pos('/',sjaar));
delete(sjaar,1,pos('/',sjaar));
result := inttostr(strtoint(svandag) - strtoint(sjaar)) + ' years since first sighting';
end;

function TWaarneming.tostring: string;
begin
result :=
fNaam + ' first viewed a ' + fVoëlNaam + #13
+ WaarnemingsGaping + ' at ' + fArea + ' on ' + fDatum;
end;

end.

```

Hoofteenheid:

```

/// gegewe kode, moenie verwyder nie ///
var
    frmVraag3: TfrmVraag3;
    objVoelkyker: TWaarneming;
implementation

{$R *.dfm}

procedure TfrmVraag3.btnV3_2_1Click(Sender: TObject);
var
    sNaam, sArea, sVoël , sDatum: string;
    iDag, iMaand, iJaar : integer;
begin
    sNaam := edtVoelkykerNaam.text;
    sArea := cmbArea.Text;
    sVoel := rgpVoëls.Items[rgpVoëls.ItemIndex];
    sDatum := inttostr(sedDag.Value) + '/' + inttostr(sedMaand.Value) + '/' +
inttostr(sedJaar.Value);
    objVoëlkyker := TWaarneming.create(sNaam, sArea, sVoël, sDatum);
    redVertoon.lines.add("");
    redVertoon.lines.add(objVoelkyker.tostring);

end;

procedure TfrmVraag3.btnV3_2_2Click(Sender: TObject);
begin
objVoëlkyker.VermeerderWaarnemings(sedWaarnemings.value);
redVertoon.Lines.Add('Totale waarnemings tot dusver: ' +
inttostr(objVoëlkyker.getWaarnemings));

end;

```

VRAAG 4

```

var
  frmVraag4: TfrmVraag4;
//gegeve kode - moenie verander nie/////
  arrTypes : array[1..100] of string;
  arrHoeveelheid : array[1..100] of integer;
  iTeller, iTotaal : integer;

implementation
// Vraag 4.1 19 marks
procedure TfrmVraag4.btnV4_1Click(Sender: TObject);
var
  myfile : textfile;
  soneline : string;
  k,l : integer;
  stemp : string;
  itemp : integer;
begin
  // provided code do not delete///
  redVertoon.Clear;
  redVertoon.Paragraph.TabCount := 1;
  redVertoon.Paragraph.Tab[0] := 100;
  //////////////////////////////////////
  iTeller := 0;
  iTotaal := 0;
  assignfile(myfile, 'water.txt');
  reset(myfile);
  while not eof(myfile) do
    begin
      readln(myfile, soneline);
      inc(iTeller);
      arrTypes[iTeller] := copy(soneline,1, pos(',', soneline) - 1);
      delete(soneline,1, pos(',', soneline));
      arrHoeveelheid[iTeller] := strtoint(soneline);
      iTotaal := iTotaal + arrHoeveelheid[iTeller];
    end;

  for k := 1 to iTeller - 1 do
    for l := k + 1 to iTeller do
      if arrHoeveelheid[k] > arrHoeveelheid[l] then
        begin
          itemp := arrHoeveelheid[k];
          arrHoeveelheid[k] := arrHoeveelheid[l];
          arrHoeveelheid[l] := itemp;
          stemp := arrTypes[k];
          arrTypes[k] := arrTypes[l];
          arrTypes[l] := stemp;
        end;
    for k := 1 to iTeller do
      redVertoon.Lines.Add(arrTypes[k] + #9 + inttostr(arrHoeveelheid[k]));
  redVertoon.Lines.Add("");
  redVertoon.Lines.Add('Total number of people = ' + inttostr(iTotaal))
end;

```

```

// Vraag 4 .2
procedure TfrmVraag4.btnV4_2Click(Sender: TObject);
var iperc : integer;
arrperc : array[1..100] of integer;
k : integer;
sline : string;
begin
//gegewe kode - moenie verwyder nie //////////////////////////////////
redVertoon.Clear;
redVertoon.Paragraph.TabCount := 4;
redVertoon.Paragraph.Tab[0] := 100;
redVertoon.Paragraph.Tab[1] := 140;
redVertoon.Paragraph.Tab[2] := 180;
redVertoon.Paragraph.Tab[3] := 220;
redVertoon.Lines.Add(#9 +'Persentasies');

redVertoon.lines.add('Tipe' + #9 + '1-5' + #9 + '6-10' + #9 + '11-20' + #9 + '21-50');
redVertoon.Lines.Add('-----');
////////////////////////////////////

for k := 1 to iTeller do
begin
    iperc := round(arrHoeveelheid[k]/iTotaal *100);
    arrperc[k] := iperc;
end;

for k := 1 to iTeller do
begin
sline := arrTypes[k] + #9;
    case arrperc[k] of
        1..5 : sline := sline + 'X';
        6..10 : sline := sline + #9 + 'X';
        11..20 : sline := sline + #9 + #9 + 'X';
        21..50 : sline := sline + #9 + #9 + #9 + 'X';
    end;
    redVertoon.lines.add(sline);
end;
end;

```