



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

FEBRUARIE/MAART 2016

PUNTE: 150

TYD: 3 uur

Hierdie vraestel bestaan uit 21 bladsye.

INSTRUKSIES EN INLIGTING

1. Hierdie vraestel is in DRIE afdelings ingedeel. Kandidate moet AL DRIE 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 nie programmeringstaal-spesifiek is nie (Delphi/Java (deur die Netbeans IDE te gebruik)).
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 in elke vraag slegs wat gevra word. Byvoorbeeld, indien die vraag nie datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
6. Jou programme moet op só 'n manier gekodeer word dat dit met enige data sal werk en nie net met die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel verskyn nie.
7. **Roetines soos soek, sorteer en seleksie moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van 'n programmeringstaal vir enige van hierdie roetines gebruik NIE.**
8. Datastrukture wat nie voorsien is nie, moet deur jou, die programmeerder, verklaar 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 eksamennommer as kommentaar verskyn in elke program wat jy kodeer, asook op elke gebeurtenis wat aangedui word.
11. Indien dit vereis word, druk die programmeringskode van al die programme/klasse wat jy voltooi het. Jy sal ná die eksamensessie 'n halfuur tyd vir drukwerk gegun 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 is in die vorm van wagwoordbeskernde uitvoerbare lêers verskaf.

LET WEL:

- Delphi-kandidate moet die lêer **DelphiDataAFRMaart2016.exe** gebruik.
- Java-kandidate moet die lêer **JavaDataAFRMaart2016.exe** gebruik.

Doen die volgende:

- Dubbelklik op die wagwoordbeskernde uitvoerbare lêer.
- Klik op die 'extract'-knoppie.
- Sleutel die volgende wagwoord in: **MountainP\$%#**

Nadat dit onttrek is, sal die volgende lys met lêers in die **DelphiDataAFRMaart2016/JavaDataAFRMaart2016**-lêergids beskikbaar wees:

Delphi-lêers**Vraag1:**

Vraag1P.dpr
Vraag1P.res
Vraag1U.dfm
Vraag1U.pas

Vraag2:

BergpasU.pas
Gamkaskloofpas.jpg
GeenKaart.jpg
Prins Alfred-pas.jpg
Vraag2P.dpr
Vraag2P.res
Vraag2U.dfm
Vraag2U.pas

Vraag3:

V3Data.txt
Vraag3P.dpr
Vraag3P.res
Vraag3U.dfm
Vraag3U.pas

Java (Netbeans)-lêers**Vraag1:**

Vraag1.form
Vraag1.java

Vraag2:

Bergpas.java
Gamkaskloofpas.jpg
GeenKaart.jpg
Prins Alfred-pas.jpg
Vraag2.form
Vraag2.java

Vraag3:

V3Data.txt
Vraag3.form
Vraag3.java

SCENARIO

Suid-Afrika word as een van die skilderagtigste lande in die wêreld beskou en lok baie toeriste na plaaslike vakansiebestemmings en internasionaal bekroonde bergpasse. Sagtewareprogramme word ontwerp om met toeriste-administrasie te help.

AFDELING A**VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

'n Verskeidenheid aspekte oor bergpasse en vakansiebestemmings word in hierdie vraag hanteer.

INSTRUKSIES:

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"> Die projek Vraag1 is in die DelphiDataAFRMaart2016-lêergids voorsien. Maak die onvoltooide projeklêer Vraag1P.dpr in die Vraag1-lêergids oop. Voeg jou eksamennummer as kommentaar in die eerste reël van die hoofvormeenheid Vraag1U.pas-lêer by. 	<ul style="list-style-type: none"> Die projek Vraag1 is in die JavaDataAFRMaart2016-lêergids voorsien. Maak die onvoltooide klas Vraag1.java wat in die lêergidse Source Packages (src), Vraag1Package voorkom, oop. Voeg jou eksamennummer as kommentaar in die eerste reël van die klas Vraag1.java by.

Doen die volgende:

- Kompileer en voer die program uit. Die GGK ('GUI') vertoon vyf afdelings met die opskrifte VRAAG 1.1 tot VRAAG 1.5. Die program het tans geen funksionaliteit nie. 'n Voorbeeld van die GGK ('GUI') word op die volgende bladsy gegee:

Vraag 1.1
Afstand van bergpas in myl 145.87
Vraag 1.1
Afvoer

Vraag 1.2
Hoogte van bergpas in meter
Verandering in temperatuur van 1000 m af

Vraag 1.3
Vraag 1.3
Laagste bergpas(se)

Vraag 1.4
Tipe akkommodasie
Getal mense
Betaalopsies
 Wi-Fi
Kontant
Kaart
Betaling per nag

Vraag 1.5
Transaksienommer
Pryse
Geen prys
Geskenkbewys vir middagete
Geskenkbewys vir toer
T-hemp
Waterbottel

- Voltooi die kode vir elke afdeling van VRAAG 1 soos in VRAAG 1.1 tot VRAAG 1.5 hieronder beskryf word.

1.1 Afstande word op sommige padkaarte in myl vertoon. Die gebruiker moet die totale afstand van 'n bergpas in myl insleutel.

Skryf kode om die afstand wat in myl ingesleutel is, in kilometer om te skakel. Vertoon die omgeskakelde waarde en eenheid (km) in die afvoercomponent wat verskaf is.

LET WEL: 1 myl = 1,6 kilometer

Voorbeeld van afvoer:

Afstand van bergpas in myl 145.87
Vraag 1.1
Afvoer 233.39 km

(3)

- 1.2 Baie toeriste stel belang in die daling in temperatuur terwyl hulle teen 'n bergpas uity. Die gebruiker moet die hoogte van die bergpas in meter insleutel. Die temperatuur by die 1 000 meter-punt op die bergpas is 18 °C. Die temperatuur verminder met 1 °C met elke 100 m-styging in die hoogte van die bergpas.

LET WEL: Die hoogte van die bergpas wat jy insleutel, moet hoër as of gelyk aan 1 000 m wees.

Skryf kode om die volgende te doen:

- Verkry die hoogte van die bergpas uit die teksblokkie.
- Valideer die waarde wat ingesleutel is soos volg:
 - Indien die waarde kleiner as 1 000 is, vertoon 'n gepaste boodskap, maak die teksblokkie skoon en stel die fokus op die teksblokkie.
 - Indien die waarde groter as of gelyk aan 1 000 is, bereken en vertoon die temperatuur vir elke 100 m wat daar bo 1 000 m teen die bergpas uitgery word tot die naaste 100 m onder die bopunt van die pas.

Voorbeeld van die afvoer indien die hoogte van die pas 1 685 m is:

Meter	Temperatuur
1000	18
1100	17
1200	16
1300	15
1400	14
1500	13
1600	12

(11)

- 1.3 Sommige toeriste wil teen die laagste bergpas uity. Twee parallelle skikkings word verskaf. Die skikking met die naam **arrPasName** bevat die name van bergpasse, terwyl die skikking met die naam **arrPasHoogtes** die hoogtes van die bergpasse wat in die **arrPasName**-skikking voorkom, bevat.

Skryf kode om die naam van die bergpas met die laagste hoogte te bepaal.

LET WEL: Jy sal gepenaliseer word as jy vaste waardes in jou kode gebruik om die getal elemente in die skikkings aan te dui. Alhoewel die skikkings verskaf word en die getal elemente bekend is, moet 'n gepaste ingeboude funksie/metode gebruik word om die getal elemente in die skikkings te bepaal.

Voorbeeld van afvoer:

Laagste bergpas(se)
Die laagste bergpas is Baviaanskloofpas. Die hoogte van die bergpas is 986.0 m.

(6)

1.4 Toeriste kan akkommodasie naby enige van die bergpasse bespreek. Die tipes akkommodasie wat beskikbaar is en die prys per persoon per nag word in die tabel hieronder gegee.

TIPE AKKOMMODASIE	PRYS PER PERSOON PER NAG
Hotel	R1 200
Bed-en-ontbyt	R1 000
Selfsorgeenheid	R750
Kampeerplek	R300

Die prys van Wi-Fi is R150 per nag.

Skryf kode om die volgende te doen:

- Verkry die tipe akkommodasie uit die kombinasie lys ('combo box').
- Verkry die getal mense wat akkommodasie vir hierdie bespreking nodig uit die teksblokkie.
- Bepaal of die merkblokkiekontrolle ('check box control') vir toegang tot Wi-Fi gekies is.
- Bepaal die koste per nag volgens die waardes wat deur die gebruiker ingesleutel is en die waardes in die tabel hierbo.
- Betaalopsies:
 - Indien die gebruiker die 'Kontant'-opsie kies, vertoon die betaling per nag, in geldeenheidsformaat met TWEE desimale plekke, in die teksblokkie wat verskaf is.
 - Indien die gebruiker die 'Kaart'-opsie kies, doen die volgende:
 - Gebruik 'n dialoogblokkie ('dialog box') om die kaartnommer in te sleutel.
 - Doen die volgende TWEE toetse om seker te maak dat die kaartnommer geldig is:
 - Maak seker dat die kaartnommer uit 9 syfers bestaan.
 - Maak seker dat die kaartnommer slegs uit syfers bestaan.

- Indien die kaartnommer geldig is:
 - Voeg 'n kaarthefing van 3% by die koste.
 - Vertoon die betaling per nag in die teksblokkie wat verskaf is.
- Indien die kaartnommer NIE geldig is NIE:
 - Gebruik 'n dialoogblokkie ('dialog box') om 'n boodskap te vertoon dat die kaartnommer ongeldig is.
 - Stel die radioknoppie ('radio button') op die 'Kontant'-opsie.
 - Vertoon die kontantbedrag as die betaling per nag.

Voorbeeld van die betaling per nag vir die data vertoon in die skermkopie hieronder:

The screenshot shows a form with the following fields and options:

- Tipe akkommodasie: Selfsorgeenheid (dropdown menu)
- Getal mense: 4 (text input)
- Wi-Fi: (checkbox)
- Betaalopsies:
 - Kontant (radio button)
 - Kaart (radio button)
- Betaling per nag: R 3 150.00 (text input)
- Vraag 1.4 (button)

Voorbeeld van die betaling per nag vir die data soos vertoon in die skermkopie hieronder. Die geldige kaartnommer 789674556 is in die dialoogblokkie ('dialog box') ingesleutel.

The screenshot shows a form with the following fields and options:

- Tipe akkommodasie: Hotel (dropdown menu)
- Getal mense: 5 (text input)
- Wi-Fi: (checkbox)
- Betaalopsies:
 - Kontant (radio button)
 - Kaart (radio button)
- Betaling per nag: R 6 334.50 (text input)
- Vraag 1.4 (button)

(19)

- 1.5 Tolgeld, wat gebruik word om die bergpasse in stand te hou, moet op bergpasroetes betaal word. Die maatskappy wat die bergpasse in stand hou, bied 'n promosie aan om toeriste te lok om met die bergpasse langs te ry. Elke tolhekkwitansie het 'n unieke transaksienommer. Die maatskappy gebruik hierdie nommers om pryse weg te gee. Die lysblokkie ('list box') wat voorsien is, bevat die pryse.

Die gebruiker moet die transaksienommer wat op 'n tolhekkwitansie verskyn, insleutel.

Die volgende kriteria word gebruik om prysweners te bepaal:

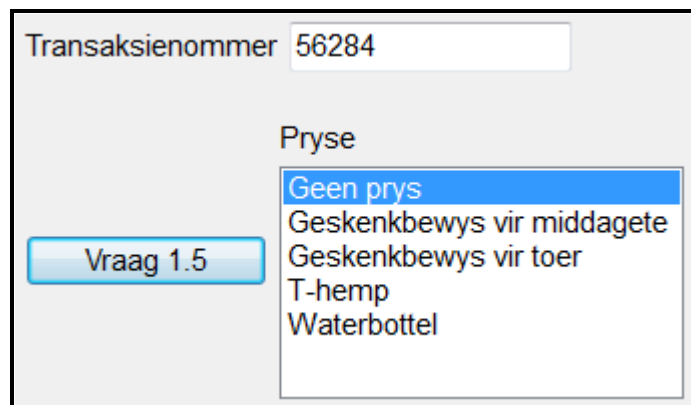
- Indien die transaksienommer 'n priemgetal is, sal die persoon 'n prys wat ewekansig uit die lys van pryse in die gegewe lysblokkie ('list box') gekies is, ontvang.
- Indien die transaksienommer NIE 'n priemgetal is NIE, sal die persoon NIE 'n prys ontvang NIE.

Gebruik die algoritme wat hieronder verskaf is om die oplossing te kodeer.

Algoritme:

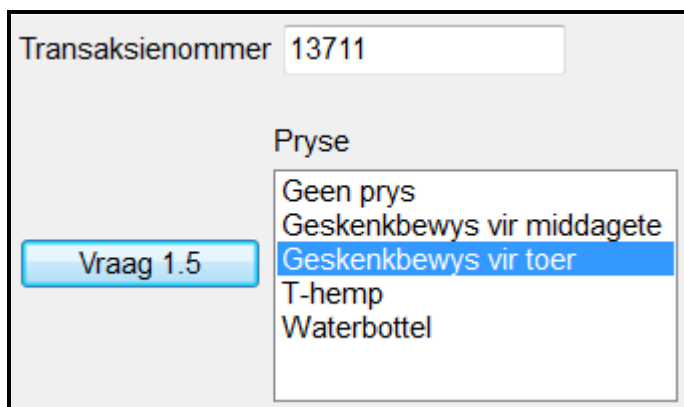
1. Verkry die transaksienommer uit die teksblokkie.
2. Bepaal of die transaksienommer 'n priemgetal is.
('n Priemgetal is 'n getal met net TWEE faktore, naamlik 1 en die getal self.)
3. Indien die transaksienommer 'n priemgetal is, doen die volgende:
 - 3.1 Genereer 'n ewekansige getal in die reeks 1 tot 4 (1 en 4 ingesluit). Die ewekansige getal dui die indeks van die prys in die lysblokkie ('list box') aan, wat die prys is wat aan die wenner toegeken sal word.
 - 3.2 Kies die ewekansig gegenereerde indeks in die lysblokkie ('list box') om die prys uit te lig ('highlight').
4. Indien die transaksienommer NIE 'n priemgetal is NIE, kies die 'Geen prys'-opsie in die lysblokkie ('list box').

Voorbeeld van afvoer vir transaksienommer 56284, wat NIE 'n priemgetal is NIE:



The screenshot shows a web application interface. At the top, there is a text input field labeled "Transaksienommer" containing the value "56284". Below this, there is a button labeled "Vraag 1.5". To the right of the button, there is a dropdown menu titled "Pryse". The dropdown menu is open, showing a list of options: "Geen prys" (highlighted in blue), "Geskenkbewys vir middagete", "Geskenkbewys vir toer", "T-hemp", and "Waterbottel".

Voorbeeld van afvoer vir transaksienommer 13711, wat 'n priemgetal is, en 2 as die ewekansig gegenereerde getal:



The screenshot shows a window with a text input field containing '13711' and a label 'Transaksienommer'. Below it is a button labeled 'Vraag 1.5'. To the right, under the label 'Pryse', is a list of prizes: 'Geen prys', 'Geskenkbewys vir middagete', 'Geskenkbewys vir toer', 'T-hemp', and 'Waterbottel'. The 'Geskenkbewys vir toer' option is highlighted in blue.

LET WEL: As gevolg van die aard van die ewekansige ('random') funksie/metode, kan die afvoer wat deur jou projek/program gegenereer word van die voorbeeld hierbo verskil.

(9)

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

TOTAAL AFDELING A: 48

AFDELING B**VRAAG 2: OBJEK-GEORIËNTEERDE PROGRAMMERING**

'n Groep bergpasse is gekies om spesifieke statistieke, soos die getal reisigers, boetes deur reisigers ontvang en besonderhede oor die gevaarvlakke van die bergpasse, te bepaal.

INSTRUKSIES:

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"> • Die projek Vraag2, wat in die DelphiDataAFRMaart2016-lêergids voorsien is, bevat: <ul style="list-style-type: none"> ○ 'n Hoofvormeenheid Vraag2U.pas ○ 'n Onvoltooide eenheidlêer BergpasU.pas • Maak die onvoltooide projeklêer Vraag2P.dpr en die klaseenheid-lêer BergpasU.pas in die Vraag2-lêergids oop. • Voeg jou eksamennummer as kommentaar in die eerste reël van beide lêers Vraag2U.pas en BergpasU.pas by. 	<ul style="list-style-type: none"> • Die projek Vraag2, wat in die JavaDataAFRMaart2016-lêergids voorsien is, bevat: <ul style="list-style-type: none"> ○ 'n GGK ('GUI')-klas Vraag2.java ○ 'n Onvoltooide objekklaslêer Bergpas.java • Maak die onvoltooide klasse Vraag2.java en Bergpas.pas in die lêergidse Source Packages (src), Vraag2Package oop. • Voeg jou eksamennummer as kommentaar in die eerste reël van die klasse Vraag2.java en Bergpas.java by.

Doen die volgende:

- Voltooi die kode vir elke afdeling van VRAAG 2 soos in VRAAG 2.1 en VRAAG 2.2 hieronder beskryf word.

2.1 'n Objekklas met die naam **TBergpas/Bergpas** met 'n paar metodes is voorsien.

Voltooi die kode in die gegewe klas (**TBergpas/Bergpas**) soos wat in VRAAG 2.1.1 tot VRAAG 2.1.6 beskryf word.

Die gegewe attribute van die klas **TBergpas/Bergpas** word in die tabel hieronder beskryf.

NAME VAN ATTRIBUTE		BESKRYWING
Delphi	Java	
fNaam	naam	Die naam van die bergpas
fAfstand	afstand	Afstand in kilometer
fGevaarVlak	gevaarVlak	Laag/Medium/Hoog
fGradient	gradient	Die gradiënt van die steilste deel van die pas in grade, byvoorbeeld 25 grade

2.1.1 Skryf kode vir 'n konstruktormetode wat die naam van 'n bergpas, die afstand van die pas in kilometer, die gevaarvlak en die gradiënt van die pas as parameters ontvang.

Gebruik die parameters om die attribute van die objekklas te inisialiseer.

(2)

2.1.2 Skryf 'n wysigingsmetode ('mutator method') met die naam **setGevaarVlak** wat die **fGevaarVlak/gevaarVlak**-attribuut na die gevaarvlak wat as 'n parameter ontvang is, opdateer.

(2)

2.1.3 Skryf 'n metode met die naam **berekenBoete** wat die tyd in **minute** wat dit die toeris geneem het om teen die bergpas uit te ry en die spoedbeperking in **kilometer per uur** wat op hierdie bergpas van toepassing is, as parameters ontvang. Die metode moet die gemiddelde spoed waarteen gery is, bereken en bepaal of die spoedbeperking oorskry is. Stuur 'n waarde van nul terug indien die spoedbeperking NIE oorskry is NIE. Indien die spoedbeperking wel oorskry is, bereken en stuur die boete wat betaal moet word, terug.

Die boete word soos volg bereken:

- Indien die gemiddelde spoed die spoedbeperking met minder as 10 km/h oorskry, is die boete nul.
- Indien die gemiddelde spoed die spoedbeperking met presies 10 km/h oorskry, is die basiese boete R500,00.
- Indien die gemiddelde spoed die spoedbeperking met meer as 10 km/h oorskry, word R100 addisioneel by die basiese boete van R500,00 getel vir elke 3 km/h of gedeelte daarvan wat die spoedbeperking met meer as 10 km/h oorskry is.

LET WEL: Gemiddelde spoed in km/h = $\frac{\text{afstand in kilometer}}{\text{tyd in uur}}$

Voorbeelde van hoe om boetes te bereken, word op die volgende bladsy gegee.

Voorbeeld 1:

Spoedbeperking = 80 km/h

Gemiddelde spoed waarteen gery is = 96 km/h

Berekening van die boete: R500,00 (vir oorskryding van die spoedbeperking met 10 km/h) + (2 x R100,00 vir die addisionele 6 km/h waarteen die gemiddelde spoed waarteen gery is die 10 km/h-speling oorskry)

Boete = R700,00

Voorbeeld 2:

Spoedbeperking = 60 km/h

Gemiddelde spoed waarteen gery is = 83 km/h

Boete = R1 000,00

(13)

2.1.4 Skryf 'n metode met die naam **voorgesteldeGevaarVlak** wat die gemiddelde reënval op die pas in die vorige sewe dae as 'n parameter ontvang. Die metode stuur 'n voorgestelde gevaarvlak (Laag/Medium/Hoog) terug deur die volgende kriteria te gebruik:

- Indien die gemiddelde reënval gelyk aan of meer as 10 mm is en die maksimum gradiënt van die pas meer as 10 grade is, word voorgestel dat die huidige gevaarvlak na die volgende gevaarvlak verhoog word.

Voorbeeld: Indien die gevaarvlak tans 'Laag' is, sal voorgestel word dat die gevaarvlak na 'Medium' verhoog word.

- As 'n verhoging nie nodig is nie of die huidige gevaarvlak reeds 'Hoog' is, moet die huidige gevaarvlak as die voorgestelde gevaarvlak teruggestuur word.

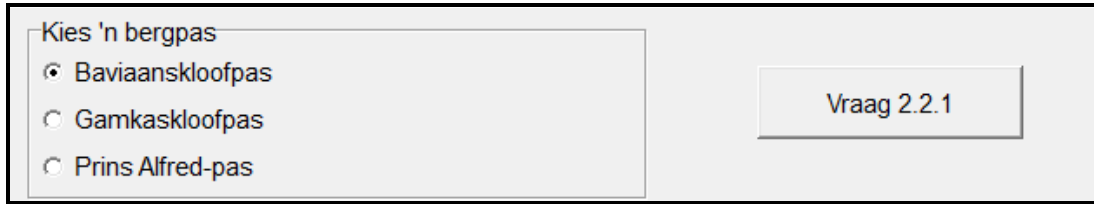
(8)

2.2 Data en GGK ('GUI'):

Die volgende data word verskaf:

- Die padkaart van sommige van die bergpasse is as .jpg-lêers gestoor. Die formaat van die lêernaam van die kaart is **<Naam van die bergpas>.jpg**.
- 'n Tweedimensionele skikking met die naam **arrReen** bevat die reënval wat vir 'n tydperk van sewe dae gemeet is vir die drie bergpasse wat in die GGK ('GUI') gegee word.

Die volgende GGK ('GUI') laat die gebruiker toe om een van drie bergpasse te kies (op die volgende bladsy):



Voltooi die kode vir die knoppies ('buttons') in die hoofvormeenheid (Delphi)/GGK ('GUI')-klas (Java) soos in VRAAG 2.2.1 tot VRAAG 2.2.4 hieronder beskryf word.

2.2.1 Knoppie [Vraag 2.2.1]

Skryf kode om die volgende te doen:

- Bepaal die bergpas gekies uit die radioknoppies ('radio buttons') wat verskaf is.
- Versoek die gebruiker om die afstand, gevaarvlak en gradiënt van die geselekteerde bergpas van die sleutelbord af in te sleutel, deur dialoogblokkies ('dialog boxes') te gebruik.
- Instansieer die **bergpas**-objek. 'n Veranderlike vir die objek is globaal verklaar.
- Vertoon 'n boodskap wat aandui dat die objek geïnstansieer is. (7)

2.2.2 Knoppie [Vraag 2.2.2]


Skryf kode om die volgende te doen:

- Gebruik die toegangsmetodes ('accessor methods') om die inligting oor die bergpas wat deur die gebruiker ingesleutel is, in die relevante teksblokkies te vertoon.
- Toets of die padkaartlêer vir die objek wat geïnstansieer is, bestaan. Indien die kaartlêer wel bestaan, vervang die **<Geen kaart gevind>**-beeld met die inhoud van die kaartlêer in die komponent wat verskaf is. Anders moet die **<Geen kaart gevind>**-beeld in die vertoonarea bly.

Voorbeeld van afvoer en die korrekte beeld indien die gebruiker Baviaanskloofpas gekies het (op die volgende bladsy):

Vraag 2.2.2		Padkaart
Bergpas	<input type="text" value="Baviaanskloofpas"/>	Geen kaart gevind
Afstand	<input type="text" value="110"/>	
Gevaarvlak	<input type="text" value="Medium"/>	
Gradient	<input type="text" value="12"/>	

Voorbeeld van afvoer en die korrekte beeld indien die gebruiker Gamskloofpas gekies het:

Vraag 2.2.2		Padkaart
Bergpas	<input type="text" value="Gamskloofpas"/>	
Afstand	<input type="text" value="75"/>	
Gevaarvlak	<input type="text" value="Laag"/>	
Gradient	<input type="text" value="11"/>	

(8)

2.2.3 Knoppie [Vraag 2.2.3]

Die gebruiker moet die spoedbeperking (in km/h) en die tyd wat gery is (in minute) in die teksblokkies wat verskaf is, insleutel.

Skryf kode om die volgende te doen:

- Verkry die spoedbeperking en tyd wat gery is, uit die teksblokkies. Gebruik hierdie waardes as argumente en roep die **berekenBoete**-metode wat die bedrag wat die reisiger as boete moet betaal, bereken en terugstuur.
- Vertoon die bedrag wat as boete, geformateer as geldeenheid, betaal moet word in die afvoerarea.

Voorbeeld van afvoer om teen die Prins Alfred-pas, met 'n afstand van 88 km en 'n spoedbeperking van 80 km/h, uit te ry. Die tyd gery was 120 minute.

Spoedbeperking	<input type="text" value="80"/>	km/h	Boete	<input type="text" value="R 0.00"/>
Tyd gery	<input type="text" value="120"/>	minute		

Voorbeeld van afvoer om teen die Gamkaskloofpas, met 'n afstand van 75 km en 'n spoedbeperking van 40 km/h, uit te ry. Die tyd gery was 88 minute.

Spoedbeperking	<input type="text" value="40"/>	km/h	Boete	<input type="text" value="R 600.00"/>
Tyd gery	<input type="text" value="88"/>	minute		

(6)

2.2.4 Knoppie [Vraag 2.2.4]

'n Tweedimensionele skikking met die naam **arrReen** bevat die reënval (in millimeter) wat gemeet is op elk van die afgelope 7 dae in elkeen van die bergpasse waaruit die gebruiker kon kies. Die volgorde van die rye in die skikking stem ooreen met die volgorde van die passe in die gegewe radioknoppies op die GGK ('GUI').

Baviaanskloof:	0, 23, 13, 1, 2, 0, 14
Gamkaskloof:	33, 3, 11, 35, 3, 0, 21
Prins Alfred:	50, 0, 0, 1, 20, 0, 2

Skryf kode om die volgende te doen:

- Gebruik die **toString**-metode om inligting oor die **bergpas**-objek in die afvoerarea te vertoon.
- Gebruik die **arrReen**-skikking en bereken die gemiddelde reënval vir die **bergpas**-objek.
- Gebruik die **voorgesteldeGevaarVlak**-metode om die voorgestelde gevaarvlak vir die pas wat op die berekende gemiddelde reënval gebaseer is, te verkry.
- Slegs indien die voorgestelde gevaarvlak van die huidige gevaarvlak verskil, moet 'n dialoogblokkie ('dialog box') gebruik word om die gebruiker te vra of die gevaarvlak na die voorgestelde vlak verander moet word, of nie. Indien die gebruiker kies om die gevaarvlak te verander, moet dit op die voorgestelde vlak gestel word deur die **setGevaarVlak**-metode te gebruik.
- Vertoon 'n boodskap in die afvoerarea wat aandui of:
 - Geen verandering voorgestel is nie
 - Die voorgestelde verandering verwerp is
 - Die gevaarvlak verander is

Gebruik die volgende data om jou program te toets (op die volgende bladsy):

	TOETSDATA 1	TOETSDATA 2	TOETSDATA 3
Bergpas	Baviaanskloof	Gamkaskloof	Prins Alfred
Afstand in km	110	75	88
Gevaarvlak	Medium	Laag	Medium
Gradiënt	12	11	15

LET WEL: Die boodskap moet die gevaarvlak in hoofletters aandui, soos in die voorbeelde hieronder getoon word.

Voorbeeld van afvoer vir die Baviaanskloofpas:

Baviaanskloofpas met 'n maksimum gradient van 12 grade het 'n gevaarvlakgradering van Medium.
Die afstand van die pas is 110 km.

Geen verandering is voorgestel nie. Gevaarvlakgradering bly op MEDIUM.

Voorbeeld van afvoer vir die Gamkaskloofpas indien die voorstel verwerp is:

Gamkaskloofpas met 'n maksimum gradient van 11 grade het 'n gevaarvlakgradering van Laag.
Die afstand van die pas is 75 km.

Voorstel is verwerp. Gevaarvlakgradering word gehou op LAAG.

Voorbeeld van afvoer vir die Prins Alfred-pas indien die voorstel aanvaar is:

Prins Alfred-pas met 'n maksimum gradient van 15 grade het 'n gevaarvlakgradering van Medium.
Die afstand van die pas is 88 km.

Gevaarvlakgradering is verander na HOOG.

(16)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van beide lêers wat jou kode bevat, in.
- Stoor jou program.
- Druk die kode van beide klasse wat jou kode bevat indien drukstukke verlang word.

TOTAAL AFDELING B: 62

AFDELING C**VRAAG 3: PROBLEEMOPLOSSINGPROGRAMMERING****SCENARIO**

Jou pa/voog is besig om 'n vakansie in die Oos-Kaap te beplan waar julle deur verskeie gebiede kan reis. Hy wil die moontlike roetes tussen verskillende dorpe bepaal en of daar bergpasse op hierdie roetes is. Jy moet die nodige sagteware ontwerp om hom hiermee te help.

INSTRUKSIES:

Delphi-programmeerders	Java-programmeerders
<ul style="list-style-type: none"> Die projek Vraag3 is in die DelphiDataAFRMaart2016-lêergids verskaf. Maak die onvoltooide projeklêer Vraag3P.dpr in die Vraag3-lêergids oop. Sleutel jou eksamennommer as kommentaar in die eerste reël van die hoofvormeenheid Vraag3U.pas in. 	<ul style="list-style-type: none"> Die projek Vraag3 is in die JavaDataAFRMaart2016-lêergids verskaf. Maak die onvoltooide klas Vraag3.java wat in die lêergidse Source Packages (src), Vraag3Package voorkom, oop. Sleutel jou eksamennommer as kommentaar in die eerste reël van die klas Vraag3.java in.

GGK ('GUI') wat voorsien is:

Die GGK ('GUI') wat voorsien is, bevat komponente vir toevoer, 'n afvoerarea en drie knoppies.

Roetebeplanner

Dorp waarvandaan vertrek word

Bestemming

3.1 Direkte roetes

3.2 Alle roetes

Sluit af

DATA:

Die gegewe tekslêer **V3Data.txt** bevat 'n onbekende getal teksreëls (wat nie 50 inskrywings oorskry nie).

Elke reël bestaan uit die naam van 'n dorp waarvandaan vertrek word, die naam van 'n buurdorp wat met 'n direkte roete van die dorp waaruit vertrek word, bereik kan word, die afstand tussen die twee dorpe en 'Ja' of 'Nee', wat aandui of daar 'n bergpas op die roete is, of nie. Verskillende karakters word gebruik om die data wat in een reël voorkom, te skei.

Formaat van elke teksreël in die tekslêer:

```
<dorp1>;<dorp2>#<afstand tussen die twee dorpe>*<bergpas>
```

Voorbeeld van data in die tekslêer:

```
Ugie;Maclear#25*Nee  
Elliot;Cala#28*Ja  
Queenstown;Whittlesea#32*Ja  
Whittlesea;Cathcart#46*Nee  
Cathcart;Hogsback#49*Ja  
Elliot;Ugie#51*Ja  
Queenstown;Cathcart#59*Nee  
:
```

Die data in die eerste twee reëls hierbo kan soos volg geïnterpreteer word:

- Die twee buurdorpe is Ugie en Maclear. Die afstand tussen die twee dorpe is 25 km en daar is nie 'n bergpas op die roete nie.
- Die twee buurdorpe is Elliot en Cala. Die afstand tussen die twee dorpe is 28 km en daar is 'n bergpas op die roete.

LET WEL:

- Jy mag NIE die data wat verskaf is, met die hand manipuleer NIE. Kode moet gebruik word om die data wat verskaf is, volgens die vereistes te manipuleer.
- Die gebruik van goeie programmeringstegnieke en modulêre ontwerp moet op die ontwerp en kodering van jou oplossing toegepas word.
- Jy kan geskikte datastrukture gebruik soos wat jy goed dink.

3.1 Direkte roetes:

Die GGK ('GUI') laat toe dat die gebruiker die dorp waarvandaan vertrek word en die dorp wat die bestemming is, in die kombinasie van ('combo boxes') wat verskaf is, te selekteer.

Gebruik die **V3Data.txt**-teks lêer om te bepaal of daar 'n direkte roete bestaan tussen die twee dorpe wat gekies is, met ander woorde of hulle buurdorpe is, of nie.

Indien 'n direkte roete wel bestaan, vertoon die afstand tussen die twee dorpe en 'n nota om aan te dui of daar 'n bergpas op die roete is, of nie.

LET WEL: Direkte roetes word bepaal sonder inagneming van die volgorde waarin die twee buurdorpe se name in die teksreëls voorkom.

Voorbeeld van afvoer indien 'n direkte roete bestaan tussen die twee dorpe wat geselekteer is en daar 'n bergpas op die roete is:

(Data in die lêer: **Elliot;Cala#28*Ja**)

Dorp waarvandaan vertrek word	Bestemming
Cala	Elliot
Die afstand van Cala na Elliot is 28 km. Die roete sluit 'n bergpas in.	

Voorbeeld van afvoer indien 'n direkte roete bestaan tussen die twee dorpe wat geselekteer is en daar nie 'n bergpas op die roete is nie:

(Data in die lêer: **Ugie;Maclear#25*Nee**)

Dorp waarvandaan vertrek word	Bestemming
Ugie	Maclear
Die afstand van Ugie na Maclear is 25 km. Die roete sluit nie 'n bergpas in nie.	

Voorbeeld van afvoer indien daar nie 'n direkte roete bestaan tussen die twee dorpe wat geselekteer is nie:

Dorp waarvandaan vertrek word	Bestemming
Queenstown	Maclear
Geen moontlike roete is tussen Queenstown en Maclear gevind nie.	

(18)

3.2 Lys van buurdorpe:

Die gebruiker moet slegs 'n dorp waarvandaan vertrek word uit die gegewe kombinasie lys ('combo box') selekteer. Vertoon al die buurdorp(e) wat direk bereik kan word vanaf die geselekteerde dorp waarvandaan vertrek word. Vertoon ook die afstand(e) tussen die dorpe.

Gebruik die data wat in die **V3Data.txt**-teks lêer verskaf is om die vereiste dorpe en afstande te verkry.

Vertoon die inligting in kolomformaat, in stygende volgorde volgens die afstande tussen die dorpe gesorteer. 'n Geskikte opskrif moet vertoon word om die dorp waarvandaan vertrek word, aan te dui.

Voorbeeld van afvoer indien Whittlesea as die dorp waarvandaan vertrek word, geselekteer word:

Dorpe wat direk van Whittlesea af bereik kan word:	
Queenstown	32 km
Cathcart	46 km

(22)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die program lêer in.
- Stoor jou program.
- 'n Drukstuk van die kode kan vereis word.

TOTAAL AFDELING C: 40
GROOTTOTAAL: 150



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

**NASIONALE
SENIOR SERTIFIKAAT**

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

FEBRUARIE/MAART 2015

MEMORANDUM

PUNTE: 150

Hierdie memorandum bestaan 32 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as basis tydens die nasiensessie gebruik word. Dit is geskep om deur die nasieners gebruik te word. Daar word van alle nasieners verwag om 'n standaardiseringsvergadering by te woon om te verseker dat die riglyne konsekwent geïnterpreteer en toegepas word tydens die nasien van kandidate se werk.
- Let daarop dat kandidate wat alternatiewe korrekte oplossings voorsien as dié wat as voorbeelde van oplossings in hierdie nasienriglyne gegee is, volle krediet vir die betrokke antwoord sal ontvang tensy die instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom word nie.
- **Bylae A, B en C** (bladsy 3–7) bevat die nasienrubriek vir elke vraag ongeag watter een van die twee programmeringstale gebruik is.
- **Bylae D, E en F** (bladsy 8–17) bevat voorbeelde van oplossings in programmeringskode vir Java vir Vraag 1 tot 3.
- **Bylae G, H en I** (bladsy 18–32) bevat voorbeelde van oplossings in programmeringskode vir Delphi vir Vraag 1 tot 3.
- Kopieë van **Bylae A, B en C** (bladsy 3–7) moet vir elke kandidaat gemaak word en tydens die nasiensessie voltooi word.

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

SENTRUMNUMMER:		EKSAMENNUMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER SE PUNT
1.1	[Knoppie – Genereer advertensie] Onttrek die markprys ✓ Onttrek die verkoopprys ✓ Onttrek die getal slaapkamers ✓ Onttrek die getal badkamers ✓ Gebruik die inligting wat onttrek is om die advertensie-reël saam te stel ✓ Kontroleer of swembad geselekteer is ✓ Voeg Swembad# by advertensie ✓ Kontroleer of verkoopprys < markprys ✓ Voeg Winkoop by advertensie ✓ Vertoon opskrif in die afvoer-area ✓ Vertoon advertensie in die afvoer-area ✓	11	
1.2.1	Radioknoppie Verfwerk: Stoor geselekteerde opknappingstipe as V ✓	1	
1.2.2	Radioknoppie Teelwerk: Stoor geselekteerde opknappingstipe as T ✓	1	
1.2.3	[Knoppie – Bereken en vertoon opkappingskoste] (a) Inisialiseer alle tel-veranderlikes na 0 ✓ Skep en inisialiseer koste-veranderlike ✓ Onttrek die oppervlakte uit die teksblokkie en skakel om na desimale waarde ✓ Kontroleer of die tipe opknapping verfwerk is ✓ Bereken die volume verf benodig in liter. Formule: $\text{getalLiter} = (\text{oppervlakte} / 8)$ ✓ Rond getalLiter op ✓ Bereken die hoeveelheid van elke dromgrootte : As die getal liter meer is as 5 ✓ Bereken hoeveelheid 5 liter-dromme ✓ Bereken koste van die getal 5 liter-dromme ✓ Bereken die oorblywende verf benodig ✓ Herhaal vir: 2 liter-dromme ✓ Ken die oorblywende verf toe aan een liter-dromme ✓ Bereken koste ✓ Vertoon oppervlakte en volume verf in liter ✓ Vertoon elke drom-grootte ✓ en getal dromme benodig ✓ Vertoon met twee desimale plekke ✓	17	

(b)	Kontroleer of die tipe opknapping teelwerk is ✓ Gebruik dialoogblokkie om die koste per vierkante meter in te sleutel ✓ Bereken die koste van teelwerk deur van die onderstaande formule gebruik te maak: $\text{KosteVanTeelwerk} = (\text{oppervlakte} + 5) \times \text{kostePerVierkanteMeter}$ ✓* Vertoon opskrif ✓ Vertoon koste van teelwerk ✓	6	
1.3.1	[Knoppie – Bereken bedrag verskuldig] Onttrek vorige lesing uit teksblokkie as heelgetal-waarde ✓ Onttrek huidige bedrag uit teksblokkie as heelgetal-waarde ✓ As huidige lesing < vorige lesing ✓ Stel huidige lesing na 0 ✓ en vertoon boodskap ✓ anders $\text{elekVerbruik} = (\text{huidigeLesing} - \text{vorigeLesing})$ ✓ as $\text{elekVerbruik} \leq 600$ ✓ $\text{elekRekening} = \text{elekVerbruik}$ ✓ anders $\text{elekRekening} = 600 + (\text{elekVerbruik} - 600) \times 1.5$ ✓ Vertoon elektrisiteitsrekening in die byskrif-komponent wat voorsien is Formaat: geldeenheid en twee desimale plekke ✓	11	
1.3.2	[Knoppie – Lys Geisers] Onttrek die grootte van die geiser uit die teksblokkie ✓ Gebruik lus om deur skikking te stap ✓ As die inskrywing in die skikking die ingesleutelde grootte van geiser bevat ✓ Begin die if-blok met kode ✓ Onttrek die naam van die geiser uit die inskrywing in die skikking ✓ Vertoon die naam van die geiser in die teks-area ✓	6	
	TOTAAL:	23	

BYLAE B:**AFDELING B:****VRAAG 2: NASIENRUBRIEK – OBJEK-GEORIËNTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER SE PUNT
2.1.1	Vyf attribute is verklaar✓ as privaat ✓ String-datatype: aansoeker naam } Heelgetal-datatype: jare } ✓ Reële datatype: besteebare inkomste, lening-bedrag, rentekoers ✓	4	
2.1.2	Konstruktor: Opskrif ✓ met drie parameters✓ Parameters korrekte data tipes ✓ Ken parameterwaardes aan drie korrekte attribute toe ✓ Verstekwaardes toegeken aan jare en rentekoers ✓	5	
2.1.3	Toegangsmetode ('Accessor') vir naam van aansoeker: Metode se opskrif ✓ return-stelling ✓	2	
2.1.4	Wysigingsmetode ('Mutator') vir jare en rentekoers: setJare: parameter is ontvang ✓ en toegeken aan regte attribuut✓ setRentekoers: is ontvang ✓ en toegeken aan regte attribuut✓	4	
2.1.5	berekenPaaielement-metode: Haal kommentaar van kode af ✓	1	
2.1.6	isGoedgekeur-metode: Skep boolese veranderlike vir goedgekeur wat op true ✓ gestel is As leningBedrag > 800000 en ✓ jare < 25✓ verander goedgekeur na false ✓ As leningBedrag <= 600000 en ✓ besteebareInkomste < ✓ 130% van paaielement ✓ verander goedgekeur na false ✓ Stuur goedgekeur terug ✓	9	
2.1.7	toString-metode: Uitleg van string-veranderlike: Elke deel van die string op 'n nuwe reël ✓, korrekte byskrifte ✓ Korrekte inligting: Aansoeker se naam, besteebare inkomste ✓, lening-bedrag, jare ✓, rentekoers met %-teken ✓	5	

2.2.1	<p>[Knoppie – Vind aansoeker]</p> <p>Onttrek die aansoeker se naam uit die kombinasie lys ✓ As 'n naam nie geselekteer is nie ✓ vertoon geskikte boodskap ✓ As 'n naam geselekteer is ✓ vind die tekslêer wat dieselfde naam as die aansoeker het ✓ As die lêer bestaan ✓ maak die tekslêer oop ✓ en lees die volgende uit tekslêer: inkomste, uitgawe ✓ lening-bedrag ✓ Bereken die besteebare inkomste ✓ Instansieer (skep) 'n nuwe lening-objek ✓ deur gebruik te maak van die korrekte parameters ✓✓ Vertoon geskikte boodskap deur gebruik te maak van 'n dialoogblokkie ✓ Maak die "Evalueer aansoek"-knoppie aktief ✓ As die lêer nie bestaan nie, ✓ Vertoon 'n geskikte boodskap ✓ Maak gebruik van 'n korrekte validasietegniek om uit 'n tekslêer te lees. ✓</p>	18	
2.2.2	<p>[Knoppie – Evalueer aansoek]</p> <p>Onttrek die rentekoers uit die teksblokkie ✓ en skakel om na reële getal ✓ Lees jare uit die kombinasie lys ✓ en verander na heelgetal ✓ Stel die rentekoers van die objek deur van die 'mutator'-metode gebruik te maak ✓ Stel die jare van die objek deur van die 'mutator'-metode gebruik te maak ✓ Kontroleer of die lening goedgekeur is deur die relevante metode te roep ✓ As goedgekeur ✓ vertoon die inligting van die huislening-objek deur van die toString-metode gebruik te maak ✓ vertoon byskrif met paaiement ✓ anders ✓ vertoon die naam van die aansoeker deur gebruik te maak van die 'accessor'-metode en vertoon die boodskap 'NIE GOEDGEKEUR NIE' ✓</p>	12	
	TOTAAL:	60	

BYLAE C:**AFDELING C:****VRAAG 3: NASIENRUBRIEK – PROBLEEMOPLOSSINGPROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER SE PUNT
3.1	<p>[Knoppie – Search for agent's name] Skep veranderlikes wat benodig word ✓ Gebruik lus om deur om deur skikking met die naam arrAgente ✓ Toets of kode in die skikkingsinskrywing ✓ die kode wat ingetik is, bevat ✓ Onttrek agent se naam uit die skikkingsinskrywing ✓ Vertoon die naam op die byskrif wat voorsien is ✓ Staak die soek-proses ✓ Inkrementeer teller ✓ As die agent se kode nie gevind is nie ✓ Vertoon die boodskap 'Ongeldige agentkode' in die byskrif ✓ Maak die teksblokkie vir agent se kode skoon ✓</p>	11	
3.2.1	Voeg 'n geskikte komponent soos 'n tabel/matriks/afvoer-area op die GGK (GUI) ✓ Herbenoem die komponent ✓	2	
3.2.2	<p>[Knoppie – Agent se verkope] Skep 'n 2d-skikking of enige ander struktuur/strukture vir verwerking ✓✓ Inisialiseer die 2d-skikking/ander struktuur/strukture ✓✓ Skep 'n veranderlike vir totale verkope ✓ Gebruik lus om deur die skikking met die naam arrVerkope te stap ✓ As agent se kode = kode in skikkingsinskrywing ✓ Onttrek maand uit skikkingsinskrywing ✓ Stel kolom se waarde deur maand te gebruik ✓ Onttrek tipe eiendom uit skikkingsinskrywing ✓ Stel ry se waarde deur tipe eiendom te gebruik ✓✓✓ Vermeerder die waarde van 2dArr[ry][kol] met 1 ✓ Vermeerder die waarde van 2dArr[ry][4] met 1 vir ry met totale ✓ Vermeerder die totale verkope veranderlike met die prys in skikkingsinskrywing ✓✓ Vertoon die inhoud van die 2d-skikking (of ander struktuur/e) in geskikte komponent, geformateer en gespaseer. Vertoon die volgende inligting: Maande as opskrifte ✓ Tipe eiendomme as byskrifte ✓ Ry- en kolomwaardes ✓✓✓ Totale van rye ✓ Vertoon die totale waarde van verkope in 'n geskikte komponent ✓</p>	24	
TOTAAL:		37	

OPSOMMING VAN LEERDER SE PUNTE:

	AFDELING A	AFDELING B	AFDELING C	
	VRAAG 1	VRAAG 2	VRAAG 3	GROOTTOTAAL
MAKS. PUNTE	53	60	37	150
LEERDER SE PUNTE				

BYLAE D: OPLOSSING VIR VRAAG 1: JAVA

```
//      'n Moontlike oplossing vir Vraag 1

//=====
//Hierdie kode is in die program gegee
//=====

    public Vraag1_Memo() {

        initComponents();
        this.setLocationRelativeTo(this);
    }

//=====
// Nuwe kode
//=====

DecimalFormat df = new DecimalFormat("R 0.00");

    char opknappingsTipe = ' ';
    double eletrRekening;

//=====
//  Vraag 1.1
//=====
    private void btnVerkoopsAdvActionPerformed(java.awt.event.ActionEvent
    evt) {
        double mark = Double.parseDouble(txfMarkwaarde.getText());
        double verkoop = Double.parseDouble(txfVerkoopsprys.getText());
        String bed = txfBeddens.getText();
        String badkamer = txfBadkamers.getText();
        String advertensie = "R" + (int)(verkoop) + "#" + bed + "Bed#"
            + badkamer + "Bad#";

        if (chkSwembad.isSelected()) {
            advertensie += "Swembad#";
        }
        if (verkoop < mark) {
            advertensie += "Winskoop";
        }
        txafvoer.setText("Huis te koop:\n");
        txafvoer.append(advertensie);
    }

//=====
//  Vraag 1.2.1
//=====
    private void rbtVerfActionPerformed(java.awt.event.ActionEvent evt) {
        opknappingsTipe = 'V';
    }

//=====
//  VRAAG 1.2.2
//=====
    private void rbtTeelActionPerformed(java.awt.event.ActionEvent evt) {
        opknappingsTipe = 'T';
    }

```

```

=====
//  Vraag 1.2.3 (a)
=====
private void btnBerekOpknappingActionPerformed(java.awt.event.ActionEvent
evt) {
    int aantEenLiterDrom = 0;
    int aantTweeLiterDrom = 0;
    int aantVyfLiterDrom = 0;
    double koste = 0.0;
    double oppervlakte = Double.parseDouble(txfOppervlakte.getText());
    if (opknappingsTipe == 'V') {
        double aantLiterVerf = oppervlakte / 8;
        int volumeLiter = (int) (Math.round(aantLiterVerf + 0.4));
        if (volumeLiter / 5 > 0) {
            aantVyfLiterDrom += (int) (volumeLiter / 5);
            volumeLiter -= aantVyfLiterDrom * 5;
            koste += aantVyfLiterDrom * 199;
        }
        if (volumeLiter / 2 > 0) {
            aantTweeLiterDrom += (int) (volumeLiter / 2);
            volumeLiter -= aantTweeLiterDrom * 2;
            koste += aantTweeLiterDrom * 92.30; }
        aantEenLiterDrom = volumeLiter;
        koste += aantEenLiterDrom * 55.50;
        txafvoerOpknapping.setText("Oppervlakte: " +
            (int)oppervlakte + " vierkante meter");
        txafvoerOpknapping.append("\nVolume verf wat benodig word: " +
            aantLiterVerf + " liter");
        txafvoerOpknapping.append("\n\n1 liter-dromme: " +
            aantEenLiterDrom);
        txafvoerOpknapping.append("\n2 liter-dromme: " +
            aantTweeLiterDrom);
        txafvoerOpknapping.append("\n5 liter-dromme: " +
            aantVyfLiterDrom);
        txafvoerOpknapping.append("\n\nTotale koste: " +
            df.format(koste)); }
    if (opknappingsTipe == 'T') {
        double teelKoste =
            Double.parseDouble(JOptionPane.showInputDialog("Sleutel die
            koste per vierkante meter in"));
        koste = (oppervlakte + 5) * teelKoste;
        txafvoerOpknapping.setText("Oppervlakte: " +
            Math.round(oppervlakte) + " vierkante meter" + "\nTotale
            koste: " + df.format(koste));}
=====
//  Vraag 1.2.3 (b)
=====
if (opknappingsTipe == 'T') {
    double teelKoste =
        Double.parseDouble(JOptionPane.showInputDialog("Sleutel die koste per
        vierkante meter in"));
    koste = (oppervlakte + 5) * teelKoste;
    txafvoerOpknapping.setText("Oppervlakte: " +
        Math.round(oppervlakte) + " vierkante meter" + "\nTotale koste: " +
        df.format(koste));
} }

```

```
//=====
//  Vraag 1.3.1
//=====
private void btnBerekenBedragActionPerformed(java.awt.event.ActionEvent
evt) {
    int vorigeLesing = 0;
    int huidigeLesing = 0;
    double eletrRekening = 0;
    vorigeLesing = Integer.parseInt(txfVorige.getText());
    huidigeLesing = Integer.parseInt(txfHuidige.getText());
    if (vorigeLesing > huidigeLesing) {
        JOptionPane.showMessageDialog(null, "Vorige lesing is groter as
huidige lesing, sleutel weer in");
        txfHuidige.setText("0");
    } else {
        int elektrGebruik = huidigeLesing - vorigeLesing;
        if (elektrGebruik <= 600) {
            eletrRekening = elektrGebruik;
        } else {
            eletrRekening = 600 + ((elektrGebruik - 600) * 1.5);
        }
    }
    lblElektrisiteit.setText(df.format(eletrRekening));
}

//=====
//  Vraag 1.3.2
//=====

private void btnVindGeisersActionPerformed(java.awt.event.ActionEvent evt)
{
    String geiserGrootte = txfGeiserGrootte.getText().trim();
    txafvoerGeisers.setText("");
    for (int i = 0; i < arrGeisers.length; i++) {
        int psnDash = arrGeisers[i].indexOf("-");
        String grootte = arrGeisers[i].substring(0, psnDash);
        if (grootte.equals(geiserGrootte)) {
            txafvoerGeisers.append(arrGeisers[i].substring(psnDash +
1) + "\n");
        }
    }
}
}
```

BYLAE E: OPLOSSING VIR VRAAG 2: JAVA**Objekklas: Huislening.java**

```
//      'n Oplossing vir Vraag 2
//=====
//  Vraag 2.1.1
//=====
    private String aansoekerNaam;
    private double besteebareInkomste;
    private double leningBedrag;
    private int jare;
    private double renteKoers;
//=====
//  Vraag 2.1.2
//=====
    public HuisLening(String aansoekerNaam, double besteebareInkomste, double
leningBedrag) {
        this.aansoekerNaam = aansoekerNaam;
        this.besteebareInkomste = besteebareInkomste;
        this.leningBedrag = leningBedrag;
        this.jare = 0;
        this.renteKoers = 0;

}
//=====
//  VRAAG 2.1.3
//=====
    public String getAansoekerNaam() {
        return aansoekerNaam;
    }
//=====
//  Vraag 2.1.4
//=====
    public void setJare(int jare) {
        this.jare = jare;
    }
    public void setRenteKoers(double renteKoers) {
        this.renteKoers = renteKoers;
    }
//=====
//  Vraag 2.1.5 - Kode wat gegee is (verwyder kommentaartekens)
//=====
    public double berekenPaaieiment() {
        double paaieiment, koers;
        int result;
        koers = renteKoers / 100.0;
        paaieiment = (koers * leningBedrag) / (1 - ((Math.pow((1 + koers),
(jare * -1)))));
        result = (int) (Math.round(paaieiment / 12));
        return result;
    }
}
```

```
//=====
//  Vraag 2.1.6
//=====
public boolean isGoedgekeur() {
    boolean goedgekeur = true;
    if (leningBedrag > 800000 && jare < 25) {
        goedgekeur = false;
    }
    if (leningBedrag <= 600000) {
        if (besteembareInkomste < (berekenPaalement() * 1.3)) {
            goedgekeur = false;
        }
    }
    return goedgekeur;
}

//=====
//  Vraag 2.1.7
//=====
public String toString() {
    String afvoer = "Naam van aansoeker: "+aansoekerNaam+
        "\nBesteembare inkomste: " + df.format(besteembareInkomste) +
        "\nLeningbedrag: " + df.format(leningBedrag) + "\nGetal jare: " +
        jare + "\nRentekoers: " + renteKoers + "%";
    return afvoer;
}
}
```

```
//=====
```

GGK-klas ('GUI class'): Vraag2.java

```
//=====
//  Kode wat gegee is
//=====
HuisLening leningObj;
//=====
//  Vraag 2.2.1
//=====
private void btnVindAansoekerActionPerformed(java.awt.event.ActionEvent
evt) {
    String aansoeker = (String) (cmbAansoekers.getSelectedItem());
    if (aansoeker.isEmpty()) {
        JOptionPane.showMessageDialog(rootPane, "Kies 'n aansoeker",
            "Warning", 0);
        txaAfvoer.setText("");
    }
    else {
        try {
            Scanner sc = new Scanner(new FileReader(aansoeker + ".txt"));
            txaAfvoer.setText("\n Huisleningaansoek is ingedien deur " +
                aansoeker);

            String naam = sc.nextLine();
            double inkomste = Double.parseDouble(sc.nextLine());
            double uitgawes = Double.parseDouble(sc.nextLine());
            double leningBedrag = Double.parseDouble(sc.nextLine());
            double besteembareInkomste = inkomste - uitgawes;
            leningObj = new HuisLening(aansoeker, besteembareInkomste,
                leningBedrag);
            JOptionPane.showMessageDialog(null, "Huisleningobjek suksesvol
```

```

        geskep");
    btnEvalueer.setEnabled(true);
} catch (FileNotFoundException e) {
    txaAfvoer.setText("Geen huislening aansoek is deur " + aansoeker
        + " ingedien nie.");
}
}
}

//=====
//  Vraag 2.2.2
//=====

private void btnEvalueerActionPerformed(java.awt.event.ActionEvent evt) {
    int jare = Integer.parseInt((String) (cmbJare.getSelectedItem()));
    double renteKoers = Double.parseDouble(txfRenteKoers.getText());
    leningObj.setRenteKoers(renteKoers);
    leningObj.setJare(jare);
    if (leningObj.isGoedgekeur()) {
        txaAfvoer.setText(leningObj.toString() + "\n\nLening GOEDGEKEUR
            met 'n maandlikse paaieiment van " +
            df.format(leningObj.berekenPaaieiment()));
    }
    else {
        txaAfvoer.setText("Naam van aansoeker: " +
            leningObj.getAansoekerNaam() + "\n");
        txaAfvoer.append("Huislening NIE GOEDGEKEUR NIE");
    }
}
}
```

BYLAE F: OPLOSSING VIR VRAAG 3: JAVA

```
//      'n Moontlike oplossing vir Vraag 3

public class Vraag3_Memo extends javax.swing.JFrame
{
//=====
//Hierdie kode is gegee in die program
//=====

// Klas wat gegee is

public String [] populateArrVerkope(){
    String[] arrVerkope = {"01#K#A122;500000", "01#K#A120;1200000",
        "01#K#A123;850000", "01#R#A121;75000", "02#K#A123;6200000",
"02#L#A121;190000", "02#K#A121;500000", "02#K#A120;1200000",
        "02#K#A120;850000", "02#R#A124;75000", "02#K#A123;620000",
"02#L#A124;690000", "02#K#A121;500000", "02#K#A120;1200000",
        "02#K#A120;850000", "02#R#A124;75000", "02#K#A123;620000",
"02#L#A124;790000", "02#R#A124;1500000", "02#K#A125;200000",
        "02#L#A121;400000", "03#R#A120;175000", "03#K#A123;6200000",
"03#L#A123;190000", "03#K#A124;500000", "03#K#A125;1200000",
        "03#K#A123;850000", "03#R#A122;75000", "03#K#A123;620000",
"03#L#A123;890000", "03#R#A123;1500000", "03#K#A125;200000",
        "03#L#A123;400000", "04#R#A124;175000", "04#K#A120;6200000",
"04#L#A123;190000", "03#R#A124;1500000", "03#K#A125;200000",
        "03#L#A122;400000", "04#R#A121;175000", "04#K#A123;6200000",
"04#L#A123;190000", "04#K#A121;500000", "04#K#A120;1200000",
        "04#K#A120;850000", "04#R#A124;75000", "04#K#A123;620000",
"04#L#A123;590000", "04#K#A123;500000", "04#K#A125;120000",
        "04#K#A124;850000", "04#R#A122;75000", "04#K#A123;6200000",
"04#L#A120;490000", "04#K#A125;500000", "04#K#A125;120000",
        "04#K#A124;850000", "04#R#A122;75000", "04#K#A123;6200000",
"04#L#A120;390000", "04#R#A120;1500000", "04#K#A121;200000",
        "04#L#A122;400000", "04#R#A124;175000", "04#K#A121;6200000",
"04#L#A123;190000", "11#K#A122;500000", "07#K#A120;1200000",
        "10#K#A123;850000", "12#R#A121;75000", "12#K#A123;6200000",
"11#L#A121;190000", "12#K#A121;500000", "12#K#A120;1200000",
        "11#K#A120;850000", "12#R#A124;75000", "12#K#A123;620000",
"11#L#A124;690000", "09#K#A121;500000", "08#K#A120;1200000",
        "11#K#A120;850000", "12#R#A124;75000", "12#K#A123;620000",
"10#L#A124;790000", "08#R#A124;1500000", "08#K#A125;200000",
        "11#L#A121;400000", "11#R#A120;175000", "07#K#A123;6200000",
"09#L#A123;190000", "07#K#A124;500000", "07#K#A125;1200000",
        "12#K#A123;850000", "10#R#A122;75000", "08#K#A123;620000",
"08#L#A123;890000", "06#R#A123;1500000", "09#K#A125;200000",
        "09#L#A123;400000", "09#R#A124;175000", "08#K#A120;6200000",
"07#L#A123;190000", "08#R#A124;1500000", "06#K#A125;200000",
        "07#L#A122;400000", "08#R#A121;175000", "08#K#A123;6200000",
"07#L#A123;190000", "11#K#A121;500000", "06#K#A120;1200000",
        "08#K#A120;850000", "08#R#A124;75000",
"12#K#A123;620000", "07#L#A123;590000",
        "12#K#A123;500000", "07#K#A125;120000",
"08#K#A124;850000", "07#R#A122;75000",
        "11#K#A123;6200000", "09#L#A120;490000",
"08#K#A125;500000", "07#K#A125;120000",
        "09#K#A124;850000", "06#R#A122;75000",
"10#K#A123;6200000", "09#L#A120;390000",
        "10#R#A120;1500000", "07#K#A121;200000",
"09#L#A122;400000", "06#R#A124;175000",
        "10#K#A121;6200000", "09#L#A123;190000"};
    return arrVerkope;
}
}
```

```
public String [] populateArrAgente(){  
String[] arrAgente = {"A120:Wes Seelig",  
    "A121:Tandra Peters",  
    "A122:Lisha Champ",  
    "A123:Melda Bissonette",  
    "A124:Leta Hansell",  
    "A125:Sid Rebello"};  
return arrAgente;  
}  
}
```

Vraag 3 – GGK-klas ('GUI class')

```

public class VRAAG3_Memo extends javax.swing.JFrame {

//=====
//Given code
//=====

PopulateArrays objAssign = new PopulateArrays();

    String [] arrSales = objAssign.populateArrSales();
    String [] arrAgents = objAssign.populateArrAgents();

public Q3() {
    initComponents();
    this.setLocationRelativeTo(this);
}

//=====
// Nuwe kode
//=====
    String agentKode = "";
    String agentNaam = "";
    String[] arrTipes = {"Kommersieel", "Resedensieel", "Landbou"};
    String[] arrMaande = {"Jan", "Feb", "Mrt", "Apr", "Mei", "Jun", "Jul", "Aug",
"Sep", "Okt", "Nov", "Des"};

    int [][] arrStats = new int [3][5];

    public void clearSales() {
        for (int r = 0; r < 3; r++) {
            for (int c = 0; c < 5; c++) {
                arrStats[r][c] = 0;
            }
        }
    }

//=====
// VRAAG 3.1
//=====
private void btnAgentNaamActionPerformed(java.awt.event.ActionEvent evt) {
    txv3.setText("");
    agentKode = txfAgentKode.getText();
    boolean gevind = false;
    int p = 0;
    while (!gevind && p < arrAgente.length) {
        if (arrAgente[p].substring(0, 4).equals(agentKode)) {
            agentNaam =
                arrAgente[p].substring(arrAgente[p].indexOf(":") + 1);
            lblAgentNaam.setText(agentNaam);
            gevind = true;
        }
        p++;
    }
    if (!gevind) {
        lblAgentNaam.setText("Ongeldige agentkode");
        txfAgentKode.setText("");
    }
}
}

```

```

=====
// VRAAG 3.2.1
=====
Plaas 'n geskikte afvoercomponent op die GGK (GUI)
Hierdie oplossing maak gebruik van 'n teksarea-komponent
'n Grid kan ook gebruik word met 'n label-komponent om totale verkope te
vertoon.
=====
// VRAAG 3.2.2
=====
private void btnAgentVerkopeActionPerformed(java.awt.event.ActionEvent evt)
{
    btnAgentNaam.doClick();
    initVerkope();
    int totaal = 0;
    for (int i = 0; i < arrVerkope.length; i++) {
        if (arrVerkope[i].substring(5, 9).equals(agentKode)) {
            int maand = Integer.parseInt(arrVerkope[i].substring(0, 2));
            int kol = maand - 1;
            char temp = arrVerkope[i].charAt(3);
            int ry = 0;
            switch (temp) {
                case 'K':
                    ry = 0;
                    break;
                case 'R':
                    ry = 1;
                    break;
                case 'L':
                    ry = 2;
                    break;
            }
            arrAgentVerkope[ry][kol]++;
            arrAgentVerkope[ry][12] += 1;

            totaal += Integer.parseInt(arrVerkope[i].substring
                (arrVerkope[i].indexOf(";") + 1));
        }
    }

    //Vertoon in teksarea
    txaV3.setText(String.format("%-13s", ""));
    for (int i = 0; i < 12; i++) {
        txaV3.append(String.format("%-4s", arrMaande[i]));
    }
    txaV3.append("\n\n");
    for (int r = 0; r < 3; r++) {
        txaV3.append(String.format("%-13s", arrTipes[r]));

        for (int c = 0; c < 13; c++) {
            txaV3.append(String.format("%-4s",
                arrAgentVerkope[r][c]));
        }
        txaV3.append("\n\n");
    }
    txaV3.append("Totale waarde van verkope: R " + totaal);
}

```

BYLAE G: OPLOSSING VIR VRAAG 1: DELPHI

```
unit Vraag1_UMemo;

interface

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

type
  TfrmVraag1 = class(TForm)
    pgcVraag1: TPageControl;
    tsQ1_1: TTabSheet;
    pnlButtons: TPanel;
    bmbClose: TBitBtn;
    tsQ1_3: TTabSheet;
    tsQ1_2: TTabSheet;
    lblMarkWaarde: TLabel;
    lblVerkoopprys: TLabel;
    lblSlaapkamers: TLabel;
    lblBadkamers: TLabel;
    chkSwembad: TCheckBox;
    btnVerkoopsAdvertensie: TButton;
    edtMarkwaarde: TEdit;
    edtVerkoopprys: TEdit;
    edtSlaapkamers: TEdit;
    edtBadkamers: TEdit;
    lblOpskrifKiloW: TLabel;
    lblVorigeLesing: TLabel;
    lblHuidigeLesing: TLabel;
    lblElektrisiteit: TLabel;
    lblGeiserGroottes: TLabel;
    edtVorige: TEdit;
    edtHuidig: TEdit;
    edtGrootte: TEdit;
    lblOpknappingArea: TLabel;
    lblVMeter: TLabel;
    redOpknapping: TRichEdit;
    btnBerekenOpknapping: TButton;
    btnVindGeisers: TButton;
    grpOpknappingsArea: TGroupBox;
    grpElektrisiteitGebruik: TGroupBox;
    btnBerekenBedrag: TButton;
    lblBedragVerskuldig: TLabel;
    grpGeiserOpsies: TGroupBox;
    grpAdvertensie: TGroupBox;
    redV1: TRichEdit;
    rgbVerf: TRadioButton;
    rgbTeel: TRadioButton;
    memGeisers: TMemo;
    grpOpknappingsMetode: TGroupBox;
    edtArea: TEdit;
    procedure btnVerkoopsAdvertensieClick(Sender: TObject);
    procedure btnBerekenOpknappingClick(Sender: TObject);
```

```

procedure btnBerekenBedragClick(Sender: TObject);
procedure rgbVerfClick(Sender: TObject);
procedure rgbTeelClick(Sender: TObject);
procedure btnVindGeisersClick(Sender: TObject);

private
  { Private declarations }
public
  { Public declarations }
end;

var
  frmVraag1: TfrmVraag1;
  cTipeOpknop : char;
  arrGeisers : array [1..20] of string = ('50-QuickSun50','100-
QuickSun100','150-QuickSun150','50-Solar Magic','50-InHotWater',
                                           '100-SunnyBath 100','150-
SunnyBath 150','50-WaterJoy 50','100-WaterJoy 100',
                                           '150-BigTub 150','50-Small
Wonder','100-Medium Wonder','150-Large Wonder',
                                           '100-SolarWarmth 100','150-
SolarWarmth 150','50-Sun Magic','50-Eco Wonder 50',
                                           '100-Eco Wonder 100','150-Eco
Wonder 150','150-Big Earth Saver');

```

Implementation

```

{$R *.dfm}
{$R+}
=====
// Vraag 1.1
=====
procedure TfrmVraag1.btnVerkoopsAdvertensieClick(Sender: TObject);
var
  rMarkWaarde, rVerkoop : real;
  sAdvertensie          : string;
  iBed, iBad            : integer;
begin
  rMarkWaarde := StrToFloat(edtMarkwaarde.Text);
  rVerkoop    := StrToFloat(edtVerkoopprys.Text);
  iBed        := StrToInt(edtSlaapkamers.Text);
  iBad        := StrToInt(edtBadkamers.Text);
  sAdvertensie := 'R' + FloatToStr(rVerkoop) + '#' +
                  IntToStr(iBed) + 'Bed#' + IntToStr(iBad) + 'Bad#';
  if chkSwembad.Checked then
    sAdvertensie := sAdvertensie + 'Swembad#';

  if rVerkoop < rMarkWaarde then
    sAdvertensie := sAdvertensie + 'Winskoop';

  redV1.Lines.Clear;
  redV1.Lines.Add('Huis te koop:' + #13 + sAdvertensie);
end;

```

```
=====
// Vraag 1.2.1
=====
procedure TfrmVraag1.rgbVerfClick(Sender: TObject);
begin
    cTipeOpknop := 'V';
end;
=====
// Vraag 1.2.2
=====
procedure TfrmVraag1.rgbTeelClick(Sender: TObject);
begin
    cTipeOpknop := 'T';
end;
=====
// Vraag 1.2.3
=====
procedure TfrmVraag1.btnBerekenOpknappingClick(Sender: TObject);
var
    rKoste, rLiterVerf, rDromkoste, rTeelKoste : real;
    iOppervlakte, iVolLiter, iEenLiter, iTweeLiter, iVyfLiter : integer;
begin
    // Vraag 1.2.3(a)
    iOppervlakte := StrToInt(edtArea.Text);
    iEenLiter := 0;
    iTweeLiter := 0;
    iVyfLiter := 0;
    rKoste := 0;
    iVolLiter := 0;
    rLiterVerf := 0;
    rDromkoste := 0;
    rTeelKoste := 0;

    if cTipeOpknop = 'V' then
        begin
            rLiterVerf := iOppervlakte / 8;
            iVolLiter := Round(rLiterVerf + 0.4);
            // Alternatief >> Afronding na naaste heelgetal
            // iVolLiter := Round(rLiterVerf);
            // if (iArea mod 8 > 0)
            // then iVolLiter := iVolLiter + 1;

            if (iVolLiter div 5) > 0 then
                begin
                    iVyfLiter := iVolLiter div 5;
                    dec(iVolLiter, (iVyfLiter * 5));
                    rKoste := rKoste + (iVyfLiter * 199.00);
                end;
            if (iVolLiter div 2) > 0 then
                begin
                    iTweeLiter := iVolLiter div 2;
                    dec(iVolLiter, (iTweeLiter * 2));
                    rKoste := rKoste + (iTweeLiter * 92.30);
                end;

            iEenLiter := iVolLiter;
            rKoste := rKoste + (iEenLiter * 55.50);
            redOpknapping.Lines.Clear;
        end;
end;
=====
```

```

        redOpknapping.Lines.Add('Oppervlakte: ' + IntToStr(iOppervlakte) + '
vierkante meter' +
        #13 +
        'Volume verf wat benodig word: ' + FloatToStrF(rLiterVerf,
ffFixed, 8,2) +
        ' liter' + #13 + #13 +
        '1 liter-dromme: ' + #9 + IntToStr(iEenLiter) + #13 +
        '2 liter-dromme: ' + #9 + IntToStr(iTweeLiter) + #13 +
        '5 liter-dromme: ' + #9 + IntToStr(iVyfLiter) + #13 +
        #13 +
        'Totale koste: R ' + FloatToStrF(rKoste, ffFixed, 8, 2));
end;//verf

```

// Vraag 1.2.3(b)

```

if cTipeOpknep = 'T' then
begin
    rTeelKoste := StrToFloat(Inputbox('Teelkoste',
        'Sleutel die koste per vierkante meter in', ''));
    rKoste := (iOppervlakte + 5) * rTeelKoste;
    redOpknapping.Lines.Clear;
    redOpknapping.Lines.Add('Oppervlakte: ' + IntToStr(iOppervlakte) + '
vierkante meter' +
        #13 +
        'Totale koste: R ' + FloatToStrF(rKoste, ffFixed, 8, 2));
end;//teel
end;

```

// Vraag 1.3.1

```

=====
procedure TfrmVraag1.btnBerekenBedragClick(Sender: TObject);
var
    rGebruik, rBedrag      : real;
    iVorigeLesing, iHuidigeLesing : integer;
begin
    // Vraag 1.3.1
    iVorigeLesing := StrToInt(edtVorige.Text);
    iHuidigeLesing := StrToInt(edtHuidig.Text);

    if (iVorigeLesing > iHuidigeLesing) then
    begin
        MessageDlg('Vorige lesing groter as huidige lesing. Sleutel weer in',
            mtError, [mbOk], 0);
        edtHuidig.Clear;
        rBedrag := 0;
    end
    else
    begin
        rGebruik := iHuidigeLesing - iVorigeLesing;
        if rGebruik <= 600 then
            rBedrag := rGebruik
        else
            rBedrag := 600 + (rGebruik - 600) * 1.5;
        end;
        lblElektrisiteit.Caption := FloatToStrF(rBedrag, ffCurrency, 8,2);
    end;
end;

```

```
=====
//  Vraag 1.3.2
=====
```

```
procedure TfrmVraag1.btnVindGeisersClick(Sender: TObject);
```

```
var
```

```
    K : integer;
```

```
    sGrootte, sGeiser : String;
```

```
begin
```

```
    // Vraag 1.3.2
```

```
    memGeisers.Clear;
```

```
    for K := 1 to 20 do
```

```
        begin
```

```
            sGeiser := arrGeisers[K];
```

```
            sGrootte := copy(sGeiser, 1, pos('-', sGeiser)-1);
```

```
            Delete(sGeiser, 1, pos('-', sGeiser));
```

```
            if sGrootte = edtGrootte.Text then
```

```
                memGeisers.Lines.Add(sGeiser);
```

```
        end;
```

```
end;
```

```
//=====
```

```
procedure TfrmVraag1.FormCreate(Sender: TObject);
```

```
begin
```

```
    CurrencyString := 'R ';
```

```
end;
```

```
end.
```

BYLAE H: OPLOSSING VIR VRAAG 2: DELPHI**Eenheidklas: Huislening_U**

```

unit Huislening_UMemo;
interface

uses
  SysUtils, Math, Messages, Dialogs;
=====
// Vraag 2.1.1
=====
Type
  THuislening = class(TObject)
    fAansoekerNaam : string;
    fBesteembareInkomste,
    fLeningBedrag  : real;
    fJare           : integer;
    fRentekoers    : real;
  public
    function berekenPaalement : real;
    constructor Create(pAansoekerNaam : String; pBesteembareInkomste,
                      pLeningBedrag : real);

    function GetAansoekerNaam : String;
    procedure SetJare(pJare : integer);
    procedure SetRentekoers(pRentekoers : real);
    function isGoedgekeur : boolean;
    function toString : String;
  end;

implementation

{ THuislening }
=====
// Vraag 2.1.2
=====
constructor THuisLening.Create(pAansoekerNaam: String; pBesteembareInkomste,
  pLeningBedrag: real);
begin
  fAansoekerNaam      := pAansoekerNaam;
  fBesteembareInkomste := pBesteembareInkomste;
  fLeningBedrag       := pLeningBedrag;
  fJare                := 0;
  fRentekoers         := 0;
end;
=====
// Vraag 2.1.3
=====
function THuisLening.GetAansoekerNaam: String;
begin
  result := fAansoekerNaam;
end;
=====
// Vraag 2.1.4
=====
procedure THuisLening.SetJare(pJare: integer);

```

```

begin
  fJare := pJare;
end;

procedure THuisLening.SetRentekoers(pRentekoers: real);
begin
  fRentekoers := pRentekoers;
end;

=====
// Vraag 2.1.5 Gegewe kode (Verwyder kommentaartekens)
=====
function THuisLening.berekenPaaieiment: real;
var
  rBedrag, rKoers : real;
begin
  rKoers := fRenteKoers / 100;
  rBedrag := (rKoers * fLeningBedrag) / (1 - Power((1+rKoers), -fJare));
  Result := Round(rBedrag / 12);
end;

=====
// Vraag 2.1.6
=====
function THuisLening.isGoedgekeur: boolean;
var
  bGeldig : boolean;
begin
  bGeldig := True;

  if (fLeningBedrag > 800000)
  then bGeldig := (fJare > 20);

  if (fLeningBedrag <= 600000)
  then bGeldig := fBesteerbareInkomste > (berekenPaaieiment * 1.3);

  Result := bGeldig;
end;
end.

=====
// Vraag 2.1.7
=====
function THuisLening.toString: String;
var
  sAfvoer : String;
begin
  sAfvoer := 'Naam van aansoeker: ' + fAansoekerNaam + #13;
  sAfvoer := sAfvoer + 'Besteerbare inkomste: ' +
    FloatToStrF(fBesteerbareInkomste, ffCurrency, 8, 2) + #13;
  sAfvoer := sAfvoer + 'Leningbedrag: ' + FloatToStrF(fLeningBedrag,
    ffCurrency, 8, 2) + #13;
  sAfvoer := sAfvoer + 'Getal jaar: ' + IntToStr(fJare) + #13;
  sAfvoer := sAfvoer + 'Rentekoers: ' + FloatToStrF(fRentekoers, ffFixed,
    8, 1) + '%' + #13;
  sAfvoer := sAfvoer + 'Lening GOEDGEKEUR met ' + #13 + 'maandelikse paaieiment van '
    + FloatToStrF(berekenPaaieiment, ffCurrency, 8, 2);
  Result := sAfvoer;
end;

end.

```

Vormklas: Vraag2_P

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

type
  TfrmVraag2 = class(TForm)
    grpLenings: TGroupBox;
    redV2: TRichEdit;
    cboAansoekers: TComboBox;
    pnlVerlaat: TPanel;
    bmbVerlaat: TBitBtn;
    lblRentekoers: TLabel;
    edtRenteKoers: TEdit;
    btnVindAansoeker: TButton;
    btnEvalueer: TButton;
    grpAansoekStatus: TGroupBox;
    grpAansoekers: TGroupBox;
    cboJare: TComboBox;
    lblJare: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure btnEvalueerClick(Sender: TObject);
    procedure btnVindAansoekerClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag2: TfrmVraag2;

implementation

uses
  HuisLening_UMemo;

const
  iAantalJare = 25;

var
  LeningObj : THuisLening;

{$R *.dfm}
{$R+}
```

```
=====
// Vraag 2.2.1
=====

procedure TfrmVraag2.btnVindAansoekerClick(Sender: TObject);
var
  sAansoekerNaam, sLeerNaam: String;
  rInkomste, rUitgawes, rLening, rBesteeInkomste: real;
  TLeer : Textfile;
  sNaam : String;
begin
  if cboAansoekers.ItemIndex = -1
  then
    begin
      MessageDlg('Kies 'n aansoeker', mtError, [mbOk], 0);
      Exit;
    end
  else
    begin
      sAansoekerNaam := cboAansoekers.Items[cboAansoekers.ItemIndex];
      sLeerNaam := sAansoekerNaam + '.txt';
    end;

    redV2.Lines.Clear;
    if NOT FileExists(sLeerNaam)
    then
      begin
        redV2.Lines.Add('Geen huisleningaansoek is deur ' + sAansoekerNaam
+ ' ingedien nie. ');
        btnEvalueer.Enabled := False;
        Exit;
      end;
      AssignFile(TLeer, sLeerNaam);
      Reset(TLeer);
      Readln(TLeer, sNaam);
      Readln(TLeer, rInkomste);
      Readln(TLeer, rUitgawes);
      Readln(TLeer, rLening);
      CloseFile(TLeer);
      redV2.Lines.Add(''n Huisleningaansoek is ingedien deur ' +
sAansoekerNaam);
      rBesteeInkomste := rInkomste - rUitgawes;
      LeningObj := THuisLening.Create(sAansoekerNaam, rBesteeInkomste,
rLening);
      MessageDlg('Huisleningobjek is suksesvol geskep', mtInformation,
[mbOk], 0);
      btnEvalueer.Enabled := true;
    end;
end;

=====
// Vraag 2.2.2
=====

procedure TfrmVraag2.btnEvalueerClick(Sender: TObject);
var
  rRentekoers : real;
  iJare       : integer;
begin
```

```
// Vraag 2.2.2
iJare := StrToInt(cboJare.Items[cboJare.ItemIndex]);
rRentekoers := StrToFloat(edtRenteKoers.Text);

LeningObj.SetJare(iJare);
LeningObj.SetRentekoers(rRentekoers);

redV2.Lines.Clear;
if LeningObj.isGoedgekeur
then
begin
redV2.Lines.Add(LeningObj.toString + #13 +
'Lening GOEDGEKEUR met 'n maandelikse paaient van R ' +
FloatToStrF(LeningObj.berekenPaaient, ffFixed, 8, 2));
end
else
begin
redV2.Lines.Add('Naam van aansoeker: ' + LeningObj.GetAansoekerNaam +
#13 + 'Lening NIE GOEDGEKEUR NIE');
end;
end;
//=====
procedure TfrmVraag2.FormCreate(Sender: TObject);
begin
CurrencyString := 'R ';
// cboAansoekers.ItemIndex := 0;
end;

end.
```

BYLAE I: OPLOSSING VIR VRAAG 3: DELPHI

```

unit Vraag3_UMemo;
//Moontlike oplossing vir Vraag 3
interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, ComCtrls, Buttons, Grids;

type
  TfrmVraag3 = class(TForm)
    grpAgentBesonderhede: TGroupBox;
    lblAgentNaam: TLabel;
    edtKode: TEdit;
    btnAgentNaam: TButton;
    btnAgentVerkoopRekord: TButton;
    grpVertoon: TGroupBox;
    lblAgentKode: TLabel;
    bmbVerlaat: TBitBtn;
    PageControl1: TPageControl;
    tabREdit: TTabSheet;
    tabGrid: TTabSheet;
    redV3: TRichEdit;
    sgrdV3: TStringGrid;
    lblTotaal: TLabel;
    procedure btnAgentNaamClick(Sender: TObject);
    procedure btnAgentVerkoopRekordClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag3: TfrmVraag3;

arrVerkope : array[1..132] of string = ('01#K#A122;500000',
  '01#K#A120;1200000', '01#K#A123;850000', '01#R#A121;75000',
  '02#K#A123;6200000', '02#L#A121;190000', '02#K#A121;500000',
  '02#K#A120;1200000', '02#K#A120;850000', '02#R#A124;75000',
  '02#K#A123;620000', '02#L#A124;690000', '02#K#A121;500000',
  '02#K#A120;1200000', '02#K#A120;850000', '02#R#A124;75000',
  '02#K#A123;620000', '02#L#A124;790000', '02#R#A124;1500000',
  '02#K#A125;200000', '02#L#A121;400000', '03#R#A120;175000',
  '03#K#A123;6200000', '03#L#A123;190000', '03#K#A124;500000',
  '03#K#A125;1200000', '03#K#A123;850000', '03#R#A122;75000',
  '03#K#A123;620000', '03#L#A123;890000', '03#R#A123;1500000',
  '03#K#A125;200000', '03#L#A123;400000', '04#R#A124;175000',
  '04#K#A120;6200000', '04#L#A123;190000', '03#R#A124;1500000',
  '03#K#A125;200000', '03#L#A122;400000', '04#R#A121;175000',
  '04#K#A123;6200000', '04#L#A123;190000', '04#K#A121;500000',
  '04#K#A120;1200000', '04#K#A120;850000', '04#R#A124;75000',
  '04#K#A123;620000', '04#L#A123;590000', '04#K#A123;500000',
  '04#K#A125;120000', '04#K#A124;850000', '04#R#A122;75000',
  '04#K#A123;6200000', '04#L#A120;490000', '04#K#A125;500000',

```

```
'04#K#A125;120000', '04#K#A124;850000', '04#R#A122;75000',
'04#K#A123;6200000', '04#L#A120;390000', '04#R#A120;1500000',
'04#K#A121;200000', '04#L#A122;400000', '04#R#A124;175000',
'04#K#A121;6200000', '04#L#A123;190000', '11#K#A122;500000',
'07#K#A120;1200000', '10#K#A123;850000', '12#R#A121;75000',
'12#K#A123;6200000', '11#L#A121;190000', '12#K#A121;500000',
'12#K#A120;1200000', '11#K#A120;850000', '12#R#A124;75000',
'12#K#A123;620000', '11#L#A124;690000', '09#K#A121;500000',
'08#K#A120;1200000', '11#K#A120;850000', '12#R#A124;75000',
'12#K#A123;620000', '10#L#A124;790000', '08#R#A124;1500000',
'08#K#A125;200000', '11#L#A121;400000', '11#R#A120;175000',
'07#K#A123;6200000', '09#L#A123;190000', '07#K#A124;500000',
'07#K#A125;1200000', '12#K#A123;850000', '10#R#A122;75000',
'08#K#A123;620000', '08#L#A123;890000', '06#R#A123;1500000',
'09#K#A125;200000', '09#L#A123;400000', '09#R#A124;175000',
'08#K#A120;6200000', '07#L#A123;190000', '08#R#A124;1500000',
'06#K#A125;200000', '07#L#A122;400000', '08#R#A121;175000',
'08#K#A123;6200000', '07#L#A123;190000', '11#K#A121;500000',
'06#K#A120;1200000', '08#K#A120;850000', '08#R#A124;75000',
'12#K#A123;620000', '07#L#A123;590000', '12#K#A123;500000',
'07#K#A125;120000', '08#K#A124;850000', '07#R#A122;75000',
'11#K#A123;6200000', '09#L#A120;490000', '08#K#A125;500000',
'07#K#A125;120000', '09#K#A124;850000', '06#R#A122;75000',
'10#K#A123;6200000', '09#L#A120;390000', '10#R#A120;1500000',
'07#K#A121;200000', '09#L#A122;400000', '06#R#A124;175000',
'10#K#A121;6200000', '09#L#A123;190000');
```

```
arrAgente:array[1..6] of string = ('A120:Wes Seelig',
'A121:Tandra Peters',
'A122:Lisha Champ',
'A123:Melda Bissonette',
'A124:Leta Hansell',
'A125:Sid Rebello');
```

```
implementation
```

```
{ $R *.dfm }
```

```
{ $R+ }
```

```
var
```

```
    sAgentKode : String = '';
```

```
    sAgentNaam : String = '';
```

```
=====
// Vraag 3.1
=====
```

```
procedure TfrmVraag3.btnAgentNaamClick(Sender: TObject);
```

```
var
```

```
    K      : integer;
```

```
    bGevind : boolean;
```

```
begin
```

```
    // Vraag 3.1
```

```
    sAgentKode := edtKode.Text;
```

```
    bGevind := false;
```

```
    K := 1;
```

```
    while (K <= Length(arrAgente)) AND NOT bGevind do
```

```
    begin
```

```
        if pos(sAgentKode, arrAgente[K]) = 1 then
```

```
        begin
```

```

        sAgentNaam := copy(arrAgente[K], pos(':', arrAgente[K])+1,
length(arrAgente[K]));
        lblAgentNaam.Caption := sAgentNaam;
        bGevind := True;
    end
else
    inc(K, 1);
end;//while
if NOT bGevind then
    begin
        edtKode.Clear;
        lblAgentNaam.Caption := 'Ongeldige agentkode';
    end;
end;

```

```

=====
// Vraag 3.2.1
=====

```

Plaas 'n geskikte afvoerkomponent op die GGK (GUI)
Hierdie oplossing maak gebruik van 'n teksarea-komponent
'n Grid kan ook gebruik word met 'n label-komponent om totale verkope te
vertoon.

```

=====
// Vraag 3.2.2
=====

```

```

procedure TfrmVraag3.btnAgentVerkoopRekordClick(Sender: TObject);
var
    arrAgentVerkope : array[1..3, 1..13] of integer;
    iIndex, iRy, iKol, iPrys, iTotaal, iRyTotaal : integer;
    sTemp : String;
begin
    // Vraag 3.2
    //inisialiseer 2D-skikking
    for iRy := 1 to 3 do
        for iKol := 1 to 13 do
            arrAgentVerkope[iRy, iKol] := 0;
        //analiseer verkope-skikking
        iTotaal := 0;
        for iIndex := 1 to length(arrVerkope) do
            begin
                sTemp := arrVerkope[iIndex];
                if pos(sAgentKode, sTemp) > 0 then
                    begin
                        iKol := StrToInt(Copy(sTemp, 1, pos('#', sTemp)-1));
                        iPrys := StrToInt(Copy(sTemp, pos(':', sTemp)+1, length(sTemp)));
                        inc(iTotaal, iPrys);

                        Delete(sTemp, 1, pos('#', sTemp));
                        case sTemp[1] of //tipe verkope
                            'K' : iRy := 1;
                            'R' : iRy := 2;
                            'L' : iRy := 3;
                        end;//case
                        inc(arrAgentVerkope[iRy, iKol], 1);
                    end; //if
                end;//for
            //bepaal ry-totale
            for iRy := 1 to 3 do

```

```

begin
  iRyTotaal := 0;
  for iKol := 1 to 12 do
    begin
      inc(iRyTotaal, arrAgentVerkope[iRy, iKol]);
    end;
  arrAgentVerkope[iRy, 13] := iRyTotaal;
end;

//Afvoer: RichEdit
redV3.Lines.Clear;
redV3.Paragraph.TabCount := 1;
redV3.Paragraph.Tab[0] := 70;
sTemp := ' ' + #9 + 'Jan' + #9 + 'Feb' + #9 + 'Mrt' + #9 + 'Apr' + #9 +
'Mei' + #9 +
      'Jun' + #9 + 'Jul' + #9 + 'Aug' + #9 + 'Sep' + #9 + 'Okt' + #9 +
      'Nov' + #9 + 'Des';
redV3.Lines.Add(sTemp);
redV3.Lines.Add(' ');
sTemp := 'Kommersieel' + #9;
for iKol := 1 to 13 do
  sTemp := sTemp + IntToStr(arrAgentVerkope[1, iKol]) + #9 ;
redV3.Lines.Add(sTemp);

redV3.Lines.Add(' ');
sTemp := 'Residensieel' + #9;
for iKol := 1 to 13 do
  sTemp := sTemp + IntToStr(arrAgentVerkope[2, iKol]) + #9 ;
redV3.Lines.Add(sTemp);

redV3.Lines.Add(' ');
sTemp := 'Landbou' + #9;
for iKol := 1 to 13 do
  sTemp := sTemp + IntToStr(arrAgentVerkope[3, iKol]) + #9 ;
redV3.Lines.Add(sTemp);

redV3.Lines.Add(' ');
redV3.Lines.Add('Totale waarde van verkope: R ' + IntToStr(iTotaal));

//Afvoer: StringGrid
//Kolomopskrifte
{ sgrdV3.Cells[1,0] := 'Jan';
sgrdV3.Cells[2,0] := 'Feb';
sgrdV3.Cells[3,0] := 'Mrt';
sgrdV3.Cells[4,0] := 'Apr';
sgrdV3.Cells[5,0] := 'Mei';
sgrdV3.Cells[6,0] := 'Jun';
sgrdV3.Cells[7,0] := 'Jul';
sgrdV3.Cells[8,0] := 'Aug';
sgrdV3.Cells[9,0] := 'Sep';
sgrdV3.Cells[10,0] := 'Okt';
sgrdV3.Cells[11,0] := 'Nov';
sgrdV3.Cells[12,0] := 'Des';
//Ryopskrifte.
sgrdV3.Cells[0,1] := 'Kommersieel';
sgrdV3.Cells[0,2] := 'Residensieel';
sgrdV3.Cells[0,3] := 'Landbou';

```

```
sgrdV3.ColWidths[0] := 100;

for iRy := 1 to 3 do
  for iKol := 1 to 13 do
    sgrdV3.Cells[iKol, iRy] := IntToStr(arrAgentVerkope[iRy, iKol]);

  lblTotaal.Caption := 'Totale waarde van verkope: R' + IntToStr(iTotaal); }
end;

end.
```