



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIORSERTIFIKAAT-EKSAMEN

INLIGTINGSTEGNOLOGIE V1

2017

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 slegs wat in elke vraag gevra word. Byvoorbeeld, indien die vraag nie datavalidering vra nie, sal geen punte vir datavalidering toegeken word nie.
6. Jou programme moet op só 'n manier gekodeer word dat dit met enige data sal werk en nie net met die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel verskyn nie.
7. Roetines, soos soek, sorteer en seleksie, moet vanuit eerste beginsels ontwikkel word. Jy mag NIE die ingeboude funksies van 'n programmeringstaal vir enige van hierdie roetines gebruik NIE.
8. Alle datastrukture moet deur jou, die programmeerder, verklaar word, tensy die datastrukture voorsien word.
9. Jy moet jou werk gereeld 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 eksamennummer 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 die disket/CD/DVD/geheuestokkie of op die skyfspasie wat aan jou toegeken is. Die lêers word in die vorm van wagwoordbeskermd, uitvoerbare lêers verskaf.

LET WEL:

- Delphi-kandidate moet die lêer **DelphiDataAFRJunie2017.exe** gebruik.
- Java-kandidate moet die lêer **JavaDataAFRJunie2017.exe** gebruik.

Doen die volgende:

- Dubbelklik op die wagwoordbeskermd uitvoerbare lêer.
- Klik op die 'Extract'-knoppie.
- Sleutel die volgende wagwoord in: **AiR%\$Port**

Nadat dit onttrek ('extracted') is, sal die volgende lys met lêers in die lêergids **DelphiDataAFRJunie2017/JavaDataAFRJunie2017** beskikbaar wees:

DELPHI-LÊERS**Vraag1:**

Goud.jpg
NieLidNie.jpg
Platinum.jpg
Silwer.jpg
Vraag1_P.dpr
Vraag1_P.dprj
Vraag1_P.res
Vraag1_U.dfm
Vraag1_U.pas

Vraag2:

DataV2.txt
Vlug_U.pas
Vraag2_P.dpr
Vraag2_P.dprj
Vraag2_P.res
Vraag2_U.dfm
Vraag2_U.pas

Vraag3:

Vraag3_P.dpr
Vraag3_P.dprj
Vraag3_P.res
Vraag3_U.dfm
Vraag3_U.pas

JAVA(NETBEANS)-LÊERS**Vraag1:**

Goud.jpg
NieLidNie.jpg
Platinum.jpg
Silwer.jpg
Vraag1.form
Vraag1.java

Vraag2:

DataV2.txt
Vlug.java
Vraag2.form
Vraag2.java

Vraag3:

Vraag3.form
Vraag3.java
VulSkikkings.java

SCENARIO

'n Aantal verskillende sagtewareprogramme word by 'n lughawe gebruik om die personeel te help om besonderhede van vlugte en passasiers te bevestig en te valideer.

AFDELING A**VRAAG 1: ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

'n Sagtewareprogram word deur die lugredery gebruik om die aanvanklike registrasie-inligting van 'n passasier vas te lê. Hierdie inligting word gebruik om die passasier se profiel op te dateer en om die lugredery te help met die spyseniering vir die passasiers aan boord.

INSTRUKSIES:

DELPHI-PROGRAMMEERERS	JAVA-PROGRAMMEERERS
<ul style="list-style-type: none"> Maak die onvolledige projeklêer met die naam Vraag1_P.dpr in die Vraag1-lêergids oop. Sleutel jou eksamennummer as kommentaar in die eerste reël van die hoofvormeenheid met die naam Vraag1_U.pas in. 	<ul style="list-style-type: none"> Maak die onvolledige klas met die naam Vraag1.java in die lêergids Source Packages (src), Vraag1Package in die Vraag1-lêergids oop. Sleutel jou eksamennummer as kommentaar in die eerste reël van die klas met die naam Vraag1.java in.

Doen die volgende:

- Kompileer en voer die program uit. Die grafiese gebruikerskoppelvlak (GGK/'GUI') vertoon vyf afdelings met die naam Vraag 1_1 tot Vraag 1_5. Die program het tans geen funksionaliteit nie.
- Voltooi die kode vir die program, soos beskryf in VRAAG 1.1 tot VRAAG 1.5 op die volgende bladsy.

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

The screenshot shows a GUI with the following elements:

- Vraag1_1:** "Ouderdom van passasier" (text box), "Geldige paspoort" (checkbox), "Vergesel van volwassene" (checkbox), "Vraag 1.1" (button), and an empty text box.
- Vraag1_2:** "Gewig van bagasie in kg" (text box), "Lugredery#Maksimum gewig" (list box with options: ZAA#25kg, UKAL#30kg, MAO#20kg, KAA#15kg, SA#5kg, ETTAL#100kg), "Vraag 1.2" (button), and an empty text box.
- Vraag1_3:** "Getal passasiers" (text box), "Vraag 1.3" (button), and an empty text box.
- Vraag1_4:** "Vertrektyd" (text box), "Instaptyd" (text box), "Vraag 1.4" (button).
- Vraag1_5:** "Afstand in kilometer" (text box), "Vlugkaart" (radio buttons for Silver, Goud, Platinum), "Vraag 1.5" (button), and a large box containing "Nie 'n lid nie".

- 1.1 Vir die doel van hierdie vraag kan jy aanneem dat alle passasiers op internasionale vlugte reis. Passasiers op internasionale vlugte moet 'n geldige paspoort hê en passasiers jonger as 16 jaar moet van 'n volwassene vergesel word.

Skryf kode om die passasier se ouderdom uit die teksblokkie ('text box') wat verskaf is, te verkry, te toets of die korrekte kontroleblokkies ('check boxes') afgemerks is en 'n geskikte boodskap te vertoon wat aandui of daar bevestig is dat die passasier mag instap ('board'), of nie.

Voorbeeld van afvoer indien die passasier 23 jaar oud is en 'n geldige paspoort het:

The screenshot shows the following state:

- Vraag1_1:** "Ouderdom van passasier" (text box) contains "23". "Geldige paspoort" (checkbox) is checked. "Vergesel van volwassene" (checkbox) is unchecked. "Vraag 1.1" (button) is highlighted. The text box next to it contains "Instap is bevestig."

Voorbeeld van afvoer indien die passasier 14 jaar oud is en 'n geldige paspoort het, maar nie van 'n volwassene vergesel word nie:

Vraag1_1

Ouderdom van passasier

14

Geldige paspoort

Vergesel van volwassene

Vraag 1.1

Instap is nie bevestig nie.

Voorbeeld van afvoer indien die passasier 14 jaar oud is, 'n geldige paspoort het en van 'n volwassene vergesel word:

Vraag1_1

Ouderdom van passasier

14

Geldige paspoort

Vergesel van volwassene

Vraag 1.1

Instap is bevestig.

(7)

- 1.2 'n Lysblokkie ('list box') met die name van verskillende lugrederye en die maksimum bagasiegewig wat per passasier toegelaat word, is voorsien. Die formaat van die teks in die lysblokkie is soos volg:

<naam van lugredery>#<maksimum gewig>kg

'n Bedrag van R50,00 per kilogram moet betaal word indien die passasier se bagasiegewig die maksimum gewig wat die lugredery spesifiseer, oorskry. Die bedrag wat per kilogram betaal moet word, is as 'n konstante verklaar in die kode wat verskaf is.

Skryf kode om die volgende te doen:

- Verkry die passasier se bagasiegewig uit die teksblokkie ('text box') en die maksimum bagasiegewig wat vir die lugredery wat in die lysblokkie ('list box') gekies is, toegelaat word.
- Gebruik die inligting wat verkry is om die oortollige gewig, indien enige, te bereken.
- Gebruik die konstante veranderlike en die oortollige gewig wat bereken is om die koste van die oortollige gewig van die passasier se bagasie te bepaal.
- Vertoon die oortollige gewig en die koste van die oortollige gewig. Die koste moet tot 'n geldeenheid met TWEE desimale plekke geformateer word.

Voorbeeld van afvoer indien die passasier se bagasie 35,89 kg (35.89kg) weeg en UKAL-lugredery in die lysblokkie ('list box') geselekteer is:

Gewig van bagasie in kg	Lugredery#Maksimum gewig
35.89	ZAA#25kg UKAL#30kg MAO#20kg KAA#15kg SA#5kg ETTAL#100kg
Oortollige gewig: 5.89kg Koste: R294.50	

Vraag 1.2

Voorbeeld van afvoer indien die passasier se bagasie 12,5 kg (12.5kg) weeg en KAA-lugredery in die lysblokkie ('list box') geselekteer is:

Gewig van bagasie in kg	Lugredery#Maksimum gewig
12.5	ZAA#25kg UKAL#30kg MAO#20kg KAA#15kg SA#5kg ETTAL#100kg
Oortollige gewig: 0.00kg Koste: R0.00	

Vraag 1.2

(10)

- 1.3 Die vlugspysenieringsmaatskappy berei passasiers se etes vooraf om etes vir die getal passasiers op die vlug te verskaf. Die ete-opsies is vegetaries en nie-vegetaries. Die standaardbeleid van die spysenieringsmaatskappy is om een derde vegetariese etes en twee derdes nie-vegetariese etes te berei. Indien die getal passasiers nie presies in drie dele verdeel kan word nie, moet die oorblywende getal voorbereide etes, nadat dit deur drie gedeeltes is, nie-vegetaries wees.

Skryf kode om die volgende te doen:

- Verkry die getal passasiers uit die teksblokkie ('text box') wat verskaf is.
- Bepaal en vertoon die getal vegetariese etes en die getal nie-vegetariese etes wat vir die passasiers berei moet word.

Voorbeeld van afvoer indien daar 180 passasiers is:

Voorbeeld van afvoer indien daar 242 passasiers is:

(5)

- 1.4 Die instaptyd van vlugte is vyf-en-dertig minute voor die vertrektyd. Die gebruiker moet die vertrektyd in die formaat <uur>h<minuut> insleutel.

Skryf kode om die volgende te doen:

- Verkry die vertrektyd uit die teksblokkie ('text box') wat verskaf is.
- Valideer die vertrektyd om seker te maak dat die volgende geld:
 - Die formaat moet <uur>h<minuut> wees.
 - Die uur-waarde moet minder as of gelyk aan 23 wees en moet uit twee syfers bestaan.
 - Die minuut-waarde moet minder as of gelyk aan 59 wees en moet uit twee syfers bestaan.
- Indien die vertrektyd ongeldig is, gebruik 'n dialoogblokkie ('dialog box') om 'n geskikte boodskap te vertoon en maak die teksblokkie ('text box') vir vertrektyd leeg.
- Indien die vertrektyd geldig is, bereken en vertoon die instaptyd in die teksblokkie ('text box') wat voorsien is.

Voorbeeld van afvoer indien die vertrektyd 14h40 is:

Voorbeeld van afvoer indien die vertrektyd 17h15 is:

Voorbeeld van afvoer indien die vertrektyd ongeldig is of in die verkeerde formaat ingesleutel is, byvoorbeeld 15:67:

(17)

- 1.5 'n Passasier wat 'n vlugkaart ('flyer card') het, kan punte verdien, afhangend van die afstand wat gereis is en of die passasier 'n silwer-, goue of platinumkaart het. Die lys voordele vir elke tipe vlugkaart word as aparte prentlêers ('image files') gestoor. Die name van die prentlêers is **Silwer.jpg**, **Goud.jpg**, **Platinum.jpg** en **NieLidNie.jpg**. Die verstekprentlêer, **NieLidNie.jpg**, word in die prentkomponent ('image component') wat verskaf is, vertoon.

Skryf kode om die volgende te doen:

- Verkry die afstand wat gereis is, uit die teksblokkie ('text box') wat verskaf is.
- Selekteer die tipe vlugkaart uit die radioknoppies ('radio buttons') wat verskaf is.
- Vertoon die ooreenstemmende prentlêer vir die vlugkaart wat geselekteer is.
- Bereken die punte wat verdien is, soos volg:
 - Alle passasiers met kaarte sal een punt verdien vir elke 1,6 kilometer gereis.
 - Passasiers met goue kaarte sal addisionele bonuspunte verdien, wat as 15% van die afstand wat gereis is, bereken word.
 - Passasiers met platinumkaarte sal addisionele bonuspunte verdien, wat as 20% van die afstand wat gereis is, bereken word.
 - Die totale punte moet tot die naaste heelgetal na onder afgerond word.
- Die paneel met die naam **pnIPunte** moet vertoon word indien 'n kaart geselekteer is. Die totale punte wat verdien is, moet in hierdie paneel vertoon word.

Voorbeeld van afvoer indien die afstand 1 233 kilometer is en die passasier 'n silwervlugkaart het:

Vraag1_5

Afstand in kilometer

Vlugkaart

Silwer

Goud

Platinum

Silwerkaart-lid

- Voorkeur-instap
- Bespreekte sitplek
- Gratis myle

Vraag 1.5

Punte verdien: 770

Voorbeeld van afvoer indien die afstand 1 560 kilometer is en die passasier 'n goue vlugkaart het:

Vraag1_5

Afstand in kilometer

Vlugkaart

Silwer

Goud

Platinum

Gouekaart-lid

- Voorkeur-instap
- Bespreekte sitplek
- Toegang tot besigheidsklassitkamer
- Gratis myle en bonusmyle
- Voorkeursekerheidskontroles

Vraag 1.5

Punte verdien: 1209

(11)

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

TOTAAL AFDELING A: 50

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

Lugrederye stoor besonderhede van vlugte en passasiers om personeel te help om inligting onmiddellik te verifieer.

Doen die volgende:

DELPHI-PROGRAMMEERDERS	JAVA-PROGRAMMEERDERS
<ul style="list-style-type: none"> • Maak die onvolledige projeklêer met die naam Vraag2_P.dpr en die klas met die naam Vlug_U.pas oop. • Sleutel jou eksamennommer as kommentaar in die eerste reël van beide die lêers Vraag2_U.pas en Vlug_U.pas in. 	<ul style="list-style-type: none"> • Maak die onvolledige klasse met die name Vraag2.java en Vlug.java in die lêergids Source Packages (src), Vraag2Package oop. • Sleutel jou eksamennommer as kommentaar in die eerste reël van beide die klasse Vraag2.java en Vlug.java in.

- Voltooi die kode vir hierdie program soos in VRAAG 2.1 en VRAAG 2.2 hieronder gespesifiseer word.

2.1 'n Gedeeltelik voltooide objekklas met die naam **TVlug/Vlug** is verskaf. Dit bevat die attribute van 'n vlug en die kode vir een metode.

Voltooi die kode in die gegewe vlugklas met die naam **TVlug/Vlug**, soos beskryf in VRAAG 2.1.1 tot VRAAG 2.1.6 wat volg.

Die tabel hieronder bevat beskrywings van die attribute van 'n vlugobjek.

NAME VAN ATTRIBUTE		BESKRYWING
DELPHI	JAVA	
fVlugNommer	vlugNommer	Vlugnommer
fStad	stad	Naam van die stad as bestemming
fDatum	datum	Datum van die vlug in die formaat JJJJ/MM/DD
fGetalPassasiers	getalPassasiers	Getal passasiers wat op die vlug bespreek is

- 2.1.1 Skryf kode vir 'n konstrktor om die volgende te doen:
- Ontvang die vlugnommer, stad van bestemming en datum van die vlug as parameters.
 - Ken die betrokke parameterwaardes aan die onderskeie attribute toe.
 - Roep die gegewe **setGetalPassasiers**-metode om die **fGetalPassasiers/getalPassasiers**-attribuut op 0 te stel. (5)
- 2.1.2 Skryf toegangsmetodes ('accessor methods') vir die **fVlugNommer/vlugNommer**- en **fGetalPassasiers/getalPassasiers**-attribuut. (4)
- 2.1.3 Skryf 'n metode met die naam **vermeerderPassasiers** om die attribuut vir die getal passasiers met 1 te vermeerder. (2)
- 2.1.4 Skryf 'n metode met die naam **berekenPersBespreek** wat die maksimum getal passasiers vir die vlug as 'n parameter sal ontvang en die persentasie sitplekke wat bespreek is, sal bereken en terugstuur.

Gebruik die formule:

$$\text{passasiers bespreek} / \text{maksimum passasiers} * 100 \quad (4)$$

- 2.1.5 Skryf 'n **toString**-metode om die besonderhede van 'n vlug in die volgende formaat te vertoon:

```
Vlugnommer: <vlugnommer>
Bestemming: <stad>
Vertrekdatum: <datum>
Getal passasiers bespreek: <getal passasiers bespreek>
```

Voorbeeld van afvoer:

```
Vlugnommer: SA528
Bestemming: Johannesburg
Vertrekdatum: 2017-07-22
Getal passasiers bespreek: 13
```

(5)

2.2 Grafiese gebruikerskoppelvlak (GGK/'GUI'):

Data wat voorsien is:

'n Tekslêer met die naam **DataV2.txt** is voorsien. Elke teksreël in die lêer bevat data van 'n passasier wat vir 'n vlug bespreek is, in die volgende formaat:

```
<vlugnommer>-<passasier se nommer>vier spasies<passasier se naam>
```

Voorbeeld van die eerste ses teksreëls wat in die tekslêer gestoor is:

```
SA528-01    Gregory Thomas
SA528-02    Henry Kensington
MA230-01    Sebastian Johnson
KU137-01    Henrietta Botha
BA630-01    Samson Nduli
BA630-02    Mary Nduli
```

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

Reël 1: Op vlug **SA528** is **Gregory Thomas** die naam van passasier nommer **1**.

Reël 2: Op vlug **SA528** is **Henry Kensington** die naam van passasier nommer **2**.

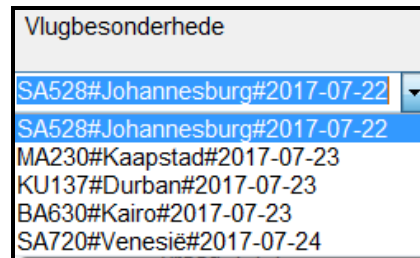
Doen die volgende om die kode vir die knoppies in die hoofvormeenheid (Delphi)/GGK('GUI')-klas (Java) te voltooi, soos hieronder beskryf.

LET WEL: Die objek **objVlug** is in die program verklaar.

2.2.1 Knoppie [Vraag 2.2.1]

'n Kombinasieblokkie ('combo box') wat die besonderhede van die verskillende vlugte bevat, is gegee.

Die inhoud van die kombinasieblokkie ('combo box') is soos volg:



Die formaat van elke teksreël in die kombinasieblokkie ('combo box') is soos volg:

```
<vlugnommer>#<stad van bestemming>#<datum van vlug>
```

Die data in die eerste reël van die kombinasieblokkie ('combo box') kan soos volg geïnterpreteer word:

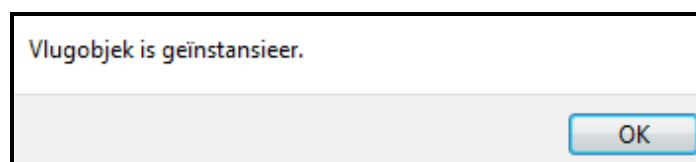
Reël 1: Vlug **SA528** na **Johannesburg** sal op **2017-07-22** vertrek.

Die gebruiker moet 'n vlug uit die kombinasieblokkie ('combo box') selekteer.

Skryf kode om die volgende te doen:

- Gebruik geselekteerde vlugbesonderhede om die objek **objVlug** te instansieer.
- Aktiveer die knoppies vir VRAAG 2.2.2 en VRAAG 2.2.3.
- Vertoon 'n boodskap wat aandui dat die vlugobjek geïnstansieer is.

Voorbeeld van afvoer:



(12)

2.2.2 Knoppie [Vraag 2.2.2]

Skryf kode om die volgende te doen:

- Kontroleer of die **DataV2.txt**-tekst lêer bestaan, of nie.
- Indien die tekst lêer NIE bestaan NIE, vertoon 'n gepaste boodskap en maak die program toe.

- Doen die volgende indien die tekslêer bestaan:
 - Laat loop met 'n lus deur die lêer.
Lees 'n teksreël (passasier se besonderhede) in die lêer.
Indien die passasier wat in die lêer gelees is, op die geselekteerde vlug is:
 - Vertoon die teksreël soos wat dit in die tekslêer gelees is.
 - Gebruik die **vermeerderPassasiers**-metode om die getal passasiers op te dateer.
 - Gebruik die **toString**-metode om die opgedateerde inligting van die vlug te vertoon.

Voorbeeld van afvoer van vlug MA230 na Kaapstad:

```
Lys van passasiers
MA230-01 Sebastian Johnson
MA230-02 Jeffrey Gordan
MA230-03 Jenny Gordan
MA230-04 Mala Govender
MA230-05 Mannie Govender
MA230-06 Tyeel Govender
MA230-07 Kaitlyn Green
MA230-08 Stefan Green
MA230-09 Johannes Mbatha

Vlugnommer: MA230
Bestemming: Kaapstad
Vertrekdatum: 2017-07-23
Getal passasiers bespreek: 9
```

(15)

2.2.3 Knoppie [Vraag 2.2.3]

Gebruik 'n dialoogblokkie ('dialog box') om die gebruiker te vra om die maksimum getal passasiers vir die vlug in te sleutel. Gebruik die **berekenPersBespreek**-metode om die persentasie sitplekke wat vir die vlug bespreek is, te bepaal. Vertoon 'n geskikte boodskap indien die vlug vol bespreek is, byvoorbeeld:



Indien die vlug NIE vol bespreek is NIE, vertoon die besprekingspersentasie tot EEN desimale plek en gebruik 'n dialoogblokkie ('dialog box') om die naam van 'n nuwe passasier in te sleutel. Die nuwe passasier se besonderhede moet in die tekslêer gestoor word en die getal passasiers wat bespreek is, moet opgedateer word. Die opgedateerde inligting moet in die afvoerarea vertoon word.

Die formaat van al die reëls in die tekslêer **DataV2.txt** moet soos volg wees:

<vlugnommer>-<passasiernommer>4 spasies<passasier se naam>

LET WEL: Die eerste passasier op 'n vlug se nommer is 01, die tweede passasier se nommer is 02, ensovoorts.

Voorbeeld van toevoer van die gebruiker en afvoer indien vlug MA230 geselekteer is, die maksimum getal passasiers 50 is en die vlug nie vol bespreek is nie. Graham Barkley word as 'n passasier by die vlug gevoeg:

Sleutel die maksimum getal passasiers in

OK Cancel

Persentasie bespreek: 18.0%

OK

Naam van nuwe passasier:

OK Cancel

Lys van passasiers

MA230-01	Sebastian Johnson
MA230-02	Jeffrey Gordan
MA230-03	Jenny Gordan
MA230-04	Mala Govender
MA230-05	Mannie Govender
MA230-06	Tyeel Govender
MA230-07	Kaitlyn Green
MA230-08	Stefan Green
MA230-09	Johannes Mbatha
MA230-10	Graham Barkley

Vlugnommer: MA230
Bestemming: Kaapstad
Vertrekdatum: 2017-07-23
Getal passasiers bespreek: 10

(13)

- Sleutel jou eksamennommer as kommentaar in die eerste reël van die klas en die vorm in.
- Stoor jou program.
- Druk die kode van beide die klas en die vorm, indien dit vereis word.

TOTAAL AFDELING B: 60

AFDELING C**VRAAG 3: PROBLEEMOPLOSSINGSPROGRAMMERING**

Die nuwe lugrederymaatskappy, Soaring Eagles, wil die intekenproses van hulle passasiers optimaliseer. Die getal toonbanke wat oopgemaak en deur hulle personeel beman sal word, hang af van die getal kliënte wat in die ry staan. Indien 'n vlug vertraag word, sal die passasiers wat op die vertraagde vlug bespreek is, gevra word om in 'n ry by 'n aparte toonbank te staan.

Doen die volgende:

DELPHI-PROGRAMMEERders	JAVA-PROGRAMMEERders
<ul style="list-style-type: none"> • Maak die onvolledige projeklêer met die naam Vraag3_P.dpr in die Vraag3-lêergids oop. • Sleutel jou eksamennommer as kommentaar in die eerste reël van die hoofvormeenheid met die naam Vraag3_U.pas in. 	<ul style="list-style-type: none"> • Maak die onvolledige klas met die naam Vraag3.java in die lêergids Source Packages (src), Vraag3Package in die Vraag3-lêergids oop. • Voeg jou eksamennommer as kommentaar in die eerste reël van die klas met die naam Vraag3.java in.

Lees die volgende afdelings voordat jy die oplossing aanpak:

- GGK ('GUI') en data wat voorsien is

- Programvereistes

- Puntetoekenning

LET WEL:

- Jy mag NIE die data wat voorsien is, per hand verander NIE. Kode moet bygevoeg word om die data wat voorsien is, volgens die vereistes te manipuleer.

- Die gebruik van goeie programmeringstegnieke en modulêre ontwerp moet in die ontwerp en kodering van jou oplossing toegepas word.

Grafiese gebruikerskoppelvlak (GGK/'GUI') wat voorsien is:

Die GGK ('GUI') wat voorsien is, bevat komponente vir gebeurtenisse wat moet plaasvind.

Gebruik die programvereistes in die vrae wat volg om op 'n geskikte komponent te besluit om in die afvoerarea met die opskrif **Lugredery se toonbanke** in die GGK ('GUI') te plaas.

Data wat voorsien is:

'n Eendimensionele skikking met die naam **arrPassasiers** is verklaar en sal die besonderhede van passasiers wat hulle instapkaarte ('boarding passes') by die vertrektoonbanke by die lughawe moet afhaal, bevat. Kode wat voorsien is, sal die **arrPassasiers**-skikking vul met 'n onbekende aantal elemente wat uit die **arrPosPassasiers**-skikking verkry word wanneer die program uitgevoer word. Die aantal elemente wat aan die **arrPassasiers**-skikking toegeken word, sal in die reeks van 1 tot 35 wees en sal willekeurig bepaal word.

Jy moet die **arrPassasiers**-skikking in jou kode gebruik.

Die formaat van die data in die **arrPassasiers**-skikking is soos volg:

(B – besigheidsklas of E – ekonomiese klas), die posisie van die passasier in die ry (begin by 01) en die vlugnommer, soos hieronder getoon:

```
<Passasier se klas><posisie van passasier in die ry>;<vlugnommer>
```

Voorbeeld van inligting van passasiers in die **arrPassasiers**-skikking:

```
E01;TDB2506
B02;TDB2506
E01;TDB1305
```

'n Tweedimensionele skikking met die naam **arrGrid** met nege rye en vier kolomme is verklaar. In jou oplossing sal hierdie skikking nie noodwendig 36 elemente bevat nie aangesien die getal passasiers wat in die **arrPassasiers**-skikking voorkom, die getal elemente wat in die **arrGrid**-skikking voorkom, sal bepaal.

Programvereistes:

3.1 Knoppie [3.1 – Rye by vertrektoonbanke]

Die program moet die passasiers wat vir hulle instapkaarte wag (wat in die **arrPassasiers**-skikking voorkom), in rye by die vertrektoonbanke plaas. Die getal passasiers wat in die **arrPassasiers**-skikking voorkom, bepaal die getal vertrektoonbanke wat oop is deur die volgende kriteria te gebruik:

GETAL OOP VERTREKTOONBANKE	GETAL PASSASIERE WAT INSTAPKAARTE MOET KRY
1	Minder as 10
2	10 tot 16
3	17 tot 24
4	Meer as 24

Doen die volgende:

- Gebruik die **arrGrid**- tweedimensionele skikking om die vertrektoonbanke wat oop is en passasiers wat aan hierdie toonbanke toegeken is, voor te stel. Passasiers moet soos volg aan toonbanke toegeken word:
 - Besigheidsklas-passasiers moet eerste in die rye by die vertrektoonbanke geplaas word, gevolg deur ekonomieseklas-passasiers.
 - Die passasiers moet so gelykop as moontlik onder die rye by die vertrektoonbanke verdeel word, soos in die voorbeelde van afvoer wat volg, getoon word.
- Vertoon geskikte opskrifte vir die vertrektoonbanke.
- Vertoon die klas en nommer van die passasiers.

Voorbeeld van afvoer indien 16 passasiers wag om instapkaarte te kry:

Toonbank 1	Toonbank 2
B06	B08
B10	B12
B13	B14
E01	E02
E03	E04
E05	E07
E09	E11
E15	E16

Voorbeeld van afvoer indien 33 passasiers wag om instapkaarte te kry:

Toonbank 1	Toonbank 2	Toonbank 3	Toonbank 4
B06	B08	B10	B12
B13	B14	B22	B23
B27	B28	B31	E01
E02	E03	E04	E05
E07	E09	E11	E15
E16	E17	E18	E19
E20	E21	E24	E25
E26	E29	E30	E32
E33			

LET WEL: Die afvoer word bepaal deur die getal elemente wat die **arrPassasiers**-skikking bevat.

3.2 Knoppie [3.2 – Passasiers op vertraagde vlug]

Indien 'n vlug vertraag is, moet die passasiers wat op die vertraagde vlug bespreek is, uit die rye by die oop vertrektoonbanke verwyder word en in 'n ry by 'n aparte toonbank geplaas word. 'n Kombinasieblokkie ('combo box') met vlugnommers word verskaf om die vlugnommer van 'n vertraagde vlug te kies.

Die gebruiker moet die vertraagde vlug selekteer en op die VRAAG 3.2-knoppie klik.

Die program moet die volgende doen:

- Verwyder die inligting van al die passasiers wat op die vertraagde vlug bespreek is, uit die rye by die vertrektoonbanke en vertoon hulle besonderhede in die afvoerarea wat verskaf is.
- Dateer die rye wat by die vertrektoonbanke vertoon word, op. Die passasiers by die vertrektoonbanke moet steeds gelykop onder die toonbanke verdeel wees; eers die besigheids- en dan die ekonomieseklas-passasiers.

Voorbeeld van afvoer indien vlug TDB2506 uit die kombinasieblokkie ('combo box') as vertraagde vlug gekies is en die **arrPassasiers**-skikking 33 elemente bevat:

Toonbank 1	Toonbank 2	Toonbank 3	Toonbank 4	Vlugnommer: TDB2506
B06	B08	B12	B14	B10
B22	B27	B28	B31	B13
E02	E03	E07	E09	B23
E11	E16	E18	E19	E01
E20	E21	E24	E25	E04
E26	E29			E05
				E15
				E17
				E30
				E32
				E33

Puntetoekenning:

VEREISTES	MAKSIMUM PUNTE
VRAAG 3.1	
Bepaal getal oop toonbanke en getal passasiers per ry	6
Ken posisie volgens die klas van die passasier toe	5
Kopieer inligting uit die gegewe skikking na die regte datastruktuur	8
Vertoon passasiers in rye by korrekte getal toonbanke	7
VRAAG 3.2	
Bepaal en vertoon passasiers op die vertraagde vlug	6
Verwyder passasiers op vertraagde vlug uit rye	6
Dateer inligting in datastrukture en afvoer op	2

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

TOTAAL AFDELING C: 40
GROOTTOTAAL: 150



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

SENIORSERTIFIKAAT-EKSAMEN

INLIGTINGSTEGNOLOGIE V1

2017

NASIENRIGLYNE

PUNTE: 150

Hierdie nasienriglyne bestaan uit 30 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as die basis vir die nasiensessie gebruik word. Dit is voorberei om deur nasieners gebruik te word. Daar word na alle nasieners verwag om 'n deeglike standaardiseringsvergadering by te woon om seker te maak dat die riglyne konsekwent geïnterpreteer en tydens die nasien van die kandidate se werk toegepas word.
- Let daarop dat kandidate wat 'n alternatiewe korrekte oplossing as wat as voorbeeld van 'n oplossing in die nasienriglyne gegee word verskaf, volle krediet vir die relevante oplossing moet kry tensy die spesifieke instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom is nie.
- **Bylaag A, B en C** (bladsy 3–8) sluit die nasienrubriek vir elke om te gebruik vir enigeen van die twee programmeringstale in.
- **Bylaag D, E en F** (bladsy 9–17) bevat voorbeelde in programmeringskode in Java van oplossings vir VRAAG 1 tot VRAAG 3.
- **Bylaag G, H en I** (bladsy 18–30) bevat voorbeelde in programmeringskode in Delphi van oplossings vir VRAAG 1 tot VRAAG 3.
- Kopieë van **Bylaag A, B en C** (bladsy 3–8) moet vir elke kandidaat gemaak word en tydens die nasiensessie voltooi word.

BYLAAG A:**AFDELING A:****VRAAG 1: NASIENRUBRIEK - ALGEMENE PROGRAMMERINGSVAARDIGHEDE**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT-PUNT
1.1	<p>Knoppie – [Vraag 1.1]</p> <p>Verkry ouderdom uit teksblokkie, skakel om na heelgetal ✓ As die paspoort-kontroleblokkie gemerk is ✓ As ouderdom < 16 ✓ en volwasse-kontroleblokkie is gemerk ✓ Vertoon 'Inteken is bevestig ' ✓ Anders Vertoon 'Inteken is nie bevestig nie' ✓ anders Vertoon 'Inteken is nie bevestig nie' ✓</p>	7	
1.2	<p>Knoppie – [Vraag 1.2]</p> <p>Verkry gewig van bagasie uit teksblokkie, skakel om na reële getal ✓ Verkry die reël met die lugredery en maksimum gewig van bagasie uit die lysblokkie ✓ // Verkry maksimum gewig uit die reël Kry die posisie van die hutskarakter Kry die getal karakters wat as gewig uit die reël gekopieer moet word deur van 'length' of enige funksie gebruik te maak ✓ Kopieer die gewig-gedeelte uit die reël ✓ Skakel gewig om na reële getal ✓</p> <p>As gewig > maksimum gewig ✓ Bereken oortollige gewig ✓ Bereken koste van oortollige gewig ✓</p> <p>Vertoon oortollige gewig ✓ Vertoon koste van oortollige gewig ✓</p>	10	
1.3	<p>Knoppie - [Vraag 1.3]</p> <p>Verkry getal passasiers uit teksblokkie, skakel om na heelgetal ✓ Bereken die getal vegetariese etes om voor te berei ✓ Bereken die getal nie-vegetariese etes om voor te berei ✓ Vertoon die getal vegetariese etes ✓ Vertoon die getal nie-vegetariese etes ✓</p>	5	

1.4	<p>Knoppie - [Vraag 1.4]</p> <p>Kontroleer of die lengte van die vertrektyd = 5 karakters) ✓ Kontroleer of die derde karakter in die string is ✓ As (lengte van vertrektyd = 5) EN (derde karakter 'h' is) Verkry uur uit toevoer as heelgetal ✓ Verkry minute uit toevoer as heelgetal ✓ As (uur <= 23) ✓ EN (minute <= 59) ✓ as minute >= 35 ✓ minute = minute - 35 ✓ anders uur = uur - 1 ✓ minute = 60 + minute - 35 ✓✓ As minute < 10 dan ✓ Intekentyd = uur + 'h' + '0' + minute ✓ anders Intekentyd = uur + 'h' + minute ✓ Vertoon die intekentyd ✓ Anders Vertoon boodskap 'Ongeldige tyd ingesleutel' } ✓ Maak vertrektyd se teksblokkie skoon } Anders Vertoon boodskap 'Ongeldige tyd ingesleutel' } ✓ Maak vertrektyd se teksblokkie skoon }</p>	17	
1.5	<p>Knoppie - [Vraag 1.5]</p> <p>Verkry afstand uit teksblokkie as reële waarde ✓ Toets indeks/item van radioknoppie wat geselekteer is ✓ Stel vlugkaartStatus op 'NieLidNie' Stel bonuspunte op 0</p> <p>As Silwer geselekteer is, Stel vlugkaartStatus op 'Silwer' ✓</p> <p>As Goud geselekteer is, Set vlugkaartStatus op 'Goud' Stel bonuspunte op 15% van afstand ✓</p> <p>As Platinum geselekteer is Stel vlugkaartStatus op 'Platinum' Stel bonuspunte op 20% van afstand ✓</p> <p>Stel lêernaam op vlugkaartStatus + '.jpeg' ✓ Laai die lêer in die beeldkomponent ✓</p> <p>As 'n kaart geselekteer was, Wys paneel met die naam pnlPunte ✓ Bereken punte Afstand / 1.6 ✓ + bonuspunte en rond na onder af ('down') ✓</p> <p>Vertoon punte op paneel ✓</p>	11	
TOTAAL:		50	

BYLAAG B:**AFDELING B:****VRAAG 2: NASIENRUBRIEK - OBJEK-GEÛRIENTEERDE PROGRAMMERING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT-PUNT
2.1.1	Konstruktor: Opskrif ✓ met drie parameters ✓ Regte datatipes ✓ Ken parameterwaardes aan drie attribute toe ✓ Roep die setGetalPassasiers-metode ✓	5	
2.1.2	Toegangsmetodes ('Accessors'): getVlugNommer ✓ met regte terugstuur-datatipe ✓ getGetalPassasiers ✓ met regte terugstuur-datatipe ✓	4	
2.1.3	vermeerderPassasiers-metode: Opskrif: vermeerderPassasiers ✓ Vermeerder die getalPassasiers-attribuut met 1 ✓	2	
2.1.4	berekenPersBespreek-metode: Opskrif: stuur reële data tipe terug, ✓ heelgetal parameter ✓ Bereken persentasie: Passasiers bespreek/maksimum passasiers * 100 ✓ stuur persentasie terug ✓	4	
2.1.5	toString-metode: Opskrif toString ✓ Metode se definisie ✓ Formaat ✓ en regte attribute ✓ (-1 vir elke verkeerde attribuut of metode wat geroep word – maksimum 2 punte) Regte terugstuur-stelling ✓	5	

(VRAAG 2.2 op die volgende bladsy.)

VRAAG 2: NASIENRUBRIEK – vervolg

2.2.1	<p>Knoppie – [2.2.1] Verkry vlugbesonderhede uit kombinasieblokkie ✓ Gebruik vlugbesonderhede om die volgende te doen: Bepaal die posisie van die hutskarakter ✓ Kopieer vlugnommer na 'n veranderlike ✓ Bepaal die posisie van die volgende hutskarakter ✓ Kopieer stad se naam na 'n veranderlike ✓ Kopieer datum na 'n veranderlike ✓✓ Instansieer die vlugobjek ✓ deur van die regte argumente gebruik te maak ✓ Maak die knoppie btnVraag222 aktief ✓ Maak die knoppie btnVraag223 aktief ✓ Vertoon 'n boodskap vir die instansiëring van die objek ✓</p>	12	
2.2.2	<p>Knoppie – [2.2.2] Roep die stelGetalPassasiers-metode van die objek ✓ <i>{Delphi: AssignFile, Reset Java: Skep objek om uit lêer te lees}</i> ✓✓ As lêer nie bestaan nie ✓ vertoon boodskap ✓, maak program toe ✓ Lus ('Loop'/stap) deur lêer ✓ Lees reël uit tekslêer ✓ Toets of vlugnommer van objek deel is van die reël ✓✓ Roep die vermeerderPassasiers-metode ✓ Voeg die reël by die afvoerarea van die GGK ✓ Eindig die lus Vertoon 'n blanko / leë reël ✓ Vertoon besonderhede van die objek deur van die toString-metode gebruik te maak ✓ Maak die lêer toe ✓</p>	15	
2.2.3	<p>Knoppie – [2.2.3] <i>{Delphi: AssignFile, Append Java: Skep 'n objek om na die lêer toe te skryf}</i> ✓✓ Gebruik dialoogblokkie om die getal passasiers in te sleutel ✓ Roep die berekenPersBespreek-metode ✓ As persentasie ≥ 100 vertoon boodskap 'Vol bespreek' ✓ anders Gebruik dialoogblokkie om persentasie bespreek te vertoon ✓ Gebruik dialoogblokkie om nuwe naam in te sleutel ✓ //Stel die passasier se verwysingsnommer saam Vlugnommer + '-' ✓+ (getalPassasiers + 1) ✓ Skryf die verwysingsnommer + ' ' + passasier se naam ✓ na die tekslêer ✓ Maak die lêer toe ✓ Roep die Vraag222-knoppie om die passasier se besonderhede en die vlugobjek te vertoon ✓</p>	13	
TOTAAL:		60	

BYLAAG C:**AFDELING C:****VRAAG 3: NASIENRUBRIEK – PROBLEEMOPLOSSING**

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT-PUNT
3.1	<p>Knoppie – [Vraag 3.1 – Vertoon rye] Gebruik die getal elemente in die arrPassasiers-skikking ✓ om die getal rye (kolomme) as volg te bepaal: Van 1 tot 9 passasiers: toonbank 1 ✓ Van 10 tot 16 passasiers: toonbank 2 ✓ Van 17 tot 24 passasiers: toonbank 3 Meer as 24 passasiers: toonbank 4 ✓ (4)</p> <p>Bepaal getal rye: Getal passasiers/getal toonbanke ✓ Opwaarts afgerond ✓ (2)</p> <p>Sorteer die passasiers in skikking volgens klas: Buitenste lus ✓; Binneste lus ✓; Regte toets ✓; Ruil elemente van skikking reg om ✓✓ (5)</p> <p>Kopieer inligting van passasiers na 2D/Grid: Inisialiseer teller om te toets of einde van skikking bereik is ✓ Lus deur rye ✓ (for) Inisialiseer teller na 0 ✓ Lus deur kolomme ✓ en toets of einde van skikking bereik is ✓ (while) Kopieer uit passasiers se skikking na 2D/Grid ✓ deur van die regte indekse gebruik te maak ✓ Dateer teller op ✓ (8)</p> <p>Vertoon passasiers in rye: Opskrifte: Lus regte aantal keer ✓ Vertoon opskrifte van kolomme ✓ Data: Lus deur rye ✓ Lus deur kolomme ✓ Vertoon passasier se klas ✓ en nommer ✓ In die regte getal kolomme en regte getal rye ✓ (7)</p>	26	

3.2	<p>Knoppie – [3.2 – Skep nuwe lys] Lees vlugnommer van verdraagde vlug uit kombinasieblokkie ✓ Vertoon opskrif wat vlugnommer bevat ✓ Lus deur skikking ✓ Toets of vlugnommer = verdraagde vlug se nommer ✓ Inkrementeer aantPassasiers op verdraagde vlug ✓ Vertoon by nuwe toonbank ✓ (6)</p> <p>Verwyder passasiers van verdraagde vlug uit rye Verminder die getal passasiers deur die getal passasiers op die verdraagde vlug af te trek van oorspronklike getal passasiers ✓ Skep tydelike skikking ✓ <i>(grootte van oorblywende passasiers)</i> Lus deur die passasiers-skikking ✓ As passasier nie op verdraagde vlug bespreek is nie ✓ Kopieer na tydelike skikking ✓ Skryf tydelike skikking oor oorspronklike passasiers-skikking ✓ Dateer rye by toonbanke op ✓ Vertoon opgedateerde afvoer ✓ (8)</p>	14	
TOTAAL		40	

OPSOMMING VAN KANDIDAAT SE PUNTE:

SENTRUMNOMMER:	EKSAMENNOMMER:			
	AFDELING A	AFDELING B	AFDELING C	
	VRAAG 1	VRAAG 2	VRAAG 3	GROOTTOTAAL
MAKS. PUNTE	50	60	40	150
KANDIDAAT SE PUNTE				

BYLAAG D: OPLOSSING VIR VRAAG 1: JAVA

```
//      'n Oplossing vir Vraag 1
//      Kode wat voorsien is
//      final double KOSTE_PER_KG = 50.0;
=====
// Vraag 1.1
=====
private void btnVraag11ActionPerformed(java.awt.event.ActionEvent evt)
{
    int ouderdom = Integer.parseInt(txfOuderdom.getText().trim());
    if (chbPaspoort.isSelected()) {
        if (ouderdom >= 16 || (ouderdom < 16 && chbMinderjarig.isSelected())) {
            txfVraag11.setText("Instap is bevestig.");
        } else {
            txfVraag11.setText("Instap is nie bevestig nie.");
        }
    } else {
        txfVraag11.setText("Instap is nie bevestig nie.");
    }
}
=====
// Vraag 1.2
=====
private void btnVraag12ActionPerformed(java.awt.event.ActionEvent evt) {

    double oortollig = 0;
    double koste = 0;
    double gewig = Double.parseDouble(txfGewig.getText());
    String lugrederyInfo = "" + lstLugrederyInfo.getSelectedValue();
    int posHash = lugrederyInfo.indexOf("#");
    double maksGewig = Double.parseDouble(lugrederyInfo.substring
        (posHash + 1, lugrederyInfo.length() - 2));

    if (gewig > maksGewig) {
        oortollig = gewig - maksGewig;
        koste = oortollig * KOSTE_PER_KG;
    }
    txaVraag12.setText("Oortollige gewig: " + String.format("%-4.2f", oortollig)
        + "kg" + "\n" + "Koste: " + String.format("R%-6.2f", koste));
}
=====
// Vraag 1.3
=====
private void btnVraag13ActionPerformed(java.awt.event.ActionEvent evt) {
    int aantPassasiers = 0;
    int aantVeg = 0;
    int aantNieVeg = 0;
    aantPassasiers = Integer.parseInt(txfAantPassasiers.getText());
    aantVeg = aantPassasiers / 3;
    aantNieVeg = aantPassasiers - aantVeg;
    txaVraag13.setText("Vegetariese etes: " + aantVeg);
    txaVraag13.append("\nNie-vegetariese etes: " + aantNieVeg);
}
}
```

```

=====
// Vraag 1.4
=====
private void btnVraag14ActionPerformed(java.awt.event.ActionEvent evt) {
    String instaptyd = "";
    boolean bLen = (txfVertrekTyd.getText().length() == 5);
    boolean bVlugTyd = (txfVertrekTyd.getText().charAt(2) == 'h');
    if (bLen && bVlugTyd) {
        int uur = Integer.parseInt(txfVertrekTyd.getText().substring(0, 2));
        int minute = Integer.parseInt(txfVertrekTyd.getText().substring(3, 5))
        if ((uur >= 0 && uur <= 23) && (minute <= 0 && minute <= 59)) {
            if (minute >= 35) {
                minute -= 35;
            }
            else {
                uur -= 1;
                minute = minute + 60 - 35;
            }
            if (uur < 10) {
                instaptyd = "0" + uur + "h";
            } else {
                instaptyd = uur + "h";
            }
            if (minute < 10) {
                instaptyd = instaptyd + "0" + minute;
            } else {
                instaptyd = instaptyd + minute;
            }
            txfVraag13.setText(instaptyd);
        }
        else {
            JOptionPane.showMessageDialog(null, "Ongeldige tyd is ingesleutel.");
            txfVertrekTyd.setText("");
        }
    }
    else {
        JOptionPane.showMessageDialog(null, "Ongeldige tyd is ingesleutel.");
        txfVertrekTyd.setText("");
    }
}
}

```

```

=====
// Vraag 1.5
=====
private void btnVraag15ActionPerformed(java.awt.event.ActionEvent evt) {
    double bonusPunte = 0;
    double punte = 0;
    String vlugkaartStatus = "NieLidNie";
    double afstand = Double.parseDouble(txfAfstand.getText());
    boolean isLid = false;
    if (rbtSilver.isSelected()) {
        isLid = true;
        vlugkaartStatus = "Silwer";
    }
    if (rbtGold.isSelected()) {
        isLid = true;
        vlugkaartStatus = "Goud";
        bonusPunte = afstand * 0.15;
    }
    if (rbtPlatinum.isSelected()) {
        isLid = true;
    }
}

```

```
        vlugkaartStatus = "Platinum";
        bonusPunte = afstand * 0.2;
    }
    String fileName = vlugkaartStatus + ".jpg";

    try {
        lblBeeld.setIcon(new
            javax.swing.ImageIcon(getClass().getResource(fileName)));
        } catch (Exception e) {
            System.out.println(e);
        }
        if (isLid) {
            pnlPunte.setVisible(true);
            punte = afstand / 1.6 + bonusPunte;
            lblPunte.setText("Punte verdien: " + (int)Math.floor(punte));
        }
    }
}
```

BYLAAG E: OPLOSSING VIR VRAAG 2: JAVA

```

//'\n Oplossing vir vraag 2
public class Vlug {

    // Kode wat voorsien is
    private String vlugNommer;
    private String stad;
    private String datum;
    private int getalPassasiers;
    =====
    // Vraag 2.1.1
    =====
    public Vlug(String vlugNommer, String stad, String datum) {
        this.vlugNommer = vlugNommer;
        this.stad = stad;
        this.datum = datum;
        setGetalPassasiers();
    }
    =====
    // Vraag 2.1.2
    =====
    public String getVlugNommer() {
        return vlugNommer;
    }
    public int getGetalPassasiers() {
        return getalPassasiers;
    }
    =====
    // Kode wat voorsien is
    =====
    public void setGetalPassasiers () {
        this. getalPassasiers = 0;
    }
    =====
    // Vraag 2.1.3
    =====
    public void vermeerderPassasiers()
    {
        getalPassasiers++;
    }
    =====
    // Vraag 2.1.4
    =====
    public double berekenPersBespreek(double maksPassasiers)
    {
        double pers = getalPassasiers/maksPassasiers * 100;
        return pers;
    }
    =====
    // Vraag 2.1.5
    =====
    public String toString()
    {
        return "Vlugnommer: " + vlugNommer + "\nBestemming: " + stad +
            "\nVertrekdatum: " + datum + "\nGetal passasiers bespreek: " +
            getalPassasiers ;
    }
}

```

GGK-KLAS: VRAAG2_OPLOSSING

```

package Vraag2Package;

import java.io.File;
import java.io.FileReader;
import java.text.DecimalFormat;
import java.util.Scanner;
import javax.swing.JOptionPane;

public class Vraag2_Memo extends javax.swing.JFrame {
=====
// Kode wat voorsien is
=====
Vlug objVlug = null;

public Question2_Memo() {
    initComponents();
    this.setLocationRelativeTo(this);
    this.setVisible(true);
    btnVraag2_2_2.setEnabled(false);
    btnVraag2_2_3.setEnabled(false);    }
=====
// Vraag 2.2.1
=====
private void btnVraag221ActionPerformed(java.awt.event.ActionEvent evt) {
    String reel = "" + cmbVlugInfo.getSelectedItemAt();
    String[] temp = reel.split("#");
    objVlug = new Vlug(temp[0], temp[1], temp[2]);
    btnVraag222.setEnabled(true);
    btnVraag223.setEnabled(true);
    JOptionPane.showMessageDialog(rootPane, "Vlugobjek is
        geinstansieer.");
    }
=====
// Vraag 2.2.2
=====
private void btnVraag222ActionPerformed(java.awt.event.ActionEvent evt) {
    objVlug.setGetalPassasiers();
    try
    {
        Scanner scLeer = new Scanner(new FileReader("DataV2.txt"));
        txaAfvoer.setText("Lys van passasiers: \n");
        while (scLeer.hasNext())
        {
            String reel = scLeer.nextLine();
            if(reel.contains(objVlug.getVlugNommer()))
            {
                txaAfvoer.append(reel + "\n");
                objVlug.vermeerderPassasiers();
            }
        }
        txaAfvoer.append("\n" + objVlug.toString());
    }
    catch (FileNotFoundException e)
    {
        JOptionPane.showMessageDialog(rootPane, "Leer bestaan nie.");
        System.exit(0);
    }
}
}

```

```
=====
// Vraag 2.2.3
=====
private void btnVraag223ActionPerformed(java.awt.event.ActionEvent evt) {
    double maksPassasiers =
        Double.parseDouble(JOptionPane.showInputDialog(null, "Sleutel getal
            passasiers in"));
    double persentasie = objVlug.berekenPersBespreek(maksPassasiers);
    if (persentasie >= 100)
        JOptionPane.showMessageDialog(rootPane, "Vol bespreek");
    else
    {
        JOptionPane.showMessageDialog(rootPane, "Persentasie bespreek: " +
            String.format("%.1f",persentasie) + "%");
        String naam = JOptionPane.showInputDialog(null, "Sleutel die naam van die
            nuwe passasier in");
        int getalPassasiers = objVlug.getGetalPassasiers()+ 1;
        String afvoer = objVlug.getVlugNommer() + '-' + getalPassasiers + "    " +
            naam;

        try
        {
            PrintWriter pw = new PrintWriter(new FileWriter("DataV2.txt",true));
            pw.println(afvoer);
            pw.close();
        }
        catch (Exception e)
        {
            JOptionPane.showMessageDialog(rootPane, "Fout met skryf na leer");
        }
        btnVraag222.doClick();
    }
}
=====
```

BYLAAG F: OPLOSSING VIR VRAAG 3: JAVA**KLAS OM SKIKKINGS OP TE VUL: Vul Skikkings**

```
// Kode wat voorsien is

package Vraag3_Package;

public class VulSkikkings {

    static String[] arrPosPassasiers = {"E01;TDB2506", "E02;TDB1305",
    "E03;TDB1305", "E04;TDB2506", "E05;TDB2506", "B06;TDB4310", "E07;TDB4310",
    "B08;TDB1305", "E09;TDB4310", "B10;TDB2506", "E11;TDB1305", "B12;TDB4310",
    "B13;TDB2506", "B14;TDB4310", "E15;TDB2506", "E16;TDB1305", "E17;TDB2506",
    "E18;TDB1305", "E19;TDB4310", "E20;TDB4310", "E21;TDB1305", "B22;TDB1305",
    "B23;TDB2506", "E24;TDB4310", "E25;TDB1305", "E26;TDB4310", "B27;TDB1305",
    "B28;TDB1305", "E29;TDB4310", "E30;TDB2506", "B31;TDB1305", "E32;TDB2506",
    "E33;TDB2506", "E34;TDB1305", "B35;TDB1305"};

    public static String[] VulEwekansig() {
        int grootte = (int) (Math.random() * (35) + 1);
        String[] arrPassasiers;
        arrPassasiers = new String[grootte];
        for (int teller = 0; teller < grootte; teller++) {
            arrPassasiers[teller] = arrPosPassasiers[teller];
        }
        return arrPassasiers;
    }

    public static String[][] inisialiseer2D() {
        String[][] arrGrid = new String[9][4];
        for (int ry = 0; ry < 9; ry++) {
            for (int kolom = 0; kolom < 4; kolom++) {
                arrGrid[ry][kolom] = "";
            }
        }
        return arrGrid;
    }
}
```

KLAS: Vraag3_Memo

```

    int aantToonbanke = 0;
    String verdraag = "";
    int rye = 0;
    int kolomme = 0;
    int aantPassasiers = 0;
//=====
//Kode wat voorsien is
//=====

// Globale skikkings
    String[] arrPassasiers = VulSkikkings.VulEwekansig();
    String[][] arrGrid = VulSkikkings.inisialiseer2D();

public Vraag3_Memo() {
    initComponents();
}
//=====
// Vraag 3.1
//=====
    private void btnVertoonActionPerformed(java.awt.event.ActionEvent evt) {
        aantPassasiers = arrPassasiers.length;
        if (aantPassasiers >= 1 && aantPassasiers < 10) {
            aantToonbanke = 1;
        }
        if (aantPassasiers >= 10 && aantPassasiers <= 16) {
            aantToonbanke = 2;
        }
        if (aantPassasiers > 16 && aantPassasiers < 25) {
            aantToonbanke = 3;
        }
        if (aantPassasiers >= 25) {
            aantToonbanke = 4;
        }
        rye = (int) Math.ceil(aantPassasiers / (double) aantToonbanke);
        kolomme = aantToonbanke;
        plaasInRye();
        vertoon();
    }

    public void plaasInRye() {
        for (int buite = 0; buite < aantPassasiers - 1; buite++) {
            for (int binne = buite + 1; binne < aantPassasiers; binne++) {
                if (arrPassasiers[binne].compareTo(arrPassasiers[buite]) < 0) {
                    String temp = arrPassasiers[binne];
                    arrPassasiers[binne] = arrPassasiers[buite];
                    arrPassasiers[buite] = temp;
                }
            }
        }
        int toonbank = 0;
        for (int ry = 0; ry < rye; ry++) {
            int kolom = 0;
            while (kolom < kolomme && toonbank < aantPassasiers) {
                arrGrid[ry][kolom] = arrPassasiers[toonbank];
                kolom++;
                toonbank++;
            }
        }
    }
}

```

```

public void vertoon() {
    txtAfvoer.setText("");
    for (int teller = 1; teller <= kolomme; teller++) {
        txtAfvoer.append(String.format("%-12s", "Toonbank " + teller));
    }
    txtAfvoer.append("\n");
    int toonbank = 0;
    for (int ry = 0; ry < rye; ry++) {
        int kolom = 0;
        while (kolom < kolomme && toonbank < aantPassasiers) {
            String[] line = arrGrid[ry][kolom].split(";");
            txtAfvoer.append(String.format("%-12s", line[0]));
            kolom++;
            toonbank++;
        }
        txtAfvoer.append("\n");
    }
}

=====
// Vraag 3.2
=====
private void cmbVertraagActionPerformed(java.awt.event.ActionEvent evt) {
    vertraag = cmbVertraag.getSelectedItem() + "";
}

private void btnNuweLysActionPerformed(java.awt.event.ActionEvent evt) {
    int hoeveel = 0;
    txtVertraag.setText("Vlugnommer: " + vertraag + "\n");
    for (int teller = 0; teller < aantPassasiers; teller++) {
        if (arrPassasiers[teller].contains(vertraag)) {
            hoeveel++;
            String[] temp = arrPassasiers[teller].split(";");
            txtVertraag.append(temp[0] + "\n");
        }
    }
    aantPassasiers = aantPassasiers - hoeveel;
    dateerOp();
}

public void dateerOp() {
    String[] temp = new String[aantPassasiers];
    int toonbank = 0;
    for (int teller = 0; teller < arrPassasiers.length; teller++) {
        if (!arrPassasiers[teller].contains(vertraag)) {
            temp[toonbank] = arrPassasiers[teller];
            toonbank++;
        }
    }
    arrPassasiers = temp;
    plaasInRye();
    vertoon();
}

```

BYLAAG G: OPLOSSING VIR VRAAG 1: DELPHI

```
unit Vraag1_U;

interface

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

type
  TfrmVraag1 = class(TForm)
    Panel1: TPanel;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    edtOud: TEdit;
    GroupBox2: TGroupBox;
    chbMinderjarig: TCheckBox;
    btnVraag11: TButton;
    edtVraag11: TEdit;
    chbPaspoort: TCheckBox;
    lstMaksGewig: TListBox;
    edtGewig: TEdit;
    Label2: TLabel;
    btnVraag12: TButton;
    GroupBox3: TGroupBox;
    edtVertrektyd: TEdit;
    Label3: TLabel;
    btnVraag14: TButton;
    edtVraag14: TEdit;
    GroupBox4: TGroupBox;
    btnVraag15: TButton;
    redVraag12: TRichEdit;
    Label6: TLabel;
    Label8: TLabel;
    GroupBox5: TGroupBox;
    Label4: TLabel;
    edtAantPassasiers: TEdit;
    btnVraag13: TButton;
    redVraag13: TRichEdit;
    Label5: TLabel;
    edtAfstand: TEdit;
    rgpVlugKaart: TRadioGroup;
    pnlPunte: TPanel;
    imgVlugKaart: TImage;
    procedure btnVraag11Click(Sender: TObject);
    procedure btnVraag12Click(Sender: TObject);
    procedure btnVraag14Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure btnVraag13Click(Sender: TObject);
    procedure btnVraag15Click(Sender: TObject);

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

const
  rKostePerKg = 50; // Vir gebruik in Vraag 1.2
```

```

var
  frmVraag1: TfrmVraag1;
  iKode: integer = -1; // stel indeks van radioknoppie
implementation
{$R *.dfm}
=====
// Vraag 1.1
=====
procedure TfrmVraag1.btnVraag11Click(Sender: TObject);
var
  sBoodskap: string;
  iOuderdom: integer;
begin
  iOuderdom := StrToInt(edtOud.Text);
  if (chbPaspoort.Checked) then
    begin
      if (iOuderdom >= 16) or ((iOuderdom < 16) AND chbMinderjarig.Checked) then
        edtVraag11.Text := 'Instap is bevestig.'
      else
        edtVraag11.Text := 'Instap is nie bevestig nie.';
      end
    else
      edtVraag11.Text := 'Instap is nie bevestig nie.';
    end;
=====
// Vraag 1.2
=====
procedure TfrmVraag1.btnVraag12Click(Sender: TObject);
var
  rGewig, rMaksGewig, rOortollig, rKostePKg: real;
  sLugredery: string;
  iPosHash: integer;
begin
  rOortollig := 0;
  rKostePKg := 0;
  rGewig := StrToFloat(edtGewig.Text);
  sLugredery := lstMaksGewig.Items[lstMaksGewig.ItemIndex];
  iPosHash := pos('#', sLugredery);
  Delete(sLugredery, 1, iPosHash);
  rMaksGewig := StrToFloat(Copy(sLugredery, 1, length(sLugredery) - 2));
  if rGewig > rMaksGewig then
    begin
      rOortollig := rGewig - rMaksGewig;
      rKostePKg := rOortollig * rKostePerKg;
    end;
  redVraag12.Text := 'Oortollige gewig: ' + FloatToStrF(rOortollig, ffFixed, 5, 0)
    + 'kg' + #13 + 'Koste: ' + FloatToStrF(rKostePKg, ffCurrency, 0, 2);
end;
=====
// Vraag 1.3
=====
procedure TfrmVraag1.btnVraag13Click(Sender: TObject);
var
  iAantPassasiers, iAantVeg, iAantNieVeg: integer;
begin
  iAantPassasiers := StrToInt(edtAantPassasiers.Text);
  iAantVeg := iAantPassasiers div 3;
  iAantNieVeg := iAantPassasiers - iAantVeg;
  redVraag13.Clear;
  redVraag13.Lines.Add('Vegetariese etes: ' + IntToStr(iAantVeg));
  redVraag13.Lines.Add('Nie-vegetariese etes: ' + IntToStr(iAantNieVeg));
end;
=====

```

```
// Vraag 1.4
=====
procedure TfrmVraag1.btnVraag14Click(Sender: TObject);
var
  bLen, bVlugTyd : Boolean;
  sVertrekTyd, sInstapTyd: string;
  iUur, iMinute: integer;
begin
  sInstapTyd := '';
  bLen := length(edtVertrekTyd.Text) = 5;
  bVlugTyd := edtVertrekTyd.Text[3] = 'h'
  if (bLen AND bVlugTyd) then
    begin
      iUur := StrToInt(Copy(edtVertrekTyd.Text, 1, 2));
      iMinute := StrToInt(Copy(edtVertrekTyd.Text, 4, 2));

      if (iUur <= 23) AND (iMinute <= 60) then
        begin
          if (iMinute >= 35) then
            iMinute := iMinute - 35
          else
            begin
              iUur := iUur - 1;
              iMinute := 60 + iMinute - 35;
            end;
          if iUur < 10 then
            sInstapTyd := '0' + IntToStr(iUur)
          else
            sInstapTyd := IntToStr(iUur);

          if iMinute < 10 then
            sInstapTyd := sInstapTyd + 'h' + '0' + IntToStr(iMinute)
          else
            sInstapTyd := sInstapTyd + 'h' + IntToStr(iMinute);
          edtVraag14.Text := sInstapTyd;
        end; // as uur en minuut geldig
      end // as formaat ongeldig
    else
      begin
        ShowMessage('Ongeldige tyd is ingesleutel.');
```

```
iKode := rgpVlugKaart.ItemIndex;
rBonusPunte := 0;
case iKode of
  0:
    begin
      sVlugkaartStatus := 'Silwer';
    end;
  1:
    begin
      sVlugkaartStatus := 'Goud';
      rBonusPunte := rAfstand * 0.15;

    end;
  2:
    Begin

      sVlugkaartStatus := 'PlatinumA';
      rBonusPunte := rAfstand * 0.20;
    end;
end;
sLeernaam := sVlugkaartStatus + '.bmp';
if FileExists(sLeernaam) then
  imgVlugKaart.Picture.LoadFromFile(sLeernaam);
if iKode >= 0 then
  begin
    pnlPunte.Show;
    rPunte := Floor(rAfstand / 1.6 + rBonusPunte);
    pnlPunte.Caption := 'Punte verdien: ' + FloatToStr(rPunte);
  end;
end;

procedure TfrmVraag1.FormActivate(Sender: TObject);
begin
  currencyString := 'R';
  pnlPunte.Hide;
end;

end.
```

BYLAAG H: OPLOSSING VIR VRAAG 2: DELPHI

```

unit Vlug_U;

interface

uses SysUtils, Math;

type
  TVlug = class
  private
    fVlugNommer : String;
    fStad : String;
    fDatum : String;
    fGetalPassasiers : integer;

  public
    constructor create(sVlugNommer:String; sStad: String; sDatum: String);
    function getVlugNommer:String;
    function getGetalPassasiers:integer;
    procedure setGetalPassasiers;
    procedure vermeerderPassasiers;
    function berekenPersBespreek(iMaks :integer):double;
    function toString: String;

  end;

implementation

{ TVlug }

=====
// Vraag 2.1.1
=====
constructor TVlug.create(sVlugNommer:String; sStad: String; sDatum: String);
begin
  fVlugNommer := sVlugNommer;
  fStad := sStad;
  fDatum := sDatum;
  setGetalPassasiers;
end;

=====
// Vraag 2.1.2
=====
function TVlug.getVlugNommer: String;
begin
  Result := fVlugNommer;
end;

function TVlug.getGetalPassasiers: integer;
begin
  Result := fGetalPassasiers;
end;

=====
// Kode wat voorsien is
=====
procedure TVlug.setGetalPassasiers;
begin
  fGetalPassasiers := 0;
end;

```

```
=====
// Vraag 2.1.3
=====
procedure TVlug.vermeerderPassasiers;
begin
    Inc(fGetalPassasiers);
end;
=====
// Vraag 2.1.4
=====
function TVlug.berekenPersBespreek(iMaks :integer):double;
begin
    result := fGetalPassasiers/iMaks*100;
end;
=====
// Vraag 2.1.5
=====
function TVlug.toString: string;
begin
    result := 'Vlugnommer: ' + fVlugNommer + #13 + 'Bestemming: ' + fStad +
        #13 + 'Vertrekdatum: ' + fDatum + #13 + 'Getal passasiers bespreek: ' +
            IntToStr(fGetalPassasiers);
end;
end.
```

HOOFVORMEENHEID: VRAAG2_U.PAS

```

unit Vraag2_U;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Vlug_U, ComCtrls, ExtCtrls;
type
  TfrmVlug = class(TForm)
    cmbVlugInfo: TComboBox;
    btnVraag221: TButton;
    redAfvoer: TRichEdit;
    btnVraag222: TButton;
    btnVraag223: TButton;
    edtNaam: TEdit;
    GroupBox1: TGroupBox;
    Label1: TLabel;
    procedure btnVraag222Click(Sender: TObject);
    procedure btnVraag221Click(Sender: TObject);
    procedure btnVraag223Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  frmVlug: TfrmVlug;
  objVlug: TVlug;

implementation
{$R *.dfm}
=====
// Vraag 2.2.1
=====

procedure TfrmVlug.btnVraag221Click(Sender: TObject);
var
  sVlugInfo, sVlugNommer, sStad, sDatum: String;
  iPos: integer;
begin
  redAfvoer.Lines.Clear;
  sVlugInfo := cmbVlugInfo.Items[cmbVlugInfo.ItemIndex];
  iPos := pos('#', sVlugInfo);
  sVlugNommer := copy(sVlugInfo, 1, iPos - 1);
  Delete(sVlugInfo, 1, iPos);

  iPos := pos('#', sVlugInfo);
  sStad := copy(sVlugInfo, 1, iPos - 1);
  Delete(sVlugInfo, 1, iPos);

  sDatum := sVlugInfo;

  objVlug := TVlug.create(sVlugNommer, sStad, sDatum);
  btnVraag222.Enabled:=true;
  btnVraag223.Enabled:=true;
  ShowMessage('Vlug-objek is geinstansieer.');
```

```

=====
// Vraag 2.2.2
=====
procedure TfrmVlug.btnVraag222Click(Sender: TObject);
var
  myLeer: TextFile;
  eenReel: string;
begin
  redAfvoer.Lines.Clear;
  objVlug.setGetalPassasiers;
  redAfvoer.Lines.Add('Lys van passasiers');
  AssignFile(myLeer, 'DataV2.txt');
  if not FileExists('DataV2.txt') then
  begin
    ShowMessage('The file does not exist');
    exit;
  end
  else
  begin
    reset(myLeer);
    while not eof(myLeer) do
    begin
      readln(myLeer, eenReel);
      if pos(objVlug.getVlugNommer, eenReel) > 0 then
      begin
        objVlug.vermeerderPassasiers;
        redAfvoer.Lines.Add(eenReel);
      end;
    end;
    redAfvoer.Lines.Add(#13 + objVlug.toString);
  end;
  CloseFile(myLeer);
end;
=====
// Vraag 2.2.3
=====
procedure TfrmVlug.btnVraag223Click(Sender: TObject);
var
  sNaam, sVerwysing: String;
  myLeer: TextFile;
  iMaksPassasiers, iGetalPassasiers: integer;
  iPersentasie : real;
begin
  iMaksPassasiers := StrToInt(Inputbox('Toevoer', 'Sleutel die maksimum getal
passasiers in', ''));
  iPersentasie := objVlug.berekenPersBespreek(iMaksPassasiers);
  if (iPersentasie >= 100) then
    ShowMessage('Vol bespreek')
  else
  begin
    ShowMessage('Persentasie bespreek: ' + FloatToStrF(iPersentasie,
      ffFixed,0,1) + '%');
    sNaam := Inputbox('', 'Naam van nuwe passasier: ', '');
    AssignFile(myLeer, 'DataV2.txt');
    Append(myLeer);
    sVerwysing := objVlug.getVlugNommer + '-' +
      IntToStr(objVlug.getGetalPassasiers + 1);
    writeln(myLeer, sVerwysing + ' ' + sNaam);
    CloseFile(myLeer);
    btnVraag222.Click;
  end;
end;
procedure TfrmVlug.FormActivate(Sender: TObject);

```

```
begin  
  btnVraag222.Enabled:=false;  
  btnVraag223.Enabled:=false;  
end;  
end.
```

BYLAAG I: OPLOSSING VIR VRAAG 3: DELPHI

```

unit Vraag3_U;

interface

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

type

  TfrmVraag3 = class(TForm)
    Panell1: TPanel;
    gpbToonbanke: TGroupBox;
    gpbVertraag: TGroupBox;
    btnVertoonRye: TButton;
    btnNuweLys: TButton;
    cbbVlugNommer: TComboBox;
    Labell1: TLabel;
    redVertraag: TRichEdit;
    stgToonbanke: TStringGrid;
    gpbVlugVertraag: TGroupBox;
    procedure FormActivate(Sender: TObject);
    procedure btnVertoonRyeClick(Sender: TObject);
    procedure btnNuweLysClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag3: TfrmVraag3;
  arrPosPassasiers, arrPassasiers, arrTemp: array[1..35] of string;
  iToonbanke, iGrootte, iRy, iKolom: integer;
  arrGrid: array[1..9, 1..4] of string;
  iGetalRye, iGetal: integer;

implementation

{$R *.dfm}
// Sorteër die skikking
Procedure Sorteër;
var
  i: Integer;
  j: Integer;
  sTemp : string;
begin
  for i := 1 to iGrootte - 1 do
    for j := 1 + i to iGrootte do
      if arrPassasiers[i] > arrPassasiers[j] then
        begin
          sTemp:=arrPassasiers[i];
          arrPassasiers[i] := arrPassasiers[j];
          arrPassasiers[j] := sTemp;
        end;
    end;
end;

```

```

// Kode wat voorsien is
procedure MaakSkikkingSkoop;
begin
  for iRy := 1 to 9 do
    for iKolom := 1 to 4 do
      begin
        arrGrid[iRy,iKolom]:= '';
        frmVraag3.stgToonbanke.Cells[iKolom,iRy]:= '';
      end;
    end;
  //=====
procedure VertoonInfo;
begin
  //Vul 2D-skikking en vertoon inligting in string grid
  iGetalRye := ceil(iGrootte/iToonbanke);
  iGetal := 0;
  iRy := 1;
  while iGetal < iGrootte do
    begin
      for iKolom := 1 to iToonbanke do
        begin
          inc(iGetal);
          arrGrid[iRy,iKolom]:=
copy(arrPassasiers[iGetal],1,pos(';',arrPassasiers[iGetal])-1);
        end;
        inc(iRy);
      end;
    for iRy := 1 to iGetalRye do
      for iKolom := 1 to iToonbanke do
        frmVraag3.stgToonbanke.Cells[iKolom,iRy] := arrGrid[iRy,iKolom];
      end;
    //=====
    //Vraag 3.1
    //=====
procedure TfrmVraag3.btnVertoonRyeClick(Sender: TObject);
var
  i: Integer;
begin
  if iGrootte <= 10 then
    iToonbanke := 1
  else
    if iGrootte <= 16 then
      iToonbanke:=2
    else
      if iGrootte <= 24 then
        iToonbanke:= 3
      else
        iToonbanke:= 4;

  for i := 1 to iToonbanke do
    stgToonbanke.Cells[i,0]:= 'Toonbank ' + IntToStr(i);
  Sorteer;
  VertoonInfo;
end;

```

```
=====
//Vraag 3.2
=====
procedure TfrmVraag3.btnNuweLysClick(Sender: TObject);
var
  i, j, iGetal, iNuweToonbank : Integer;
  sVlugNommer : string;
begin
  redVertraag.Clear;
  iNuweToonbank:=0;
  sVlugNommer:=cbbVlugNommer.Items[cbbVlugNommer.itemIndex];
  redVertraag.Lines.Add('Vlugnommer:' + sVlugNommer);
  for i := 1 to iGrootte do
    begin
      if pos(sVlugNommer,arrPassasiers[i]) > 0 then
        begin
          redVertraag.Lines.Add(copy(arrPassasiers[i],1,
            pos(';',arrPassasiers[i])-1));
        end
      else
        begin
          inc(iNuweToonbank);
          arrTemp[iNuweToonbank]:= arrPassasiers[i];
        end;
    end;
  end;

  for j := 1 to 35 do
    arrPassasiers[j]:= '';
  iGrootte := iNuweToonbank;
  arrPassasiers:=arrTemp;
  MaakSkikkingSkoon;
  Sorteer;
  VertoonInfo;
end;

// Kode wat voorsien is
=====
procedure TfrmVraag3.FormActivate(Sender: TObject);
var
  a:integer;
begin
  arrPosPassasiers[1] := 'E01;TDB2506';
  arrPosPassasiers[2] := 'E02;TDB1305';
  arrPosPassasiers[3] := 'E03;TDB1305';
  arrPosPassasiers[4] := 'E04;TDB2506';
  arrPosPassasiers[5] := 'E05;TDB2506';
  arrPosPassasiers[6] := 'B06;TDB4310';
  arrPosPassasiers[7] := 'E07;TDB4310';
  arrPosPassasiers[8] := 'B08;TDB1305';
  arrPosPassasiers[9] := 'E09;TDB4310';
  arrPosPassasiers[10] := 'B10;TDB2506';
  arrPosPassasiers[11] := 'E11;TDB1305';
  arrPosPassasiers[12] := 'B12;TDB4310';
  arrPosPassasiers[13] := 'B13;TDB2506';
  arrPosPassasiers[14] := 'B14;TDB4310';
  arrPosPassasiers[15] := 'E15;TDB2506';
  arrPosPassasiers[16] := 'E16;TDB1305';
  arrPosPassasiers[17] := 'E17;TDB2506';
  arrPosPassasiers[18] := 'E18;TDB1305';
end;
```

```
arrPosPassasiers[19] := 'E19;TDB4310';  
arrPosPassasiers[20] := 'E20;TDB4310';  
arrPosPassasiers[21] := 'E21;TDB1305';  
arrPosPassasiers[22] := 'B22;TDB1305';  
arrPosPassasiers[23] := 'B23;TDB2506';  
arrPosPassasiers[24] := 'E24;TDB4310';  
arrPosPassasiers[25] := 'E25;TDB1305';  
arrPosPassasiers[26] := 'E26;TDB4310';  
arrPosPassasiers[27] := 'B27;TDB1305';  
arrPosPassasiers[28] := 'B28;TDB1305';  
arrPosPassasiers[29] := 'E29;TDB4310';  
arrPosPassasiers[30] := 'E30;TDB2506';  
arrPosPassasiers[31] := 'B31;TDB1305';  
arrPosPassasiers[32] := 'E32;TDB2506';  
arrPosPassasiers[33] := 'E33;TDB2506';  
arrPosPassasiers[34] := 'E34;TDB1305';  
arrPosPassasiers[35] := 'B35;TDB1305';
```

```
MaakSkikkingSkoon;
```

```
iGrootte:= random(35)+1;
```

```
for a := 1 to iGrootte do  
  arrPassasiers[a]:=arrPosPassasiers[a];
```

```
end;
```

```
end.
```