



# 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><b>Knoppie – [Vraag 1.1]</b></p> <p>Verkry ouderdom uit teksblokkie, skakel om na heelgetal ✓            As die paspoort-kontroleblokkie gemerk is ✓                As ouderdom &lt; 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><b>Knoppie – [Vraag 1.2]</b></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 &gt; 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><b>Knoppie - [Vraag 1.3]</b></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><b>Knoppie - [Vraag 1.4]</b></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 &lt;= 23) ✓ EN (minute &lt;= 59) ✓              as minute &gt;= 35 ✓                minute = minute - 35 ✓            anders              uur = uur - 1 ✓              minute = 60 + minute - 35 ✓✓          As minute &lt; 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><b>Knoppie - [Vraag 1.5]</b></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	
<b>TOTAAL:</b>		<b>50</b>	

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT-PUNT
2.1.1	<b>Konstruktor:</b> Opskrif ✓ met drie parameters ✓ Regte datatipes ✓ Ken parameterwaardes aan drie attribute toe ✓ Roep die setGetalPassasiers-metode ✓	5	
2.1.2	<b>Toegangsmetodes ('Accessors'):</b> getVlugNommer ✓ met regte terugstuur-datatype ✓ getGetalPassasiers ✓ met regte terugstuur-datatype ✓	4	
2.1.3	<b>vermeerderPassasiers-metode:</b> Opskrif: vermeerderPassasiers ✓ Vermeerder die getalPassasiers-attribuut met 1 ✓	2	
2.1.4	<b>berekenPersBespreek-metode:</b> Opskrif: stuur reële data tipe terug, ✓ heelgetal parameter ✓ Bereken persentasie: Passasiers bespreek/maksimum passasiers * 100 ✓ stuur persentasie terug ✓	4	
2.1.5	<b>toString-metode:</b> 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><b>Knoppie – [2.2.1]</b>          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><b>Knoppie – [2.2.2]</b>          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><b>Knoppie – [2.2.3]</b>  <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 <math>\geq 100</math> 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	
<b>TOTAAL:</b>		<b>60</b>	

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	KANDIDAAT-PUNT
3.1	<p><b>Knoppie – [Vraag 3.1 – Vertoon rye]</b>            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><b>Bepaal getal rye:</b>            Getal passasiers/getal toonbanke ✓            Opwaarts afgerond ✓ (2)</p> <p><b>Sorteer die passasiers in skikking volgens klas:</b>            Buitenste lus ✓; Binneste lus ✓; Regte toets ✓; Ruil            elemente van skikking reg om ✓✓ (5)</p> <p><b>Kopieer inligting van passasiers na 2D/Grid:</b>            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><b>Vertoon passasiers in rye:</b>            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><b>Knoppie – [3.2 – Skep nuwe lys]</b>          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><b>Verwyder passasiers van verdraagde vlug uit rye</b>          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	
<b>TOTAAL</b>		<b>40</b>	

**OPSOMMING VAN KANDIDAAT SE PUNTE:**

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



**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.getSelectedItem();
    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.');
```

```
      edtVertrekTyd.Text := '';
    end;
  end
else
  begin
    ShowMessage('Ongeldige tyd is ingesleutel.');
```

```
    edtVertrekTyd.Text := '';
  end;
end;
=====
```

```
// Vraag 1.5
```

```
=====
```

```
procedure TfrmVraag1.btnVraag15Click(Sender: TObject);
```

```
var
```

```
  rAfstand, rBonusPunte, rPunte: real;
```

```
  sVlugkaartStatus, sLeernaam: string;
```

```
  I: integer;
```

```
begin
```

```
  sVlugkaartStatus := 'NieLidNie';
```

```
  rAfstand := StrToFloat(edtAfstand.Text);
```

```
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';

```

```
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.
```