



basic education

Department:
Basic Education
REPUBLIC OF SOUTH AFRICA

NASIONALE SENIOR SERTIFIKAAT

GRAAD 12

INLIGTINGSTEGNOLOGIE V1

NOVEMBER 2014

MEMORANDUM

PUNTE: 150

Hierdie memorandum bestaan uit 28 bladsye.

ALGEMENE INLIGTING:

- Hierdie nasienriglyne moet as basis tydens die nasiensessie gebruik word. Dit is geskep om deur die nasieners gebruik te word. Daar word van alle nasieners vereis om 'n standaardiseringsvergadering by te woon om te verseker dat die riglyne konsekwent geïnterpreteer en toegepas word tydens die nasien van leerders se werk.
- Let daarop dat leerders wat alternatiewe korrekte oplossings voorsien as dié wat as voorbeelde van oplossings in hierdie nasienriglyne gegee is, volle krediet vir die relevante antwoord sal ontvang tensy die instruksies in die vraestel nie gevolg is nie of die vereistes van die vraag nie nagekom word nie.
- **Bylae A, B en C** (bladsy 3–8) bevat die nasienrubriek vir elke vraag ongeag watter een van die twee programmeringstale gebruik is.
- **Bylae D, E en F** (bladsy 9–16) bevat voorbeelde van oplossings in programmeringskode vir Java vir Vraag 1 tot 3.
- **Bylae G, H en I** (bladsy 17–28) bevat voorbeelde van oplossings in programmeringskode vir Delphi vir Vraag 1 tot 3.
- Kopieë van **Bylae A, B en C** (bladsy 3–8) moet vir elke leerder gemaak word en tydens die nasiensessie voltooi word.

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
	<i>Indien leerder probleem het om te lees van kombinasie lys, penaliseer slegs een keer</i>		
1.1	Knoppie - [Bevestig aflewering] Verkry vertrek vanuit kombinasie lys (om te gebruik as teks) EN Verkry bestemming vanuit kombinasie lys (om te gebruik as teks) ✓ Verkry aantal kilometers vanuit teksblokkie ✓ skakel om na getal ✓ (kan ook omgeskakel word in 1.2) Skep string wat vertrekpunt, bestemming en afstand ✓ saamvoeg en toeken aan die byskrifkomponent wat voorsien is ✓	5	
1.2	Knoppie - [Afleringskoste] Verkry die opsie gekies vanuit die lysblokkie ✓ Korrekte veranderlike gebruik in toets ✓ All 4 moontlikhede (A1..A4 OF itemindeks 0..3) ✓ Korrekte seleksiestruktuur struktuur (if/case/switch) ✓ Gebruik korrekte tarief vir elke opsie ✓ Bereken koste: tarief x afstand ✓ Toets of spoedpos geselekteer is ✓, tel 100 by koste ✓ Ken koste toe aan gegewe byskrifkomponent ✓ met formatering, 1 of 2 desimale plekke ✓ (Aanvaar oplossings waar spasies vooraf nie in ag geneem is nie)	10	
1.3	Knoppie - [Afleringsboks se nommer] Skep veranderlike om boksnommer te stoor ✓ Toets of spoedpos geselekteer is ✓ Stel boksnommer = 4 ✓ else ✓ (kan ook nog 'n if-stelling wees) Genereer ewekansige getal ✓ in die korrekte interval ✓ Verseker dat die waarde nie 4 is nie ✓ ✓ Vertoon boksnommer ✓	9	

VRAAG 1: NASIENRUBRIEK - ALGEMENE PROGRAMMERINGSVAARDIGHEDE
- vervolg

1.4	Knoppie- [Valideer strepieskode] Verkry die strepieskode ✓ Stel somOnEwe en somEwe na 0 ✓ Lus ✓ vanaf eerste tot tweede laaste syfer ✓ Toets of die posisie van die syfer ewe is ✓ Tel syfer by posisie se waarde ✓ (heelgetal) by somEwe ✓ Else Tel syfer by posisie se waarde by somOnEwe ✓ Vermenigvuldig somOnEwe met 3 ✓ Tel somEwe en somOnEwe bymekaar ✓ Bereken kontrolesyfer: totaal modulus 10 ✓ en trek af van 10 ✓ Toets of kontrolesyfer gelyk is aan laaste syfer ✓ (Selfde datatipe) Vertoon gepaste boodskap dat strepieskode geldig is } ✓ Else Vertoon gepaste boodskap dat strepieskode ongeldig is } (Vertoon van die kontrolesyfer nie nodig in if of else nie)	14	
1.5	Knoppie [Vertoon en stoor aflewerings] Verkry die naam van stad wat gekies is uit die kombinasie lys ✓ Vertoon die stad se naam in die afvoerarea as 'n opskrif ✓ Skep 'n tekslêer ✓ met die korrekte saamgestelde naam ✓ Lus vanaf eerste posisie ✓ tot laaste posisie in die skikking ✓ Toets of stadnaam deel is ✓ van die korrekte skikkinginskrywing ✓ Vertoon die skikkingelement in die afvoerarea indien gevind ✓ en stoor die skikkingelement in die tekslêer ✓ Een aflewering per lyn ✓ Maak die tekslêer buite die lus toe ✓	12	
	TOTAAL:	50	

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
2.1.1	Konstruktor: Opskrif met SLEGS vier waardes✓ Korrekte datatipes✓ Ken parameter waardes toe aan die vier attribute✓ (verstekwaarde vir brandstofGebruik kan ingesluit word)	3	
2.1.2	Wysigings en toegangs METODES: setBrandstofGebruik (geen return)✓ met korrekte parameter waarde toegeken aan attribute✓ getBrandstofGebruik (moet waarde terug stuur)✓ met korrekte terugstuur datatipe✓	4	
2.1.3	berekenAfstand METODE: data tipe van terugstuur 'n nommer ✓ trek odoBegin af van odoEind ✓ (-1 indien parameters ontvang word) stuur antwoord terug✓	3	
2.1.4	bepaalTolgeld METODE: Parameter ontvang✓ Bepaal korrekte ry in 2D afhangende van roeteNr✓✓ (trek 1 punt af per fout – maks 2) Bepaal korrekte kolom in 2D afhangende VragmotorNr Tr1 OF ✓Tr2: 1 ^{ste} kolom✓, Tr3: 2 ^{de} kolom✓, Tr4 OF Tr5: 3 ^{de} kolom✓ Vind en stuur tolgeld waarde terug✓ by posisie [ry] ✓ [kolom]✓ (-1 vir verkeerde volgorde)	10	
2.2.1	Knoppie – [Kry data uit lêer]: Verkry vragmotornommer uit kombinasie lys en skakel om na string✓ {Delphi: AssignFile, Reset Java: Create object om te lees vanaf lêer} ✓✓ Toets indien die lêer nie bestaan nie✓ & vertoon boodskap✓ Gebruik lusteller/afleringsnommer van laaste inskrywing✓ Inkrementeer laaste afleringsnommer vir nuwe Aflering nommer✓ Lus deur lêer ✓ Lees lyn vanaf teks lêer✓ Vind ✓ en hou laaste voorkoms van die voertuig nommer✓ Skei inligting deur (#) te gebruik – om die odometer lesing te verkry split/copy/pos/indexOf ✓ Verkry odometer lesing vanaf string✓ Vertoon waardes in teksblokkies✓	14	

**VRAAG 2: NASIENRUBRIEK-OBJEK-GEÛRIENTEERDE PROGRAMMERING -
vervolg**

2.2.2	<p>Knoppie – [Nuwe aflewering]: Verkry nuweAfleweringsnommer, begin oudometer lesing, eind-odometer lesing en voertuignommer ✓ <i>Instantiansieer die objek met waardes:</i> Links van toekenning ✓ Regs van toekenning, Korrekte aantal parameters en volgorde (4) ✓ en skakel datatipes om ✓ (of kry uit veranderlikes met regte datatipes) Vertoon boodskap nadat objek geskep is ✓ Gebruik objek ✓ en roep berekenAfstand ✓ en bereken brandstofgebruik ✓ (deel deur 5) (Java: Verseker dat brandstofGebruik double waarde is) Roep set metode en stel brandstofGebruik attribute ✓ na berekende waarde Maak die brandstof gebruik en tolgeld knoppies ✓ beskikbaar (Kan ook in 2.2.3 gedoen word)</p>	10	
2.2.3	<p>Knoppie – [Vertoon aflewering]: Vertoon inligting in teksarea ✓ deur gebruik van toString ✓</p>	2	
2.2.4	<p>Knoppie [Kontroleer brandstof gebruik]: Lees brandstof wat gebruik is in vanaf teksblokkie en skakel om na reële getal ✓ Gebruik objek en roep getMethod om brandstofGebruik te kry ✓ Bereken verskil ✓ bepaal % ✓ Toets of verskil minder is as 10% ✓ (Neem in ag vir positief en negatief) ✓ Verander brandstof gebruik deur set metode te gebruik ✓ Vertoon boodskap om aan te dui dat verandering plaasgevind het ✓ Else Vertoon foutboodskap ✓</p>	9	
2.2.5	<p>Knoppie – [Bereken tolgeld] Lees die roete nommer vanaf die teksblokkie ✓ Gebruik objek en roep bepaalTolgeld metode ✓ en Stuur roetenommer as 'n parameter ✓ Vertoon tolgeld ✓ in rand formaat (R##.##) ✓</p>	5	
	TOTAAL:	60	

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

SENTRUMNOMMER:		EKSAMENNOMMER:	
VRAAG	BESKRYWING	MAKS. PUNTE	LEERDER-PUNT
3.1	<p>Knoppie – [Laaitem] Verklaar veranderlike vir laaikode ✓ Toets of breekbaar geselekteer is ✓ Toets of ✓ breekbareItems < 20 ✓ Skep laaikode F ✓ nommer ✓ Dateer breekbareItems op ✓ deur * by te voeg ✓ Else ✓ (Toets of nie-breekbaar geselekteer is) Toets of ✓ nieBreekbareItems < 30 ✓ Skep laaikode ✓ Dateer nieBreekbareItems op deur * by te voeg ✓ Toets of daar spasie is om item te laai ✓ Vertoon laaikode in die teksblokkie ✓ Else ✓ Vertoon boodskap dat item nie gelaai is nie ✓ Vertoon opskrif ✓ Vertoon string wat breekbareItems verteenwoordig ✓ Vertoon string wat nieBreekbareItems verteenwoordig ✓</p>	20	
3.2	<p>Knoppie – [Kontroleer vraagstatus]: Bepaal aantal breekbare items EN Bepaal aantal nie-breekbare items ✓ <i>(Punt indien waardes 4 of 13 toegeken word)</i> <i>(Punt word toegeken indien waardes in 3.1 gebruik word)</i> Bereken persentasie breekbare items ✓ Bereken persentasie nie-breekbare items ✓ Vertoon kolomopskrifte ✓ en detail in kolomme ✓ Vertoon breekbare item besonderhede, ✓ geformateer, persentasie tot 2 desimale plekke ✓ Vertoon nie-breekbare item besonderhede ✓, geformateer, persentasie tot 2 desimale plekke If voorwaarde <i>(persentasie breekbaar >=50 ✓ AND ✓ persentasie nie-breekbaar >=50 ✓)</i> Nested if aanvaarbaar Vertoon boodskap dat aflewering kan “voort gaan” in teksarea ✓ else Vertoon boodskap “mag nie voortgaan” ✓ If (voorwaarde) ✓ <i>(aantBreekbaar <10)</i> Vertoon breekbare items in teksarea, berekening ✓ If (voorwaarde) ✓ <i>(aantNieBreekbaar <15)</i> Vertoon nie breekbare items in teksarea, berekening ✓ NOTA: Die kondisie kan ook omgedraai word met ooreenstemmende boodskappe wat vertoon word. Getalle (10 en 15) kan gebruik word ipv persentasies</p>	17	

VRAAG 3: NASIENRUBRIEK – PROBLEEMOPLOSSING - vervolg

3.3	Knoppie – [Verwyder vraag] Maak breekbare items skoon ✓ Maak nie breekbare items skoon ✓ Maak teksarea skoon ✓	3	
	TOTAAL:	40	

OPSOMMING VAN LEERDER PUNTE:

	AFDELING A	AFDELING B	AFDELING C	
	VRAAG 1	VRAAG 2	VRAAG 3	GROOT-TOTAAL
MAKS. PUNTE	50	60	40	150
LEERDER PUNTE				

BYLAAG D: OPLOSSING VIR VRAAG 1: JAVA

```
// 'n Oplossing vir Vraag 1
package Vraag1Package;

import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Calendar;
import java.util.Scanner;
import javax.swing.JOptionPane;

public class Vraag1_Oplossing extends javax.swing.JFrame {

    int kilometers = 635;

    public Vraag1_Oplossing() {
        initComponents();
        this.setLocationRelativeTo(this);
        this.setVisible(true);
        lstKgs.setSelectedIndex(0);
        txfStrepieskode.setText("639382000393");
    }
    =====
    // Vraag 1.1
    =====
    private void btnAflerwingActionPerformed(java.awt.event.ActionEvent evt) {
        String vertrek = (String) (cmbVertrek.getSelectedItem());
        String bestemming = (String) (cmbBestemming.getSelectedItem());
        kilometers = Integer.parseInt(txfAfstand.getText());
        lblAflerwer.setText(vertrek + " na " + bestemming + " : " + kilometers + "
        km");
    }
    =====
    // Vraag 1.2
    =====
    int posisie = (int) (lstKgs.getSelectedIndex());
        double kosteVervoer = 0;
        switch (posisie) {
            case 0:
                kosteVervoer = 0.6 * kilometers;
                break;
            case 1:
                kosteVervoer = 1.0 * kilometers;
                break;
            case 2:
                kosteVervoer = 1.25 * kilometers;
                break;
            case 3:
                kosteVervoer = 1.65 * kilometers;
                break;
        }

        if (chbSpoedPos.isSelected()) {
            kosteVervoer += 100;
        }
        txfKoste.setText(String.format("R%2.2f", kosteVervoer));
    }
}
```

```

=====
// Vraag 1.3
=====
private void btnBoksNummerActionPerformed(java.awt.event.ActionEvent evt) {
    int boksNommer = 0;
    if (chbSpoedPos.isSelected()) {
        boksNommer = 4;
    } else {
        do {
            boksNommer = (int) (Math.random() * 5) + 1;
        } while (boksNommer == 4);
    }
    txfBoksNommer.setText("" + boksNommer);
}
=====
// Vraag 1.4
=====
private void btnStrepieskodeActionPerformed(java.awt.event.ActionEvent evt) {
    String strepiesKode = txfStrepieskode.getText();
    int somOnewe = 0;
    int somEwe = 0;
    for (int tel = 0; tel < strepiesKode.length() - 1; tel++) {
        if ((tel + 1) % 2 == 0) {
            somEwe = somEwe + Integer.parseInt(strepiesKode.substring(tel,
                tel + 1));
        } else {
            somOnewe = somOnewe + Integer.parseInt(
                strepiesKode.substring(tel, tel + 1));
        }
    }
    int sum = somOnewe * 3 + somEwe;
    int toetsGetal = 10 - (sum % 10);
    if (toetsGetal == Integer.parseInt(strepiesKode.substring
        (strepiesKode.length() - 1))) {
        txfVertoonStrepiesKode.setText("Die strepieskode nommer is geldig.
            Kontrolesyfer: " + toetsGetal);
    } else {
        txfVertoonStrepiesKode.setText("Die strepieskode nommer is NIE
            geldig nie. Korrekte kontrolesyfer: " + toetsGetal);
    }
}
=====
// Vraag 1.5
=====
private void btnVertoonAfleweringsActionPerformed(java.awt.event.ActionEvent
    evt) {
    String plek = (String) (cmbStadNaam.getSelectedItem());
    afvoerArea.setText(plek + "\n");
    try {
        PrintWriter out = new PrintWriter(new FileWriter("Desember2014" +
            plek + ".txt"));
        for (int i = 0; i < arrDesAflewerings.length; i++) {
            if (arrDesAflewerings[i].indexOf(plek) >= 0) {
                afvoerArea.append(arrDesAflewerings[i] + "\n");
                out.println(arrDesAflewerings[i]);
            }
        }
        out.close();
    } catch (IOException e) {
        JOptionPane.showMessageDialog(null, "FOUT");
    }
}

```

BYLAAG E: OPLOSSING VIR VRAAG 2: JAVA

```
// Moontlike oplossing vir vraag 2
```

OBJEKKLAS: AFLEWERING (GEGEE)

```
public class Aflewering {
//=====
// Gegewe kode
//=====
    private int afleweringNom;
    private String vragmotorNom;
    private double brandstofGebruik;
    private int odoBegin;
    private int odoEind;

    double[][] tolGeld = {{105.50, 135.00, 210.00},
                          {35.00, 54.00, 82.00},
                          {85.00, 129.00, 205.00},
                          {112.00, 170.00, 219.00}};

    public String toString() {
        DecimalFormat df = new DecimalFormat("0.0");
        String afvoer = "Afleweringsnommer: " + afleweringNom +
            "\nVragmotornommer: " + vragmotorNom + "\nOdometerlesing:
            \n\t(Begin) " + odoBegin + "\n\t(Eind) " + odoEind +
            "\nBrandstof gebruik: " + df.format(brandstofGebruik) + " liter";
        return afvoer;
    }
//=====
// Vraag 2.1.1
//=====

    public Aflewering(int afleweringNom, String vragmotorNom, int odoBegin, int
odoEind) {
        this.afleweringNom = afleweringNom;
        this.vragmotorNom = vragmotorNom;
        this.odoBegin = odoBegin;
        this.odoEind = odoEind;
    }
//=====
// Vraag 2.1.2
//=====
//Accessor metode

    public double getBrandstofGebruik() {
        return brandstofGebruik;
    }

//Mutator metode

    public void setBrandstofGebruik(double brandstof) {
        brandstofGebruik = brandstof;
    }
}
```

```
=====
// Vraag 2.1.3
=====
```

```
public int berekenAfstand() {
    return odoEind - odoBegin;
}
}
```

```
=====
// Vraag 2.1.4
=====
```

```
public double bepaalTolgelde(String roete) {
    double tolBedrag = 0;
    int ry = Integer.parseInt(roete.substring(2, 3)) - 1;

    if (vragmotorNom.equals("Tr1") || vragmotorNom.equals("Tr2")) {
        tolBedrag = tolGeld[ry][0];
    } else if (vragmotorNom.equals("Tr3")) {
        tolBedrag = tolGeld[ry][1];
    } else {
        tolBedrag = tolGeld[ry][2];
    }

    /* Alternatief:
    switch (vragmotorNom) {
        case "Tr1":
        case "Tr2":
            tolBedrag = tolGeld[ry][0];
            break;
        case "Tr3":
            tolBedrag = tolGeld[ry][1];
            break;
        default:
            tolBedrag = tolGeld[ry][2];
            break;
    }

    */
    return tolBedrag;
}
```

GUI CLASS: 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_Oplossing extends javax.swing.JFrame {
=====
// Gegewe kode
=====
Aflewering objAflewering;

    public Vraag2_Oplossing() {
        initComponents();
        this.setLocationRelativeTo(this);
        this.setVisible(true);
        btnTolgeld.setEnabled(false);
        btnBrandstofVerander.setEnabled(false);
    }

// Kode vir grafieka nie gekopieer
=====
// Vraag 2.2.1
=====
private void btnKryVanuitLeerActionPerformed(java.awt.event.ActionEvent evt) {
    File leer = new File("AfleweringsInligting.txt");
    if (!leer.exists()) {
        JOptionPane.showMessageDialog(rootPane, "Leer bestaan nie");
        System.exit(0);
    } else {

        String vragmotorNo = (String) cmbVragmotorNommer.getSelectedItem();
        try {
            String laasteVragmotorLyn = "", lyn = "";
            Scanner sc = new Scanner(new FileReader(
                "AfleweringsInligting.txt"));
            String[] temp;
            while (sc.hasNext()) {
                lyn = sc.next();
                if (lyn.contains(vragmotorNo)) {
                    laasteVragmotorLyn = lyn;
                }
            }
            temp = lyn.split("#");
            int nuweRoete = Integer.parseInt(temp[0]) + 1;
            txfNuweRoeteNommer.setText("" + nuweRoete);
            temp = laasteVragmotorLyn.split("#");
            txfBeginOdometer.setText(temp[2]);
        }
        catch (Exception e) {
        }
    }
}
}

```

NSS – Memorandum

```

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

private void btnNuweAfleweringActionPerformed(java.awt.event.ActionEvent evt) {
    int nuweRoeteNommer = Integer.parseInt(txfNuweRoeteNommer.getText());
    int beginOdoLesing = Integer.parseInt(txfBeginOdometer.getText());
    int eindOdoLesing = Integer.parseInt(txfEindOdometer.getText());
    String vragmotorNo = (String) cmbVragmotorNommer.getSelectedItem();
    objAflewering = new Aflewering(nuweRoeteNommer, vragmotorNo,
        beginOdoLesing, eindOdoLesing);
    JOptionPane.showMessageDialog(rootPane, "Aflewerings-objek is suksesvol
        geskep.");
    int afstand = objAflewering.berekenAfstand();
    objAflewering.setBrandstofGebruik(afstand / 5.0);
    btnTolgeld.setEnabled(true);
    btnBrandstofVerander.setEnabled(true);
}

=====
// Vraag 2.2.3
=====
private void btnVertoonAfleweringActionPerformed(java.awt.event.ActionEvent
    evt) {
    txaAfvoer.setText(objAflewering.toString());
}

=====
// Vraag 2.2.4
=====
private void btnBrandstofVeranderActionPerformed(java.awt.event.ActionEvent
    evt)
{
    double brandstofIngegooi = Double.parseDouble(txfBrandstof.getText());
    double brandstofGebruik = objAflewering.getBrandstofGebruik();
    if (Math.abs(brandstofIngegooi - brandstofGebruik) / brandstofGebruik < 0.1)
    {
        objAflewering.setBrandstofGebruik(brandstofIngegooi);
        txfBrandstofBoodskap.setText("Brandstof gebruik van " + brandstofGebruik
            + " na " + brandstofIngegooi + " liter verander");
    } else {
        txfBrandstofBoodskap.setText("FOUT: Die verskil in brandstof gebruik is
            te groot");
    }
}

=====
// Vraag 2.2.5
=====
private void btnTolgeldActionPerformed(java.awt.event.ActionEvent evt) {
    String output = String.format("%-23sR%2.2f", "Tolgeld betaalbaar:",
        objAflewering.bepaalTolgelde(txfRoete.getText()));
    lblTolgeld.setText(output);
}

```

BYLAAG F: OPLOSSING VIR VRAAG 3: JAVA

```
// Moontlike oplossing Vir vraag 3

package Vraag3Package;

import javax.swing.JOptionPane;

public class Vraag3_Oplossing extends javax.swing.JFrame {

//Globale veranderlikes
    String breekbareItems = "";
    String nieBreekbareItems = "";

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

public Vraag3_Oplossing() {
    initComponents();
    this.setLocationRelativeTo(this);
    this.setVisible(true);
    rbtBreekbaar.setSelected(true);
}

=====
// Vraag 3.1
=====
private void btnLaaiActionPerformed(java.awt.event.ActionEvent evt) {
    String laaiKode = "";
    if (rbtBreekbaar.isSelected()) {
        if (breekbareItems.length() < 20) {
            laaiKode = "B" + (breekbareItems.length() + 1);
            breekbareItems += "*";
        }
    } else {
        if (nieBreekbareItems.length() < 30) {
            laaiKode = "NB" + (nieBreekbareItems.length() + 1);
            nieBreekbareItems += "*";
        }
    }
    txflaaiKode.setText(laaiKode);
    if (laaiKode.equals("")) {
        JOptionPane.showMessageDialog(null, "Laai van item kan nie verwerk
        word nie - Geen laaispasie nie \n", "Information", WIDTH);
    }
    txafvoer.setText("Laai vordering-vertoonarea:
        \n===== \n\n");
    txafvoer.append(String.format("%-24s%-25s\n", "Breekbare items:",
        breekbareItems));
    txafvoer.append(String.format("%-24s%-25s", "Nie-breekbare items:",
        nieBreekbareItems));
}
}
```

```
=====
// Vraag 3.2
=====

private void btnStatusActionPerformed(java.awt.event.ActionEvent evt) {
    int aantBreekbaar = breekbareItems.length();
    int aantNieBreekbaar = nieBreekbareItems.length();
    double persBreekbaar = (aantBreekbaar) / 20.0 * 100;
    double persNieBreekbaar = (aantNieBreekbaar) / 30.0 * 100;
    txafvoer.setText(" Vragstatusverslag:\n =====\n");
    txafvoer.append(String.format("%-15s%-25s%-15s\n", " Item tipe", "Getal
        items", "Persentasie gelaai"));
    txafvoer.append(String.format("%-15s%-25s%-13.2f\n", " Breekbaar",
        aantBreekbaar, persBreekbaar));
    txafvoer.append(String.format("%-15s%-25s%-13.2f\n", " Nie-breekbaar",
        aantNieBreekbaar, persNieBreekbaar));
    if (persBreekbaar >= 50 && persNieBreekbaar >= 50) {
        txafvoer.append("\n Die aflewering mag voortgaan.");
    }
    else
    {
        txafvoer.append("\n Die aflewering mag nie voortgaan nie.");
        if (aantBreekbaar < 10) {
            txafvoer.append("\n Getal breekbare items wat nog benodig word: "
                + (10 - aantBreekbaar));
        }
        if (aantNieBreekbaar < 15) {
            txafvoer.append("\n Getal nie-breekbare items wat nog benodig
                word: " + (15 - aantNieBreekbaar));
        }
    }
}

=====
// Vraag 3.3
=====

private void btnVerwyderActionPerformed(java.awt.event.ActionEvent evt) {
    breekbareItems = "";
    nieBreekbareItems = "";
    txafvoer.setText("");
}
}
```


BYLAAG G: OPLOSSING VIR VRAAG 1: DELPHI

```
unit Vraag1_U_Memo;

interface
//Moontlike oplossing vir Vraag1

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

type
  TfrmVraag1 = class(TForm)
    grpQ1_1: TGroupBox;
    grpQ1_3: TGroupBox;
    grpQ1_2: TGroupBox;
    grpQ1_4: TGroupBox;
    lblAflewerVanaf: TLabel;
    lblAflewerNA: TLabel;
    lblAantalKM: TLabel;
    edtKM: TEdit;
    btnBevestigAflewing: TButton;
    grpDLabel: TGroupBox;
    lblAfleweringsKode: TLabel;
    grpSpeedpost: TGroupBox;
    btnAfleweringsKoste: TButton;
    grpRange: TGroupBox;
    edtAfleweringsKoste: TEdit;
    lstKM: TListBox;
    btnAfleweringsBoksNommer: TButton;
    edtAfleweringsBoksNommer: TEdit;
    btnSkepStrepiesKode: TButton;
    edtVertoonStrepiesKode: TEdit;
    cboVertrekVanaf: TComboBox;
    cboBestemming: TComboBox;
    lblUPCBarCode: TLabel;
    edtUPKStrepiesKode: TEdit;
    chkSpoedPos: TCheckBox;
    grpQ1_5: TGroupBox;
    cboStadNaam: TComboBox;
    btnVertoonAflewerings: TButton;
    redAfvoerArea: TRichEdit;
    lblCity: TLabel;
    procedure btnBevestigAflewingClick(Sender: TObject);
    procedure btnAfleweringsKosteClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure btnAfleweringsBoksNommerClick(Sender: TObject);
    procedure btnSkepStrepiesKodeClick(Sender: TObject);
    procedure btnVertoonAfleweringsClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag1: TfrmVraag1;
  iKilometers : Integer = 635; //verstek waarde
```

```
//given do not change
arrDesAflewerings : array[1..32] of String =
('2014-12-01 Durban na Kaapstad',
'2014-12-01 Polokwane na Johannesburg',
'2014-12-02 Kaapstad na Johannesburg ',
'2014-12-02 Polokwane na Potchefstroom ',
'2014-12-02 Bloemfontein na Port Elizabeth',
'2014-12-03 Polokwane na Potchefstroom',
'2014-12-03 Kaapstad na Port Elizabeth ',
'2014-12-03 Port Elizabeth na Potchefstroom ',
'2014-12-04 Port Elizabeth na Durban',
'2014-12-04 Polokwane na Kimberley',
'2014-12-04 Kaapstad na Kimberley ',
'2014-12-04 Polokwane na Potchefstroom ',
'2014-12-04 Kimberley na Port Elizabeth',
'2014-12-05 Durban na Kimberley',
'2014-12-05 Bloemfontein na Potchefstroom',
'2014-12-05 Durban na Potchefstroom',
'2014-12-05 Kaapstad na Potchefstroom',
'2014-12-05 Polokwane na Kaapstad',
'2014-12-06 Kaapstad na Johannesburg ',
'2014-12-06 Polokwane na Potchefstroom ',
'2014-12-06 Bloemfontein na Kimberley',
'2014-12-06 Polokwane na Johannesburg',
'2014-12-07 Kaapstad na Port Elizabeth ',
'2014-12-07 Port Elizabeth na Potchefstroom ',
'2014-12-07 Potchefstroom na Durban',
'2014-12-07 Kaapstad na Kimberley',
'2014-12-08 Kaapstad na Kimberley ',
'2014-12-08 Polokwane na Potchefstroom ',
'2014-12-08 Kimberley na Port Elizabeth',
'2014-12-08 Potchefstroom na Kimberley',
'2014-12-09 Bloemfontein na Polokwane',
'2014-12-09 Durban na Bloemfontein');
```

implementation

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

```
procedure TfrmVraag1.btnBevestigAfleweringClick(Sender: TObject);
begin
=====
// Vraag 1.1
=====
    iKilometers := StrToInt(edtKM.Text);
    lblAfleweringsKode.Caption :=
        cboVertrekVanaf.Items[cboVertrekVanaf.ItemIndex] + ' na ' +
        cboBestemming.Items[cboBestemming.ItemIndex] + ' : ' +
        edtKm.Text + 'km';
end;

procedure TfrmVraag1.btnAfleweringsKosteClick(Sender: TObject);
var
    iPosisie      : integer;
    rKosteVervoer : real;
begin
=====
// Vraag 1.2
=====
    iPosisie := lstKM.ItemIndex;
    rKosteVervoer := 0;
Kopiereg voorbehou
```

Blaai om asseblief

NSS – Memorandum

```

Case (iPosisie) of
  0: rKosteVervoer := 0.60 * iKilometers;
  1: rKosteVervoer := 1.00 * iKilometers;
  2: rKosteVervoer := 1.25 * iKilometers;
  3: rKosteVervoer := 1.65 * iKilometers;
end;

if chkSpoedPos.Checked = True
then
begin
  rKosteVervoer := rKosteVervoer + 100;
end;
edtAfleweringsKoste.Text := FloatToStrF(rKosteVervoer, ffCurrency, 8,2);
end;

procedure TfrmVraag1.btnAfleweringsBoksNommerClick(Sender: TObject);
var
  iBoksNommer : integer;
begin
=====
// Vraag 1.3
=====
  if chkSpoedPos.Checked = true
  then
begin
  iBoksNommer := 4;
end
else
begin
  //genereer 'n lukraak getal tussen 1 en 5 maar nie 4
  repeat
    iBoksNommer := random(5)+1;
  until iBoksNommer <> 4;
end;
  edtAfleweringsBoksNommer.Text := IntToStr(iBoksNommer);
end;

procedure TfrmVraag1.btnSkepStrepiesKodeClick(Sender: TObject);
var
  sStrepiesKode : string;
  iSomOnewe, iSomEwe, iTeller, iTotaal, iKontroleSyfer : Integer;
begin
=====
// Vraag 1.4
=====
  sStrepiesKode := edtUPKStrepiesKode.Text;
  iSomOnewe := 0;
  iSomEwe := 0;
  for iTeller := 1 to Length(sStrepiesKode)-1 do
begin
  if (iTeller MOD 2) = 0
  then inc(iSomEwe, StrToInt(sStrepiesKode[iTeller]))
  else inc(iSomOnewe, StrToInt(sStrepiesKode[iTeller]));
end;
  iTotaal := (iSomOnewe * 3) + iSomEwe;
  iKontroleSyfer := 10 - (iTotaal mod 10);
  if iKontroleSyfer = StrToInt(sStrepiesKode[Length(sStrepiesKode)])
  then
    edtVertoonStrepiesKode.Text := 'Die strepieskode is geldig. ' +
      'Kontrolesyfer: ' + IntToStr(iKontroleSyfer)
  else
    edtVertoonStrepiesKode.Text := 'Die strepieskode is NIE geldig nie. ' +

```

```
        ' Korrekte kontrolesyfer: ' +
IntToStr(iKontroleSyfer) ;
end;

procedure TfrmVraag1.btnVertoonAfleweringsClick(Sender: TObject);
var
    sStad, sLeerNaam    : string;
    txtFile             : TextFile;
    iTeller             : Integer;
begin
=====
// Vraag 1.5
=====
    redAfvoerArea.Clear;
    sStad := cboStadNaam.Items[cboStadNaam.ItemIndex];
    redAfvoerArea.Lines.Add(sStad);
    sLeerNaam := 'Desember2014'+sStad + '.txt';
    AssignFile(txtFile, sLeerNaam);
    Rewrite(txtFile);
    for iTeller := 1 to 32 do
        begin
            if pos(sStad, arrDesAflewerings[iTeller]) > 0
            then
                begin
                    redAfvoerArea.Lines.Add(arrDesAflewerings[iTeller]);
                    Writeln(txtFile, arrDesAflewerings[iTeller]);
                end;
            end;
        CloseFile(txtFile);
    end;

procedure TfrmVraag1.FormCreate(Sender: TObject);
begin
    lstKM.Selected[0] := True;
    CurrencyString := 'R';
    Randomize;
end;

end.
```

BYLAAG H: OPLOSSING VIR VRAAG 2: DELPHI**KLASEENHEID: AFLEWERING_U.PAS**

```

unit Aflewinging_U;
  //Moontlike Oplossing vir Vraag 2 - class unit.

interface

uses
  sysUtils;

type
  TAflewering = class(TObject)
  private
    fAfleweringNom      : integer;
    fVragmotorNom       : string;
    fBrandstofGebruik  : real;
    fOdoBegin          : integer;
    fOdoEind           : integer;
  public
    function toString: string;
    constructor Create(iAfleweringNom: integer; sVragmotorNom : string;
                      iOdoBegin, iOdoEind : integer);

    function getBrandstofGebruik: real;
    procedure setBrandstofGebruik (rBrandstofGebruik : Real);
    function berekenAfstand: Integer;
    function bepaalTolgelde(sRoete : string): real;
  end;

var
  =====
  //Given to be used in Vraag 2.1.4
  =====
  tolgeld : array[1..4,1..3] of real =
    ((105.50, 135.00, 210.00), (35.00, 54.00, 82.00),
     (85.00,129.00,205.00), (112.00, 170.00, 219.00));

implementation
  =====
  // Vraag 2.1.1.
  =====
  constructor TAflewering.Create(iAfleweringNom: integer; sVragmotorNom:
  string;
    iOdoBegin, iOdoEind: integer);
  begin
    fAfleweringNom      := iAfleweringNom;
    fVragmotorNom       := sVragmotorNom;
    fOdoBegin          := iOdoBegin;
    fOdoEind           := iOdoEind;
  end;
  =====
  // Vraag 2.1.2.
  function TAflewering.getBrandstofGebruik: real;
  begin
    Result := fBrandstofGebruik;
  end;

```

```

procedure TAflewering.setBrandstofGebruik(rBrandstofGebruik: Real);
begin
  fBrandstofGebruik := rBrandstofGebruik;
end;
=====
// Vraag 2.1.3.
=====
function TAflewering.berekenAfstand: Integer;
begin
  Result := fOdoEind - fOdoBegin;
end;
=====
// Vraag 2.1.4.
=====
function TAflewering.bepaalTolgelde(sRoete: string): real;
var
  iRy : integer;
begin
  Result := 0;
  iRy := StrToInt(sRoete[3]); //3de karakter
  if (fVragmotorNom = 'Tr1') OR (fVragmotorNom = 'Tr2')
  then Result := tolgeld[iRy, 1]
  else if (fVragmotorNom = 'Tr3')
  then Result := tolgeld[iRy, 2]
  else Result := tolgeld[iRy, 3];

  {Alternative:
  case fVragmotorNom[3] of
    '1', '2' : Result := tolgeld[iRy, 1];
    '3'      : Result := tolgeld[iRy, 2];
    '4', '5' : Result := tolgeld[iRy, 3];
  end; //case
  }
end;

function TAflewering.toString: string;
begin
  Result := 'Afleweringsnommer: ' + IntToStr(fAfleweringNom) + #13 +
    'Vragmotornommer: ' + fVragmotorNom + #13 +
    'Odometerlesing: ' + #13 +
    #9 + '(Begin) ' + IntToStr(fOdoBegin) + #13 +
    #9 + '(Einde) ' + IntToStr(fOdoEind) + #13 +
    'Brandstof gebruik: ' + FloatToStr(fBrandstofGebruik) + '
    liter';
end;

end.

```

HOOFVORM EENHEID: VRAAG2_U.PAS

```
unit Vraag2_U_Memo;
//Moontlike oplossing vir Vraag 2 - Formunit.

interface

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

type
  TfrmVraag2 = class(TForm)
    pnlTitle: TPanel;
    grpCreateDisplay: TGroupBox;
    GroupBoxOptionC: TGroupBox;
    lblVragmotorNommer: TLabel;
    cboVragmotorNommer: TComboBox;
    redAfvoer: TRichEdit;
    lblEindOdoLesing: TLabel;
    edtEindOdometer: TEdit;
    btnNuweAflewing: TButton;
    btnVertoonAflewing: TButton;
    lblWerklikeBrandstofGebruik: TLabel;
    edtBrandstofGebruik: TEdit;
    btnBrandstofVerander: TButton;
    edtBrandstofBoodskap: TEdit;
    grpTollFees: TGroupBox;
    lblRoete: TLabel;
    edtRoete: TEdit;
    btnTolgeld: TButton;
    pnlTolgeld: TPanel;
    lblBeginOdoLesing: TLabel;
    edtBeginOdometer: TEdit;
    lblNuweRoeteNommer: TLabel;
    edtNuweRoeteNommer: TEdit;
    btnKryVanafLeer: TButton;
    lblTolgeld: TLabel;

    procedure FormCreate(Sender: TObject);
    procedure btnTolgeldClick(Sender: TObject);
    procedure btnNuweAflewingClick(Sender: TObject);
    procedure btnVertoonAflewingClick(Sender: TObject);
    procedure btnBrandstofVeranderClick(Sender: TObject);
    procedure btnKryVanafLeerClick(Sender: TObject);

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

var
  frmVraag2: TfrmVraag2;

  Aflewing : TAflewing;

implementation
```

```

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

procedure TfrmVraag2.FormCreate(Sender: TObject);
begin
  CurrencyString := 'R';
end;

procedure TfrmVraag2.btnKryVanafLeerClick(Sender: TObject);
var
  txtFile : TextFile;
  sLyn, sRoeteNr, sVragmotorNr, sBeginOdo : string;
  iNuweRoete : Integer;
begin
  =====
  // Vraag 2.2.1
  =====
  if NOT FileExists('AfleweringsInligting.txt')
  then
  begin
    MessageDlg('AfleweringsInligting.txt bestaan nie', mtError, [mbOK], 0);
    Exit;
  end;

  sVragmotorNr := cboVragmotorNommer.Items[cboVragmotorNommer.ItemIndex];

  AssignFile(txtFile, 'AfleweringsInligting.txt');
  Reset(txtFile);
  while not EOF(txtFile) do
  begin
    readln(txtFile, sLyn);
    sRoeteNr := copy(sLyn, 1, pos('#', sLyn)-1);
    if Pos(sVragmotorNr, sLyn) > 0
    then
    begin
      Delete(sLyn, 1, pos('#', sLyn));
      Delete(sLyn, 1, pos('#', sLyn));
      sBeginOdo := sLyn;
    end; //if
  end;
  closeFile(txtFile);

  iNuweRoete := StrToInt(sRoeteNr) + 1;
  edtNuweRoeteNommer.Text := IntToStr(iNuweRoete);
  edtBeginOdometer.Text := sBeginOdo;
end;

procedure TfrmVraag2.btnNuweAflewingClick(Sender: TObject);
var
  iAfstand, iNuweRoeteNr, iBeginOdoLesing, iEindOdoLesing : integer;
  sVragmotorNr : string;
begin
  =====
  // Vraag 2.2.2
  =====
  sVragmotorNr := cboVragmotorNommer.Items[cboVragmotorNommer.ItemIndex];
  iNuweRoeteNr := StrToInt(edtNuweRoeteNommer.text);
  iBeginOdoLesing := StrToInt(edtBeginOdometer.text);
  iEindOdoLesing := StrToInt(edtEindOdometer.text);

```


NSS – Memorandum

```
Aflewering := TAflewering.Create(iNuweRoeteNr, sVragmotorNr,
    iBeginOdoLesing, iEindOdoLesing);
MessageDlg('Aflewering objek suksesvol geskep.', mtInformation,
    [mbOK],0);

iAfstand := Aflewering.berekenAfstand;
Aflewering.setBrandstofGebruik(iAfstand / 5.0);

btnTolgeld.Enabled := True;
btnBrandstofVerander.Enabled := True;
end;

procedure TfrmVraag2.btnVertoonAfleweringClick(Sender: TObject);
begin
=====
// Vraag 2.2.3
=====
    redAfvoer.Clear;
    redAfvoer.Lines.Add(aflewering.toString);
end;

procedure TfrmVraag2.btnBrandstofVeranderClick(Sender: TObject);
var
    rBrandstofByvoeg, rBrandstofGebruik : real;
begin
=====
// Vraag 2.2.4
=====
    rBrandstofByvoeg := StrToFloat( edtBrandstofGebruik.Text);
    rBrandstofGebruik := aflewering.getBrandstofGebruik;

    if (Abs(rBrandstofByvoeg - rBrandstofGebruik) / rBrandstofGebruik) < 0.1
    then
        begin
            Aflewering.setBrandstofGebruik(rBrandstofByvoeg);
            edtBrandstofBoodskap.Text := 'Brandstof gebruik van ' +
                FloatToStrF(rBrandstofGebruik, ffFixed, 12, 1) + ' na ' +
                FloatToStrF(rBrandstofByvoeg, ffFixed, 12, 1) + ' liter verander';
        end
    else
        edtBrandstofBoodskap.Text := 'FOUT: Verskil in brandstof gebruik is te
            groot';
    end;
end;

procedure TfrmVraag2.btnTolgeldClick(Sender: TObject);
var
    sRoeteNr : string;
begin
=====
// Vraag 2.2.5.
=====
    sRoeteNr := edtRoete.Text;
    lblTolgeld.Caption := 'Tolgeld betaalbaar: ' +
        FloatToStrF(Aflewering.bepaalTolgelde(sRoeteNr), ffCurrency, 8, 2);
end;

end.
```

BYLAAG I: OPLOSSING VIR VRAAG 3: DELPHI

```

unit Vraag3_U_Memo;
// Moontlike oplossing vir Vraag 3.

interface

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

type
  TfrmVraag3 = class(TForm)
    grpLaaiSone: TGroupBox;
    btnVerwyderVrag: TButton;
    btnLaaiItem: TButton;
    btnLaaiStatus: TButton;
    rgpItemTipe: TRadioGroup;
    redAfvoer: TRichEdit;
    edtLaaiKode: TEdit;
    lblLaaiKode: TLabel;
    procedure btnVerwyderVragClick(Sender: TObject);
    procedure btnLaaiItemClick(Sender: TObject);
    procedure btnLaaiStatusClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmVraag3: TfrmVraag3;

  sBreekbareItems      : string;
  sNBreekbareItems    : string;

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

procedure TfrmVraag3.btnLaaiItemClick(Sender: TObject);
var
  sLaaiKode : string;
begin
  =====
  // Vraag 3.1
  =====
  case rgpItemTipe.ItemIndex of
    0 : begin
      if length(sBreekbareItems) < 20
      then
        begin
          sLaaiKode := 'B' + IntToStr(Length(sBreekbareItems)+1);
          sBreekbareItems := sBreekbareItems + '*';
        end
      else sLaaiKode := '';
      end; //breekbaar
  end;

```

```

1 : begin
    if length(sNBreekbareItems) < 30
    then
        begin
            sLaaiKode := 'NB' + IntToStr(Length(sNBreekbareItems)+1);
            sNBreekbareItems := sNBreekbareItems + '*';
        end
    else sLaaiKode := '';
    end;//nie-breekbaar
end;//case
if sLaaiKode = ''
then
    begin
        edtLaaiKode.Clear;
        MessageDlg('Laai van item kan nie verwerk word nie - Geen laaispasie
nie',
                    mtInformation, [mbok], 0);
    end;//indien geen spasie
else
    begin
        edtLaaiKode.Text := sLaaiKode;
        redAfvoer.Clear;
        redAfvoer.Paragraph.TabCount := 1;
        redAfvoer.Paragraph.Tab[0] := 150;
        redAfvoer.Lines.Add('Laaiordering-vertoonarea:');
        redAfvoer.Lines.Add('=====');
        redAfvoer.Lines.Add(' ');
        redAfvoer.Lines.Add('Breekbare items:      ' + #9 + sBreekbareItems);
        redAfvoer.Lines.Add('Nie-breekbare items:' + #9 + sNBreekbareItems);
    end;//spasie beskikbaar
end;

procedure TfrmVraag3.btnLaaiStatusClick(Sender: TObject);
var
    iAantBreekbaar, iAantNieBreekbaar : integer;
    rPersBreekbaar, rPersNieBreekbaar: real;
beginbegin
=====
// Vraag 3.2
=====
    iAantBreekbaar := Length(sBreekbareItems);
    iAantNieBreekbaar := Length(sNBreekbareItems);
    rPersBreekbaar := iAantBreekbaar / 20 * 100;
    rPersNieBreekbaar:= iAantNieBreekbaar / 30 * 100;
    redAfvoer.Clear;
    redAfvoer.Paragraph.TabCount := 2;
    redAfvoer.Paragraph.Tab[0] := 100;
    redAfvoer.Paragraph.Tab[1] := 275;
    redAfvoer.Lines.Add('Vragstatusverslag:');
    redAfvoer.Lines.Add('=====');
    redAfvoer.Lines.Add(' ');
    redAfvoer.Lines.Add('Item tipe' + #9 + 'Getal items' + #9 + 'Persentasie
gelaai');
    redAfvoer.Lines.Add('Breekbaar' + #9 + IntToStr(iAantBreekbaar) + #9 +
                        FloatToStrF(rPersBreekbaar, ffFixed, 8,2));
    redAfvoer.Lines.Add('Nie-breekbaar' + #9 + IntToStr(iAantNieBreekbaar) +
#9 +
                        FloatToStrF(rPersNieBreekbaar, ffFixed, 8,2));
    redAfvoer.Lines.Add(' ');

```

```
if (rPersBreekbaar >= 50) and (rPersNieBreekbaar >= 50)
then
begin
redAfvoer.Lines.Add('Die aflewering mag voortgaan. ');
end
else
begin
redAfvoer.Lines.Add('Die aflewering mag nie voortgaan nie. ');
if(iAantBreekbaar < 10 )
then redAfvoer.Lines.Add('Getal breekbare items nog nodig: ' +
IntToStr(10 - iAantBreekbaar) );

if (iAantNieBreekbaar < 15 )
then redAfvoer.Lines.Add('Getal nie-breekbare items nog nodig: ' +
IntToStr(15 - iAantNieBreekbaar));

end;
end;

procedure TfrmVraag3.btnVerwyderVragClick(Sender: TObject);
begin
=====
// Vraag 3.3
=====
sBreekbareItems := '';
sNBreekbareItems := '';
redAfvoer.Clear;
end;

procedure TfrmVraag3.FormCreate(Sender: TObject);
begin
CurrencyString := 'R';
end;

end.
```