



### **Oefening 1:**

De huidige dag en maand worden ingegeven in de vorm van DDMM. Van 5 personen wordt hun verjaardag (in de vorm van DDMM) ingegeven. Alle invoer wordt gecontroleerd.

Druk af, per persoon, hoeveel dagen het nog duurt vooraleer het hun verjaardag is.

Schrijf een applicatie. Een tabel met het aantal dagen van de maand wordt geïnitialiseerd.

De verjaardagen van de 5 personen wordt in een array bijgehouden.

Zorg ervoor dat de methode main enkel uit declaraties en het oproepen van methodes, bestaat. Schrijf geen code dubbel.

```
// Verjaardag.java
```

```
// Java extension packages  
import javax.swing.*;
```

```
public class Verjaardag  
{  
    private static final int aantalDagen[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  
  
    public static void main (String args[])  
    {  
        //huidige datum  
        int huidigeDatum = invoerDatum();  
  
        int verjaardagen[] = new int[5];  
        invoerVerjaardagen(verjaardagen);  
  
        String result = maakUitvoerAantalDagen(huidigeDatum, verjaardagen);  
  
        JOptionPane.showMessageDialog(null, result);  
        System.exit(0);  
    } // einde methode main  
  
    static void invoerVerjaardagen(int verjaardagen[])  
    {  
        for (int i=0 ; i<verjaardagen.length; i++)  
            verjaardagen[i] = invoerDatum();  
    } //einde methode invoerVerjaardagen
```

```

static int invoerDatum()
{
    String datumS, result;
    int datum, dag, maand;
    boolean geldigeDatum;
    do
    {
        geldigeDatum = true;
        result = "";
        datumS = JOptionPane.showInputDialog("Geef huidige datum (DDMM) ");
        datum = Integer.parseInt(datumS);
        if (datum < 101 || datum > 3112)
            result = "ongeldige datum";
        else
        {
            dag = datum / 100;
            maand = datum % 100;
            if (maand < 1 || maand > 12)
                result = "ongeldige maand : " + maand;
            else
                if (dag < 1 || dag > aantalDagen[maand])
                    result = "ongeldige dag : " + dag + '-' + maand;
        }
        if (result.length() != 0)
        {
            geldigeDatum = false;
            JOptionPane.showMessageDialog(null, result, "foutboodschap",
                JOptionPane.WARNING_MESSAGE);
        }
    } while (!(geldigeDatum));
    return datum;
} //einde methode invoerDatum

```

```

static String maakUitvoerAantalDagen(int huidigeDatum, int verjaardagen[])
{
    String result = "";
    int aantalDagen;
    for (int i=0 ; i<verjaardagen.length; i++)
    {
        aantalDagen = berekeningAantalDagen(huidigeDatum, verjaardagen[i]);
        result += "Nog " + aantalDagen + " vooraleer persoon " + (i+1) + " verjaart\n";
    }
    return result;
} //einde methode maakUitvoerAantalDagen

```

```

static int berekeningAantalDagen(int huidigeDatum, int verjaardag)
{
    int aantal = 0;
    int huidigeDag = huidigeDatum / 100;
    int huidigeMaand = huidigeDatum % 100;
    int verjaardagDag = verjaardag / 100;
    int verjaardagMaand = verjaardag % 100;
    if (huidigeMaand < verjaardagMaand)
    {
        for (int i = huidigeMaand+1 ; i < verjaardagMaand ; i++ )
            aantal += aantalDagen[i];
        aantal += (aantalDagen[huidigeMaand] - huidigeDag) + verjaardagDag;
    }
    else
        if (huidigeMaand > verjaardagMaand ||
            (huidigeMaand == verjaardagMaand && huidigeDag > verjaardagDag))
        {
            for (int i = huidigeMaand+1 ; i <= 12; i++ )
                aantal += aantalDagen[i];

            for (int i = 1 ; i < verjaardagMaand ; i++ )
                aantal += aantalDagen[i];
            aantal += (aantalDagen[huidigeMaand] - huidigeDag) + verjaardagDag;
        }
        else
        {
            aantal = verjaardagDag - huidigeDag;
        }
    return aantal;
} //einde methode maakUitvoerAantalDagen

} // einde klasse Verjaardag

```

#### **andere oplossing :**

in de array “verjaardagen” op index 0 de huidige datum bijhouden. De vijf personen op index 1 t.e.m. 5.

## Oefening 2:

Drie studenten dienen maximum 5 examens af te leggen.

Per student wordt ingegeven: aantal afgelegde examens en de punten van die examens (op 20) (invoer controleren).

Een student is geslaagd indien hij op elk afgelegd examen 10 of meer heeft behaald en indien zijn percentage 60 of meer is.

Per student wordt afgebeeld: zijn punten, zijn percentage en “Geslaagd of “Niet geslaagd”.

Het gemiddelde van de percentages van de 3 studenten wordt ook weergegeven. Het gemiddelde wordt steeds weergegeven met twee cijfers na de komma.

Schrijf hiervoor een applicatie. De uitvoer wordt weergegeven in een JTextArea. Stockeer de punten en de percentages in afzonderlijke arrays.

```
// Examen.java

// Java core packages
import java.awt.Font;
import java.text.DecimalFormat;

// Java extension packages
import javax.swing.*;

public class Examens
{
    private final static int aantalExamens = 5;

    public static void main(String args[])
    {
        JTextArea outputArea = new JTextArea();
        outputArea.setFont(new Font("Monospaced",Font.PLAIN, 12));

        int aantalStudenten = 3;
        int examen[][];
        examen = new int[aantalStudenten][];
        invoer(examen);

        int percentages[] = new int[aantalStudenten];
        percentageBerekenen(examen, percentages);

        String result = maakUitvoer(examen,percentages);
        outputArea.setText(result);

        double gemPercentage = gemPercentages(percentages);
        DecimalFormat twoDigits = new DecimalFormat("0.00");
        outputArea.append("\ngemiddelde : " + twoDigits.format(gemPercentage));

        JOptionPane.showMessageDialog(null, outputArea, "Resultaten",
        JOptionPane.PLAIN_MESSAGE);
    }
}
```

```

        System.exit(0);

    } // einde methode main

    static void invoer(int array[][])
    {
        String kolomS;
        int kolom;
        for (int rij = 0; rij < array.length; rij++)
        {
            do
            {
                kolomS =
                JOptionPane.showInputDialog("Hoeveel examens heeft de student " + (rij+1) + " afgelegd : ");
                kolom = Integer.parseInt(kolomS);

                } while (kolom <= 0 || kolom > aantalExamens);
                array[rij] = new int[kolom];
                invoerPunten(array[rij]);
            }
        } //einde methode invoer

    static void invoerPunten(int array[])
    {
        String puntenS;
        int punten;
        for (int kolom = 0; kolom < array.length; kolom++)
        {
            do
            {
                puntenS =
                JOptionPane.showInputDialog("Examen " + (kolom + 1) + " : ");
                punten = Integer.parseInt(puntenS);
                } while (punten < 0 || punten > 20);
                array[kolom] = punten;
            }
        } //einde methode invoerPunten
    }

```

```

static void percentageBerekenen(int examen[][], int percentages[])
{
    int som;

    for (int rij=0;rij<examen.length;rij++)
    {
        som = 0;
        for (int kolom=0;kolom<examen[rij].length;kolom++)
            som+=examen[rij][kolom];
        percentages[rij]=(100*som)/(examen[rij].length*20);
    }
} //einde methode percentageBerekenen

```

```

static String maakUitvoer(int examen[][], int percentages[])
{
    String result="";
    boolean geslaagd;
    for (int rij=0;rij<examen.length;rij++)
    {
        result += "Student " + (rij+1);
        geslaagd = true;
        for (int kolom=0;kolom<examen[rij].length;kolom++)
        {
            result += "  " + examen[rij][kolom];
            if (examen[rij][kolom] < 10)
                geslaagd = false;
        }
        result += "\npercentage : " + percentages[rij];
        if (percentages[rij] < 60 || !(geslaagd))
            result += "  Niet geslaagd\n";
        else
            result += "  Geslaagd\n";
    } //end for
    return result;
} //einde methode maakUitvoer

```

```

static double gemPercentages(int percentages[])
{
    int som = 0;
    for (int rij=0;rij<percentages.length;rij++)
    {
        som += percentages[rij];
    }
    return (double)som / percentages.length;
} //einde methode gemPercentages

```

```

} // einde klasse Examens

```

### Oefening 3:

In een klein vliegtuig zijn er 8 rijen en 4 zetels per rij. De reiziger mag een gewenste rij (1 t.e.m. 8) en zetel (1 t.e.m. 4) opgeven.

Als de zetel bezet is, dan wordt er een overzicht afgedrukt van de plaatsing in het vliegtuig en wordt in de statusbalk een bericht weergegeven.

Als de zetel niet bezet is wordt ze gereserveerd en wordt in de statusbalk een bericht weergegeven.

Foutboodschappen verschijnen ook in de statusbalk.

Schrijf hiervoor een applet.

```
// Vliegtuig.java

// Java core packages
import java.awt.*;
import java.awt.event.*;
import java.text.*;

// Java extension packages
import javax.swing.*;

public class Vliegtuig extends JApplet implements ActionListener
{

    private JLabel rijLabel, kolomLabel;
    private JTextField rijField, kolomField;
    private JTextArea outputArea;
    private JButton reservatieButton;
    private boolean reservatie[][];

    public void init()
    {
        Container container = getContentPane();
        container.setLayout(new FlowLayout());

        rijLabel = new JLabel("gewenste rij : ");
        container.add(rijLabel);

        rijField = new JTextField(10);
        container.add(rijField);

        kolomLabel = new JLabel("gewenste zetel : ");
        container.add(kolomLabel);

        kolomField = new JTextField(10);
        container.add(kolomField);

        reservatieButton = new JButton("Reserveer");
```

```

    reservatieButton.addActionListener(this);
    container.add(reservatieButton);

    outputArea = new JTextArea(9,40);
    outputArea.setFont(new Font("Monospaced",Font.PLAIN, 12));
    container.add( outputArea);

    reservatie = new boolean[8][4];
} // einde methode init

public void actionPerformed(ActionEvent actionEvent)
{
    if (rijField.getText().length() == 0 ||
        kolomField.getText().length() == 0)
        showStatus("rij en zetel ingeven");
    else
    {
        int rij = Integer.parseInt(rijField.getText());
        int zetel = Integer.parseInt(kolomField.getText());
        String result;

        if (rij < 0 || rij > reservatie.length)
            showStatus("rij moet liggen tussen 1 en " + reservatie.length);
        else
            if (zetel < 0 | zetel > reservatie[--rij].length)
                showStatus("zetel moet liggen tussen 1 en " +
                    reservatie[rij].length);
            else
                if (reservatie[rij][--zetel])
                {
                    result = maakToonPlaatsen(reservatie);
                    outputArea.setText(result);
                    showStatus("deze ingegeven plaats is "+
                        "reeds gereserveerd");
                }
                else
                {
                    reservatie[rij][zetel] = true;
                    showStatus("de ingegeven plaats is gereserveerd");
                }
            }
    }
} //einde methode actionPerformed

```



```

public String maakToonPlaatsen(boolean[][] array)
{
    String result = "\n";
    for (int rij = 0; rij < array.length; rij++)
    {
        for (int kolom = 0; kolom < array[rij].length; kolom++)
        if (array[rij][kolom])
            result += " Bezet ";
        else
            result += " Vrij ";
        result += "\n";
    }
    return result;
} //einde methode maakToonPlaatsen

} // einde klasse Vliegtuig

```