

```
// Lösningar till gammal tentamen
```

```
// Uppgift 1 a
```

```
/*
```

Felet beror på att det är klassen på referensen som avgör vilka metoder man får köra på ett objekt, inte klassen på objektet.

På rad 5 i klassen C har referensen x klassen A, trots att den i detta specifika fall egentligen

refererar till ett objekt av klassen B. I klassen A finns inte metoden b och därför får vi felmeddelande

när vi försöker köra x.b().

Lösningen på detta är att skriva om rad 5 i klassen C så att den faktiskt ändrar referensens klass

med en typomvandling:

```
System.out.println(((B) x).b());
```

```
*/
```

```
// Uppgift 1 b
```

```
/*
```

raden med l1: new List() är fel eftersom List är ett gränssnitt

raden med l3: new List() är fel eftersom List är ett gränssnitt

raden med l4: det är olika typer till vänster och höger om likhetstecknet

raden med l6: new Collection() är fel eftersom Collection är ett gränssnitt

raden med l7: det är olika typer till vänster och höger om likhetstecknet (ingen downcasting)

```
*/
```

```
// Uppgift 2
```

```
public class Personnummer {
    private String nr;
```

```
    private int toInt(String str, int i) {
        if (Character.isDigit(str.charAt(i)))
            return Integer.parseInt(str.substring(i, i+1));
        else
            throw new IllegalArgumentException();
    }
}
```

```
public Personnummer(String s) {
    // Kontrollera att numret är korrekt
    int k = s.indexOf('-');
    if (k != 6 || s.length() != 11)
        throw new IllegalArgumentException();
    String t = s.substring(0, 6) + s.substring(7); // ta bort tecknet '-'
    // beräkna kontrollsumman
    int sum = 0;
    for (int i=0; i<9; i++) {
        int tal = toInt(t, i);
        int j = tal * (2-i%2); // multiplicera med 2 eller 1
        sum += j/10 + j%10; // addera siffrorna i resultatet till summan
    }
    sum %= 10;
    if ((toInt(t, 9) + sum) % 10 != 0)
        throw new IllegalArgumentException();
    // Korrekt!
    nr = s;
}
}
```

```
public String toString() {
    return nr;
}
}
```

```
public boolean ärKvinnan() {
    return toInt(nr, 9) % 2 == 0;
}
}
```

I gammal 2. txt

```
}

public boolean ärMan() {
    return !ärKvinnan();
}

// Uppgift 3 a

public class Abonnement {
    private String namn;
    private double perMån;
    private double perMånDag;
    private double perMånÖvr;
    private double friaDag;
    private double friaÖvr;

    // Konstruktor
    public Abonnement(String na, double pMån, double pMånDag,
        double pMånÖvr, double frDag, double frÖvr) {
        if (na == null || pMån < 0 || pMånDag < 0 ||
            pMånÖvr < 0 || frDag < 0 || frÖvr < 0)
            throw new IllegalArgumentException();
        namn = na; perMån = pMån; perMånDag = pMånDag;
        perMånÖvr = pMånÖvr; friaDag = frDag; friaÖvr = frÖvr;
    }

    // Metoder
    public String avläsNamn() {
        return namn;
    }

    public double totKostnad(double miDag, double miÖvr) {
        if (miDag < 0 || miÖvr < 0)
            throw new IllegalArgumentException();
        double antalDag = Math.max(miDag - friaDag, 0);
        double antalÖvr = Math.max(miÖvr - friaÖvr, 0);
        return perMån + antalDag * perMånDag + antalÖvr * perMånÖvr;
    }
}

// Uppgift 3 b

import javax.swing.*;
import java.util.*;
import java.io.*;

public class JfrAbonnement {
    public static void main(String[] arg) throws IOException {
        // skapa fältet med abonneringar
        Abonnement[] a = new Abonnement[100];
        Locale.setDefault(Locale.US); // decimalpunkter i filen
        Scanner sc = new Scanner(new File("mobildata.txt"));

        int n = 0; // antal olika abonneringar
        while (sc.hasNext())
            a[n++] = new Abonnement(sc.next(), sc.nextDouble(),
                sc.nextDouble(), sc.nextDouble(),
                sc.nextDouble(), sc.nextDouble());

        // Gör beräkningar för olika abonneringar
        while (true) {
            String s = JOptionPane.showInputDialog
                ("Hur många minuter ringer du per månad\n" +
                    "under dagtid respektive övrig tid?");
            if (s == null)
                break;
        }
    }
}
```

I gammal 2. txt

```
        break;
    Scanner sc2 = new Scanner(s);
    double minDag = sc2.nextDouble();
    double minÖvr = sc2.nextDouble();
    double billigast = Double.MAX_VALUE; // hittills
    int index = 0; // för det billigaste alternativet
    for (int i=0; i<n; i++) {
        // beräkna total kostnad per månad
        double kostnad = a[i].totKostnad(minDag, minÖvr);
        if (kostnad < billigast) {
            billigast = kostnad;
            index = i;
        }
    }
    // visa den lägsta kostnaden
    JOptionPane.showMessageDialog(null,
        a[index].avläsNamn() + " är billigast.\n" +
        "Kostnad: " + billigast + " kr / månad");
}
System.exit(0);
}
```

// Uppgift 4

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Rotate extends JFrame {

    public Rotate() {
        add(new DiskPanel(100, Color.red, 10));
        setSize(200, 200);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }

    public static void main(String[] args) {
        Rotate d = new Rotate();
    }
}

class DiskPanel extends JPanel implements ActionListener {
    private int maxDiameter; // Skivans höjd
    private Color color; // Skivans färg
    private int wait; // Väntetid mellan bilder
    private int diameter; // Synlig bredd
    int warp = 2;

    public DiskPanel(int size, Color color, int wait) {
        Timer t = new Timer(10, this);
        maxDiameter = size;
        this.wait = wait;
        this.color = color;
        setBackground(Color.white);
        diameter = maxDiameter / 2;
        t.start();
    }

    public void actionPerformed(ActionEvent e) {
        if (diameter <= 0 || diameter >= maxDiameter)
            warp = -warp;
        diameter += warp;
        repaint();
    }

    public void paintComponent(Graphics g) {
```

I gammal 2. txt

```
super.paintComponent(g);
int width = getSize().width;
int height = getSize().height;
g.setColor(color);
g.fillOval(width/2 - diameter/2, height/2 - maxDiameter/2, diameter,
maxDiameter);
}
}
```

// Uppgift 5

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
```

```
public class Push extends JFrame implements ActionListener {
    JButton[] b;
    JLabel lab = new JLabel("Poäng: 0");
    JButton nytt = new JButton("Nytt spel");
    javax.swing.Timer t1 = new javax.swing.Timer(30000, this); // 30 sek
    javax.swing.Timer t2;
    int poäng, aktiv, antal;

    public Push(int n, int tid) {
        antal = n;
        t2 = new javax.swing.Timer(tid, this);
        JPanel p1 = new JPanel(), p2 = new JPanel();
        setLayout(new GridLayout(2, 1));
        add(p1); add(p2);
        b = new JButton[antal];
        for (int i=0; i<antal; i++) {
            b[i] = new JButton("Knapp " + (i+1));
            b[i].setForeground(Color.red);
            b[i].setEnabled(false);
            b[i].addActionListener(this);
            p1.add(b[i]);
        }
        p2.add(lab); p2.add(nytt);
        nytt.addActionListener(this);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        pack();
        setVisible(true);
    }

    private void nyttSpel() {
        poäng = 0;
        lab.setText("Poäng: 0");
        aktiv = (int) (Math.random() * antal);
        b[aktiv].setEnabled(true);
        t1.restart(); t2.restart();
    }

    private void stoppa() {
        t1.stop(); t2.stop();
        b[aktiv].setEnabled(false);
    }

    private void bytKnapp() {
        b[aktiv].setEnabled(false);
        aktiv = (aktiv + (int) (Math.random() * (antal - 1)) + 1) % antal;
        b[aktiv].setEnabled(true);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == t1)
            stoppa();
        else if (e.getSource() == t2)
```

I gammal 2. txt

```
        bytKnapp();
    else if (e.getSource() == nytt)
        nyttSpel();
    else if (e.getSource() == b[aktiv]) {
        Toolkit.getDefaultToolkit().beep();
        lab.setText("Poäng: " + ++poäng);
        bytKnapp();
        t2.restart();
    }
}

public static void main(String[] arg) {
    new Push(Integer.parseInt(arg[0]), Integer.parseInt(arg[1]));
}
}
```