

KÄYTTÖLIITTYMÄTEKNIIKAT

TENTTI 25.1.2010

Kaikista tehtävistä saa 0 – 6 pistettä. Tentin läpäisemiseksi pitää saada puolet pisteistä. Syntaksiin liittyvät pienet yksityiskohtavirheet eivät ole merkittäviä.

- (a) Tilakoneajattelua voi soveltaa GUI-sovellusten tekemisen yhteydessä. Miten? Mitä hyötyä siitä on? Mitä ovat tilat? Entä siirtymät? (3p)

(b) Mitä tarkoittaa Model-View-Controller GUI-komponenttien yhteydessä? Miten se esiintyy Javan (AWT & Swing) GUI-komponenteissa? Entä Java-pohjaisissa GUI-sovelluksissa? (3p)
- Mikä on tapahtumankäsittelijä? Millaisena se esiintyy eri alustojen/kirjastojen (esim. Javan, xhtml:n, ...) yhteydessä? Miten se toimii Javan yhteydessä? Mitä sellaisen toteuttaminen Javan yhteydessä tarkoittaa? Pohdi erityisesti erilaisia tapoja toteuttaa tapahtumankäsittelijä – mitä etuja ja haittoja niihin liittyy? (6p)
- Oheinen ohjelma 'Minefield' (jatkuu paperin kääntöpuolella) kääntyy Java-kääntäjällä ongelmitta, ja se myös toimii käytännössä. Ohjelmassa on kuitenkin runsaasti huonoja teknisiä toteutusratkaisuja. Kerro, mitä ongelmia ohjelmassa on? (6p)

```
import java.awt.*;
import java.awt.event.*;
public class Minefield extends Frame {
    private boolean[] mines = new boolean[20];
    private int solved = 0;
    private Panel p = new Panel();
    public static void main(String[] args) { new Minefield(); }
    Minefield() {
        super("Minefield"); p.setLayout(new GridLayout(5,4));
        for (int i=0; i<20; i++) {
            Button b = new Button(""+(i+1)); p.add(b);
            b.addActionListener(new ActionListener(){
                public void actionPerformed(ActionEvent e) {
                    Button b1 = (Button)(e.getSource());
                    int target = Integer.parseInt(b1.getLabel().trim());
                    if (mines[target-1]) new GameOverNewTry(Minefield.this,"Sad news ...",
                        "Mine! You died! Another one?!?");
                    else {
                        solved++; b1.setVisible(false);
                        if (solved ≥ 10) new GameOverNewTry(Minefield.this,
                            "Great news ...", "You did it! Another one?");
                    }
                }
            });
            mines[i] = (Math.random() < 0.15);
        }
        add(p); addWindowListener(new IkkunanSulkija()); pack(); setVisible(true);
    }
}
```

```

} // Minefield()

public void restartGame() {
    for (int i=0; i<20; i++) {
        mines[i] = (Math.random() < 0.15);
        p.getComponent(i).setVisible(true);
    }
    solved = 0;
}

class GameOverNewTry extends Dialog {
    GameOverNewTry(Frame f, String s1, String s2) {
        super(f, s1, true); setSize(150,75);
        add(new Label(s2), BorderLayout.NORTH);
        Button b1 = new Button("Yes");
        Button b2 = new Button("No");
        add(b1, BorderLayout.WEST); add(b2, BorderLayout.EAST);
        b1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                restartGame(); setVisible(false); dispose(); });
        b2.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent e) {
                dispose(); System.exit(0); });
        pack(); setVisible(true);
    } //
} // GameOverNewTry
} // Minefield

class IkkunanSulkija extends WindowAdapter {
    public void windowClosing(WindowEvent e)
        { e.getWindow().dispose(); System.exit(0); }
} // IkkunanSulkija

```