

KÄYTTÖLIITTYMÄTEKNIIKAT

TENTTI 14.12.2009

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

- (a) Anonyymillä sisäluokalla voidaan toteuttaa tapahtumankäsittelijöitä. Anna esimerkki selaisesta ja kerro, milloin tapahtumankäsittelijän toteuttaminen anonyymin sisäluokan avulla on järkevää? Miten sellainen tapahtumankäsittelijä voi vaikuttaa toimintaympäristöön? (3p)

(b) Mitä tarkoittaa Model-View-Controller GUI-komponenttien yhteydessä? Miten se esiintyy Javan (AWT & Swing) GUI-komponenteissa? Entä Java-pohjaisissa GUI-sovelluksissa? (3p)
- (a) Tilakoneajattelua voi soveltaa GUI-sovellusten tekemisen yhteydessä. Miten? Mitä hyötyä siitä on? Mitä ovat tilat? Entä siirtymät? (3p)

(b) Mitä tarkoittaa ohut käyttöliittymäasiakas (thin client)? Mitkä ovat kyseisen teknisen konseptin suurimpia etuja/ongelmia? (3p)
- Tee GUI-sovellus, joka koostuu neljästä komponentista. Ylhäällä on tekstikenttä, alhaalla on toinen tekstikenttä. Keskellä on rinnakkain kaksi nappulaa, joissa tekstit "Heippa" ja "Moikka". Liitä ylhäällä olevaan tekstikenttään toiminnallisuus, jossa returnin painaminen siirtää ylhäällä olevan tekstikentän sisällön alhaalla olevan tekstikentän uudeksi sisällöksi. Tee nappuloiden painamisen kohdalla samoin, mutta tulosta nyt nappuloiden tekstisisältö alhaalla olevaan tekstikenttään. Lopeta sovellus ikkunan sulkemisen myötä. Tee edellä kuvattu ohjelma. (6p)

Seuraavassa lyhyt kuvaus neljästä luokasta. **(Jatkuu paperin kääntöpuolella.)**

```
Frame()
Frame(String title)
String getTitle()
void setTitle(String title)
void setLayout(LayoutManager mgr)
Image getIconImage()
void setIconImage(Image image)
MenuBar getMenuBar()
void setMenuBar(MenuBar mb)
boolean isResizable()
void setResizable(boolean r)
void setState(int state)
int getState()
void remove(MenuComponent m)
static Frame[] getFrames()
```

Taulukko 1: Luokan Frame konstruktoreita ja metodeja.

```

Button()
Button(String t)
String getLabel()
void setLabel(String t)
void addActionListener(ActionListener l)
void removeActionListener(ActionListener l)
void setActionCommand(String c)
String getActionCommand()

void setText(String t)
String getText()
String getSelectedText()
int getSelectionStart()
int getSelectionEnd()
boolean isEditable()
void setEditable(boolean b)
void select(int s, int e)
void selectAll()
void setCaretPosition(int p)
int getCaretPosition()
void addTextListener(TextListener l)
void removeTextListener(TextListener l)

```

Taulukko 2: Luokan `TextComponent` metodeja.

```

TextField()
TextField(String t)
TextField(int cols)
int getColumns()
void setColumns(int cols)
char getEchoChar()
void setEchoChar(char c)
boolean echoCharIsSet()
void addActionListener(ActionListener l)
void removeActionListener(ActionListener l)

```

Taulukko 3: Luokan `TextField` konstruktoreita ja metodeja.

`BorderLayout()` — Konstruktori, ei tilaa komponenttien välissä.
`BorderLayout(int hgap, int vgap)` — Konstruktori, jonka yhteydessä komponenttien reunoille määritellään tyhjää `gap`-arvojen verran.
 * Luokkavakiot `SOUTH`, `NORTH`, `CENTER`, `EAST`, `WEST`.

Taulukko 4: Luokan `BorderLayout` konstruktoreita ja metodeja.
