

OHJ-7400 Graafisen käyttöliittymän ohjelmointi

Juha-Matti Vanhatupa, ei laskimia

Tentti 17.12.2010

1. Selitä mitä ovat (1p/kohta)?
 - a. qss-tiedosto
 - b. Hildon
 - c. MeeGo
 - d. MetaObjectSystem
 - e. reentrant-luokka
 - f. UIC-kääntäjä

2. a) Mitä menetelmää käyttäen tapahtumapohjaista ohjelmointia toteutetaan seuraavissa ympäristöissä? Kerro lyhyesti. (1p/kohta)
 - i) Qt
 - ii) MFC
 - iii) GTK

b) Millä tavoin Qt:ssa voidaan estää käyttöliittymän jumiutumista? (3p)

3. a) Kerro mitkä ovat Qt:n model/view-mallin osat, ja selitä niiden tehtävät. (3p)

b) Kerro miten Qt:n model/view-mallin osat kommunikoivat. (3p)

4. Alla on listattuna aloittelevan Qt -ohjelmoijan koodia. Koodissa on kuitenkin lukuisia virheitä. Selosta koodin virheet: kerro miksi nuo kohdat ovat virheellisiä ja kerro miten korjaisit ne. Ohjelmoijan tarkoituksena oli piirtää kymmenen ympyrää. (6p)

```
// mainwindow.h
#ifdef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QPainter>

namespace Ui {
```

HUOM: Käännä

```

        class MainWindow;
    }

class MainWindow : public QMainWindow
{
    Q_object

public:
    explicit MainWindow(QWidget *parent = 0);
    ~MainWindow();

protected:
    void paintEvent(QPaintEvent *event);
private:
    Ui::MainWindow *ui;
};

#endif // MAINWINDOW_H

// mainwindow.cpp
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::paintEvent(QPaintEvent *event)
{
    QPainter paint;
    for (int i = 0; i < 10; ++i) {
        paint.drawEllipse(40+i*10,40+i*10,200,200);
    }
    repaint();
}

```