

# QT-ohjelmointi C++:lla

Tässä muutama yksinkertainen esimerkki Qt-grafiikkakirjaston käytöstä. Jos käytät Debiania tai Ubuntu (universe), käyttämäni Kdevelop-kehitysympäristö löytyy suoraan paketinhallinnasta nimellä kdevelop3.

## 1. tetrawidget.cpp

```
#include <qpoint.h>
#include <qfont.h>
#include <qfontdialog.h>

#include "tetrawidget.h"

using namespace std;

tetraWidget::tetraWidget(QWidget* parent, const char* name, WFlags fl)
    : tetraWidgetBase(parent,name,fl)
    , piirrin(label)
{
    // Matriisi ja väri
    vakiokulma.rotate(5); // Muodostetaan kallistusmatriisi
    piirrin.setPen( Qt::red );

    // Piirtojutut reilaan
    piirrin.save();
    skaalaa();

    // Ajastin käyntiin
    connect(&ajastin,SIGNAL(timeout()),SLOT(kelloTapahtuma()));
    ajastin.start(50);
}

tetraWidget::~tetraWidget()
{
    piirrin.restore();
}

/*$SPECIALIZATION$*/
void tetraWidget::skaalaa()
{
    // Matriisi talteen
    piirrin.restore();
    piirrin.save();
```

```
// Piirtolaaaatikon ja piirtimen sijainti
tekstiloota = label->rect();
tekstiloota.moveBy(-tekstiloota.width()/2, -tekstiloota.height()/2);
piirrin.translate(tekstiloota.width()/2, tekstiloota.height()/2);

// Fontin koko
QFont fontti = piirrin.font();
fontti.setPointSize(label->width() / 10);
piirrin.setFont(fontti);
```

```
}

void tetraWidget::kelloTapahtuma( )
{
    static QString viesti = "Osta kissa!\nVain 13 euroa.";

    piirrin.eraseRect(tekstiloota);
    piirrin.setWorldMatrix(vakiokulma,TRUE);
    piirrin.drawText(tekstiloota, AlignCenter, viesti );
}

void tetraWidget::vaihdaFontti()
{
    bool ok;
    QFont font_uusi = QFontDialog::getFont(&ok, piirrin.font(), this);

    if (ok) piirrin.setFont(font_uusi);
}

#include "tetrawidget.moc"
```

## 2. tetrawidget.h

```
#ifndef _TETRAWIDGET_H_
#define _TETRAWIDGET_H_

#include "tetrawidgetbase.h"

#include <qlabel.h>
#include <qpainter.h>
#include <qtimer.h>
#include <qwmatrix.h>
#include <qrect.h>

class tetraWidget : public tetraWidgetBase
{
    Q_OBJECT
private:
    QPainter piirrin;
```

```

    QTimer ajastin;
    QWMatrix vakiokulma;
    QWMatrix kallistus;
    QRect tekstiloota;

public:
    tetraWidget(QWidget* parent = 0, const char* name = 0, WFlags fl = 0 );
    ~tetraWidget();
    /*$PUBLIC_FUNCTIONS$*/

public slots:
    /*$PUBLIC_SLOTS$*/
    virtual void skaalaa();
    virtual void vaihdaFontti();

protected:
    /*$PROTECTED_FUNCTIONS$*/

protected slots:
    /*$PROTECTED_SLOTS$*/
    virtual void kelloTapahtuma();
};

#endif

```

## 3. laskuri.cpp

```

#include "laskuri.h"
laskuri::laskuri(QWidget *parent, const char *name)
    : QLabel(parent, name)
{
    setKasvatus(1);
    setArvo(0);
    setPaletteBackgroundColor(Qt::cyan);
    setAlignment(Qt::AlignRight);
    setFrameStyle( QFrame::Panel | QFrame::Sunken );
}

laskuri::~laskuri()
{
}

void laskuri::setArvo( int arvo )
{
    FARvo = arvo;
    setNum(FARvo);
}

void laskuri::setKasvatus( int kasvatus )

```

```

{
    FKasvatus = kasvatus;
}

void laskuri::kasvata( )
{
    setArvo( arvo() + kasvatus() );
}

void laskuri::nollaa( )
{
    setArvo(0);
}

```

## 4. laskuri.h

```

#ifndef LASKURI_H
#define LASKURI_H

#include <qlabel.h>

/**
 * @author Joel Lehtonen
 */
class laskuri : public QLabel
{
    Q_OBJECT
    Q_PROPERTY( int arvo READ arvo WRITE setArvo )
    Q_PROPERTY( int kasvatus READ kasvatus WRITE setKasvatus )
public:
    laskuri(QWidget *parent = 0, const char *name = 0);
    virtual ~laskuri();
    int arvo() const { return FARvo; };
    int kasvatus() const { return FKasvatus; };
    void setArvo(int arvo);
    void setKasvatus(int kasvatus);

public slots:
    void kasvata();
    void nollaa();

private:
    int FARvo;
    int FKasvatus;
};

#endif

```