

Uživatelská rozhraní

10. cvičení

Martin Němec
martin.nemec@vsb.cz
A1047
+420 597 324 438

Úvod do Qt GUI toolkit

- Qt (cute)
 - Samostatná knihovna společnosti TrollTech.
 - Podpora platforem Windows, Linux, Mac.
 - Vzhled i chování prvků je implementováno v C++.
 - Podpora pro datové přenosy (Cut/Copy/Paste, Drag & Drop).



Prerekvizity

- **C++ kompilér, gcc, MSVC**

<http://java.sun.com/javase/downloads/index.jsp>

- **Manuály**

Specifikace API:

<http://doc.trolltech.com>

Tutoriály:

<http://doc.trolltech.com>

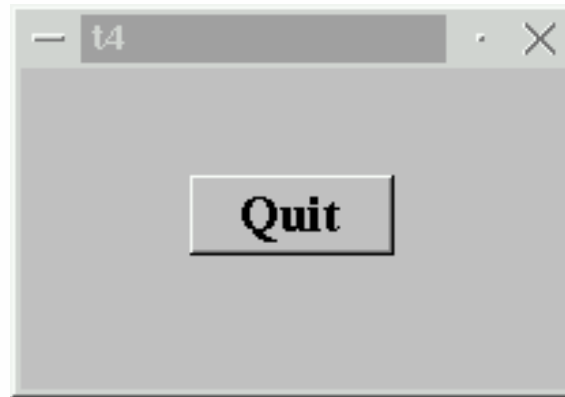
- **Editory**

MS Visual Studio, PSPad, ...



Základní komponenty

- **QApplication, QWidget**
 - Okno aplikace, obsahuje standardní prvky, komunikuje s OS.
 - Kontejnerem pro další komponenty (např. QPushButton, QLabel, QLineEdit) je buď QApplication nebo námi vytvořená třída, která dědí z QWidget.



Další komponenty

- **Atomické komponenty:**

QLabel, QPushButton, QButtonGroup, QLineEdit,
QListBox,...

Umožňují interakci s uživatelem a zobrazují informace.



Postup při použití obecné komponenty (I)

- **1. Instanciaci.**

- `QPushButton *btn = new
 QPushButton(QString("Prevod"), this);`

- **2. Konfigurace.**

- `btn.setText(QString("TLAČÍTKO"));`

- **3. Jedná-li se o kontejner, vložení potomků.**

- **4. Nejedná-li se o QApplication, vložení komponenty do layoutu** (např. QHBoxLayout, QGridLayout).

- `box->addWidget(btn);`

Postup při použití obecné komponenty (II)

- **5. Registrace posluchče.**

V definici třídy:

```
Q_OBJECT
```

```
private slots:
```

```
    void clickPrevodBtn();
```

V implementaci třídy:

```
connect( btnPrevod, SIGNAL(clicked()), this,  
        SLOT(clickPrevodBtn()) );
```

Layout Management

- LM ovládají rozmístění komponent v rámci kontejneru.
- Horizontal Layout
- Vertical Layout
- Grid Layout



Horizontal Layout

- Vkládá komponenty na řádek zleva doprava.

```
QBoxLayout *hbox = new QHBoxLayout( this );  
QPushButton btn1, btn2, btn3;
```

```
hbox->addWidget( &btn1 );  
hbox->addWidget( &btn2 );  
hbox->addWidget( &btn3 );
```



Vertical Layout

- Vkládá komponenty do sloupce shora dolů.

```
QBoxLayout *vbox = new QVBoxLayout( this );  
QPushButton btn1, btn2, btn3;
```

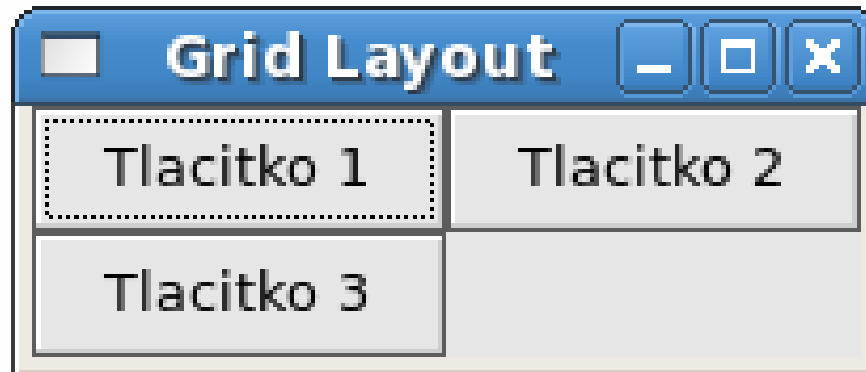
```
vbox->addWidget( &btn1 );  
vbox->addWidget( &btn2 );  
vbox->addWidget( &btn3 );
```



Grid Layout

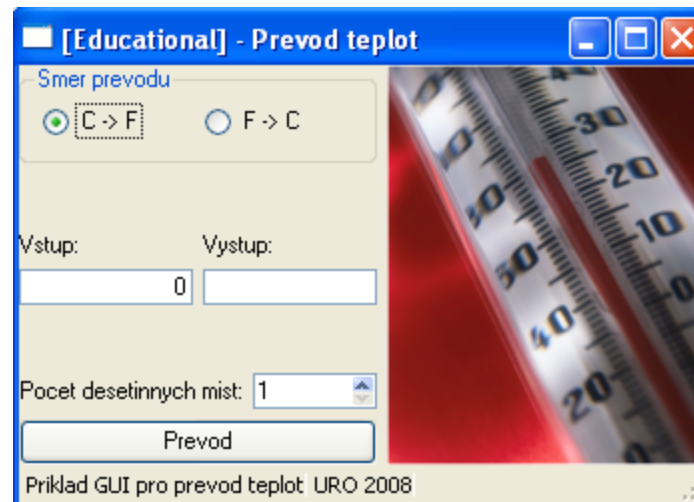
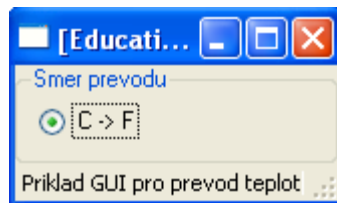
- Umísťuje komponenty do buněk mřížky obdobně jako.
- Řádky i sloupce mohou mít různou velikost.
- Komponenta může zabírat více buněk (rowspan, colspan).

```
QGridLayout *top = new QGridLayout( this, 2, 2 );  
top->addWidget( btn1, 0, 0 );
```



Praktická úloha

- Pro cvičení použijeme šablonu, na které se budeme dále seznamovat s komponentami zahrnutými v Qt API.
- Design si můžete upravit libovolně podle sebe. Pokuste se vyzkoušet základní prvky, včetně nastavení jejich vlastností.



Použití třídy QWidget

```
class MyWidget: public QWidget {  
Q_OBJECT
```

```
QLineEdit *inEd;
```

```
public:
```

```
    MyWidget( QWidget *parent=0, const char *name=0 );
```

```
private slots:
```

```
    void clickPrevodBtn();
```

```
};
```

Základní operace v konstruktoru

```
MyWidget::MyWidget( QWidget *parent, const  
    char *name )  
    : QWidget( parent, name ) {
```

```
    QVBoxLayout *vbox = new QVBoxLayout( this );
```

```
    QVBoxLayout *hbox = new QHBoxLayout( vbox, 5 );
```

```
    QVBoxLayout *levy = new QVBoxLayout( hbox );
```

```
    QString *boxName = new QString( "Smer prevodu" );
```

```
    QPushButtonGroup *smerFr = new QPushButtonGroup( 2,  
        Horizontal, *boxName, this, "smerFr" );
```

Zpracování událostí

```
MyWidget::MyWidget( QWidget *parent, const char *name )  
    : QWidget( parent, name ) {
```

```
    connect(btnPrevod, SIGNAL(clicked()), this,  
           SLOT(clickPrevodBtn()));
```

```
}
```

```
void MyWidget::clickPrevodBtn() {  
}
```



Dopracovat výpočet a vzhled.

Děkuji za pozornost.

