Uživatelská rozhraní

Knihovna Qt

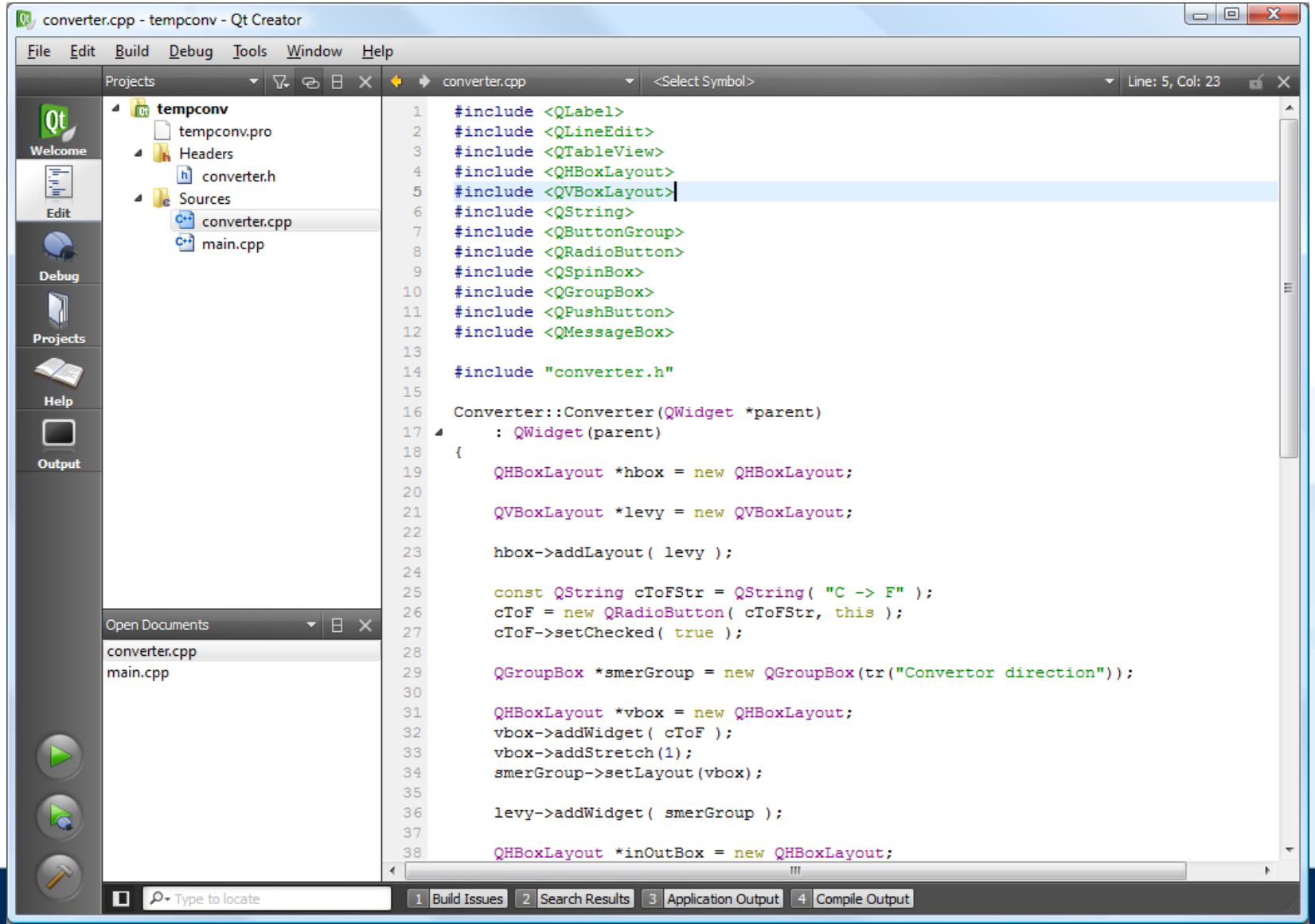
- Trolltech (1994) v Oslu (Norsko) vytváří grafické uživatelské rozhraní (GUI) pro C++
- multi-platformová GUI C++ knihovna, určena pro vývoj aplikací (Unix/X, Windows)
- Signály a sloty

Knihovna Qt

Hlavní stránky https://www.qt.io/

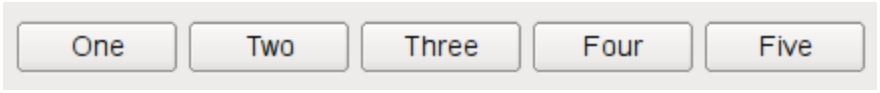
Dokumentace https://doc.qt.io/

Qt Creator

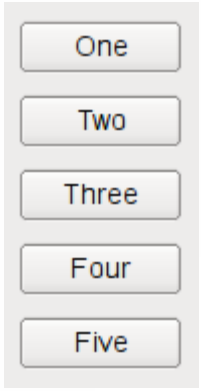

converter.cpp - tempconv - Qt Creator
File Edit Build Debug Tools Window Help
Projects
tempconv
tempconv.pro
Headers
converter.h
Sources
converter.cpp
main.cpp
Open Documents
converter.cpp
main.cpp
Welcome
Edit
Debug
Projects
Help
Output
converter.cpp
<Select Symbol>
Line: 5, Col: 23
#include <QLabel>
#include <QLineEdit>
#include <QTableView>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QString>
#include <QButtonGroup>
#include <QRadioButton>
#include <QSpinBox>
#include <QGroupBox>
#include <QPushButton>
#include <QMessageBox>
#include "converter.h"
Converter::Converter(QWidget *parent)
: QWidget(parent)
{
QHBoxLayout *hbox = new QHBoxLayout;
QVBoxLayout *levy = new QVBoxLayout;
hbox->addLayout(levy);
const QString cToFStr = QString("C -> F");
cToF = new QRadioButton(cToFStr, this);
cToF->setChecked(true);
QGroupBox *smerGroup = new QGroupBox(tr("Convertor direction"));
QHBoxLayout *vbox = new QHBoxLayout;
vbox->addWidget(cToF);
vbox->addStretch(1);
smerGroup->setLayout(vbox);
levy->addWidget(smerGroup);
QHBoxLayout *inOutBox = new QHBoxLayout;
Type to locate
1 Build Issues
2 Search Results
3 Application Output
4 Compile Output

Layout

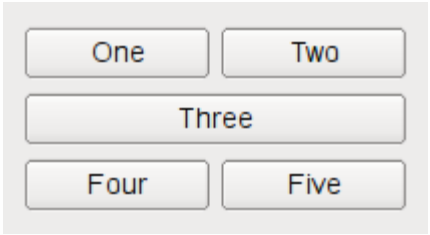
QHBoxLayout



QVBoxLayout



QGridLayout



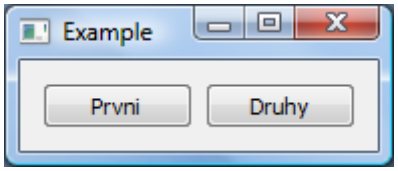
QFormLayout

https://doc.qt.io/qt-5/layout.html

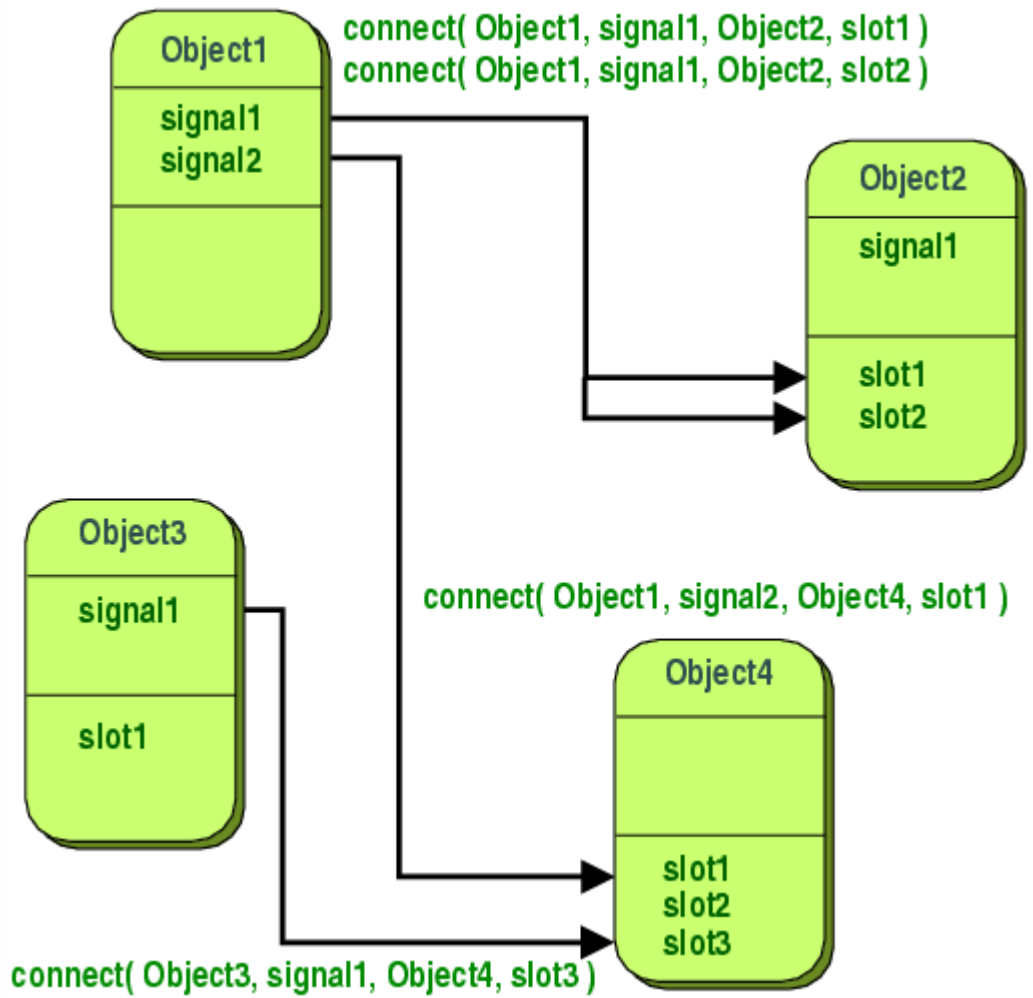
Ukázky

```
#include <QApplication>
#include <QPushButton>
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QPushButton  *button = new QPushButton("Quit");
    QObject::connect(button, SIGNAL(clicked()), &app, SLOT(quit()));
    button->show();
    return app.exec();
}
```

```
#include <QApplication>
#include <QHBoxLayout>
#include <QPushButton>
int main(int argc, char *argv[]){
   QApplication app(argc, argv);    // hlavni okno aplikace

   Qwidget *window = new QWidget;
   window->setWindowTitle("Example");

   QPushButton *prvni = new QPushButton("Prvni");
   QPushButton *druhy = new QPushButton("Druhy");

   // propojeni signalu a slotu  connect(prvni, SIGNAL(clicked()), &app,  SLOT(stiskPrvni(int)));

   QHBoxLayout *layout = new QHBoxLayout; // horizontalni rozmisteni komponent v okne aplikace
   layout->addWidget(prvni);
   layout->addWidget(druhy);
   window->setLayout(layout);
   window->show();

   return app.exec();
}
```



Signály a sloty



Signály a sloty

```
class Priklad1
    {
    public:
        Priklad1(); // konstruktor
        int hodnota() const { return _hodnota; }
        void nastavHodnotu( int );
    private:
        _hodnota val;
    };
```

Signály a sloty

```
class Priklad1 : public QObject  {
    Q_OBJECT
  public:
    Foo();
    int hodnota() const { return _hodnota; }
  public slots:
    void nastavHodnotu( int );
  signals:
    void hodnotaZmenena(int);
  private:
    int _hodnota;
  };
```

Signály a sloty

```
void Priklad1::nastavHodnotu( int h )
   {
      if ( h != _hodnota ) {
         _hodnota = h;
         emit hodnotaZmenena(h);
      }
   }
   //  signál: hodnotaZmenena
```

Signály a sloty

Priklad1 a, b; // definice dvou objektu dedicich z QObject

//prirazeni signalu

connect(&a, SIGNAL(hodnotaZmenena(int)), &b, SLOT(nastavHodnotu(int)));

b.nastavHodnotu(11);

// a == není definováno b == 11

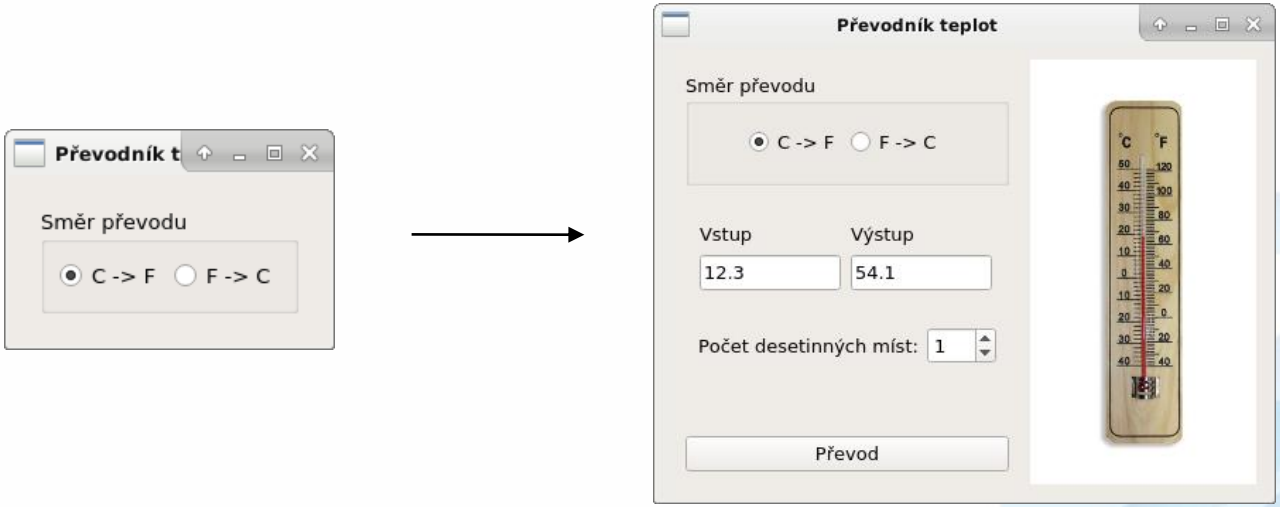
a.nastavHodnotu(79);

// a == 79 b == 79

b.hodnota();

Náplň cvičení:

- Vytvořte převodník teplot



Obrázek

```
QString  *imgFilename = new QString( "soubor.png" );
QPixmap  *imgPixmap = new QPixmap( *imgFilename );
QLabel  *obr = new QLabel;
obr->setPixmap( *imgPixmap );
hbox->addWidget( obr );
```

Prosím, dopracujte opět funkčnost a vzhled. Projekt si můžete libovolně rozšířit.

Děkuji za pozornost.