

# Podstawy programowania w Qt4

Rafał Rawicki

21 kwietnia 2007

# Co to jest Qt?

- ▶ Zestaw bibliotek dla języka C++
- ▶ oraz narzędzia rozszerzające jego możliwości
- ▶ Ponad 400 klas
- ▶ Introspekcja oraz mechanizm sygnałów i slotów - braki C++

# Obsługiwane platformy

- ▶ **Microsoft Windows** – 98, NT 4.0, ME, 2000, and XP
- ▶ **Unix/X11** – Linux, Sun Solaris, HP-UX, HP Tru64 UNIX, IBM AIX, SGI IRIX i wiele innych.
- ▶ **Mac OS X** – Mac OS X 10.3+
- ▶ **Systemy wbudowane** – Linux z obsługą framebuffera

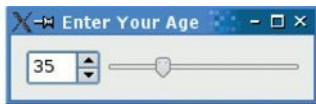
W przygotowaniu:

- ▶ **Windows Mobile i Windows CE.**

# Na każdej platformie natywny wygląd



Windows



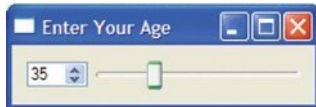
Plastique



CDE



Motif



Windows XP



Mac

# Licencje

Qt4 jest oferowane w dwóch opcjach:

- ▶ **Open Source** – Na wolnej licencji GNU GPL.
- ▶ **Commercial** – Komercyjny sposób dystrybucji z supportem.

W wersji 3 biblioteka Qt była dostępna na licencji GNU GPL tylko dla systemu Linux. Wraz z wydaniem Qt4 możemy tworzyć Wolne Oprogramowanie także pod pozostałe systemy.

# Hello World!

- ▶ Rozmieszczanie widgetów za pomocą layoutów
- ▶ Sygnały i sloty
- ▶ Qmake

# Łatwe rozmieszczanie elementów dzięki layoutom

```
1 QHBoxLayout* layout = new QHBoxLayout;  
2 layout->addWidget(exit_button);  
3 ...  
4 main_window->setLayout(layout);
```

- ▶ Programista nie wpisuje pozycji elementów na sztywno.
- ▶ Okienka wyglądają dobrze niezależnie od ustawionego rozmiaru.

## Łatwe rozmieszczanie elementów dzięki layoutom c. d.

- ▶ QHBoxLayout - rozkład w poziomie
- ▶ QVBoxLayout - rozkład w pionie
- ▶ QGridLayout - siatka

Elementy są kładzione na layoucie w kolejności wywoływania

layout ->addWidget(...).



# Mechanizm sygnałów i slotów

```
1 QObject::connect(exit_button, SIGNAL(clicked()), &app, SLOT(quit()));
```

Możemy połączyć:

- ▶ Jeden sygnał do wielu slotów
- ▶ Wiele sygnałów do tego samego slotu
- ▶ Sygnał do sygnału

Sygnały są rozłączane automatycznie w przypadku usunięcia któregoś z obiektów, ale można to zrobić ręcznie.

# Mechanizm sygnałów i slotów c. d.

Mechanizm sygnałów i slotów jest bezpieczny ze względu na typy.  
(Typy sygnału muszą zgadzać się z typami slotu.)

Wyjątek:

```
1 QObject::connect(widget1, SIGNAL(valueChanged(int, const QString&)),  
2                   widget2, SLOT(setValue(int)));
```

Jeżeli slot ma mniej parametrów, dodatkowe parametry są ignorowane.

# Ograniczenia mechanizmu sygnałów i slotów

Nie możemy tworzyć sygnałów/slotów w:

- ▶ Klasach będących szablonami
- ▶ Klasach zagnieżdżonych w innych klasach

Nie możemy również tworzyć połączeń przy pomocy `std::bind1st`, `std::bind2nd`, `boost::bind` itp.

# Qmake

- ▶ **Qmake** – Manager budowania automatycznie tworzy prawidłowy plik Makefile dla każdej platformy.

```
$ qmake-qt4 -project
```

```
$ qmake-qt4
```

```
$ make
```

# Ftp Client

- ▶ Qt Designer
- ▶ Klasy Model/View
- ▶ QNetwork
- ▶ QtLinguist

# Qt Designer

Designer jest graficznym narzędziem dostarczanym razem z Qt ułatwiającym projektowanie interfejsu naszej aplikacji.

- ▶ Designer generuje opis wyglądu w pliku xml `nazwa_klasy.ui`
- ▶ Dodajemy do pliku projektu `FORMS += nazwa_klasy.ui`
- ▶ Dołączamy plik nagłówkowy `ui_nazwa_klasy.h`
- ▶ Klasa okienka dziedziczy po `Ui::NazwaKlasy`

# Qt Designer c. d.

- ▶ Rozmieszczanie elementów interfejsu
- ▶ Łączenie sygnałów i slotów
- ▶ Buddy Widgets
- ▶ Zasoby

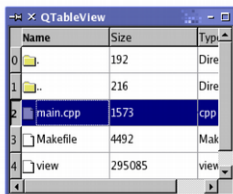
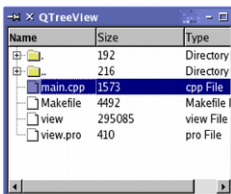
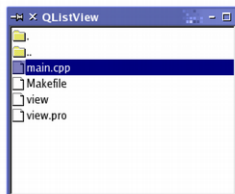
# MVC a Qt4

Implementacja popularnego wzorca projektowego.

- ▶ **Widok** – pobiera dane z modelu i prezentuje użytkownikowi aplikacji
- ▶ **Model** – pobiera dane ze źródła i udostępnia widokowi



# MVC a Qt4 - Widoki



# MVC a Qt4 - Modele

Wszystkie modele dziedziczą z klasy `QAbstractItemModel` Kilka gotowych jest dostarczanych razem z Qt:

- ▶ `QStringListModel`
- ▶ `QStandardItemModel`
- ▶ `QDirModel`
- ▶ `QSqlQueryModel`, `QSqlTableModel`,  
`QSqlRelationalTableModel`

Własne modele tworzymy dziedzicząc po klasach

`QAbstractItemModel`, `QAbstractListModel`, `QAbstractTableModel`.

# Qt Linguist i Lokalizacja aplikacji

lupdate ▷ linguist-qt4 ▷ lrelease

```
1 int main(int argc, char** argv){
2     QApplication app(argc, argv);
3     QTranslator translator;
4
5     translator.load("ftp_client_" + QLocale::system().name(),
6                   qApp->applicationDirPath());
7     app.installTranslator(&translator);
8
9     FtpClient ftp;
10    ftp.show();
11
12    return app.exec();
13 }
```

## Dalsze informacje

- ▶ <http://doc.trolltech.com>
- ▶ <http://qtcentre.org>
- ▶ <http://qtforum.org>
- ▶ [labs.trolltech.com](http://labs.trolltech.com)

# Pytania?

# Dziękuję za uwagę

mail: [rawicki@gmail.com](mailto:rawicki@gmail.com)

jid: [rawicki@chrome.pl](mailto:rawicki@chrome.pl)