

Лекция 8. GUI тулкиты

Разработал: Максимов А.Н.

Версия 2. 11/2017

Содержание

- Работа с библиотекой qt
- Open GL

Часть 1. Qt

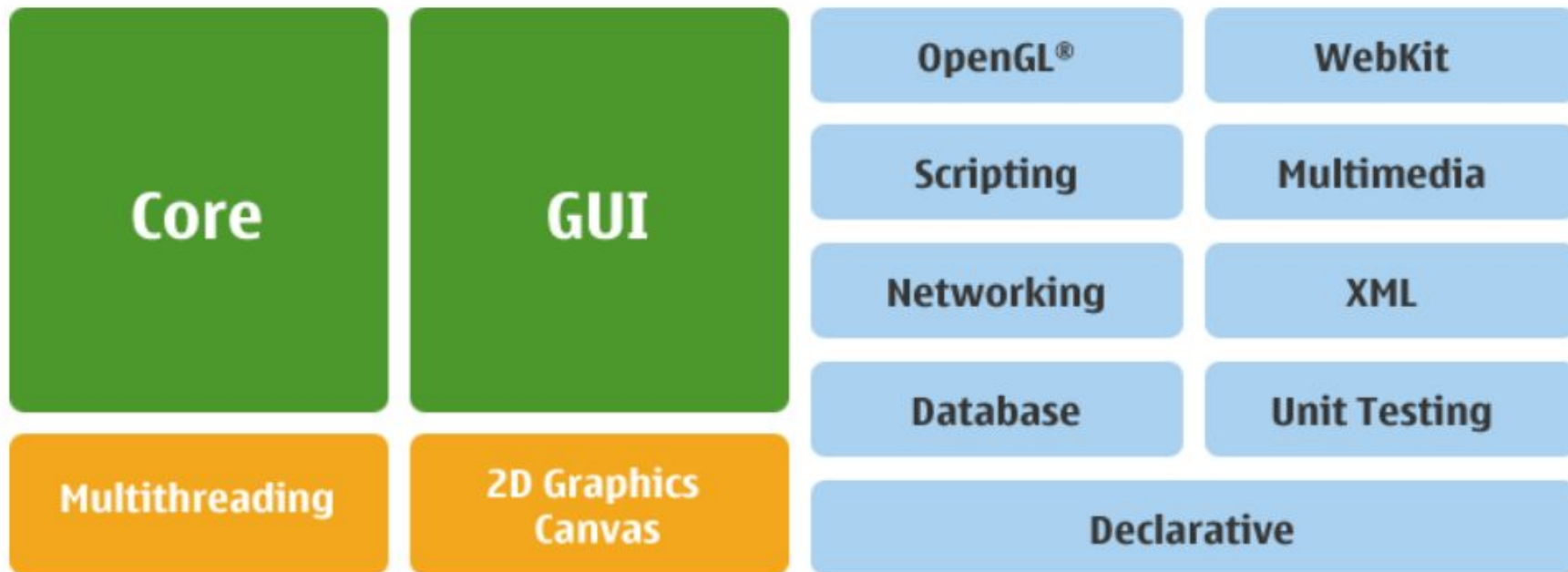
Qt – распространенная кросс-платформенная библиотека для разработки пользовательского интерфейса

Библиотека организована в виде модулей, обеспечивающих богатый набор классов для создания современных приложений.



Архитектура Qt

Основные модули библиотеки:



Подробное описание API:

<http://doc.qt.io/qt-5/>

Qt. Как установить

1. Под windows:

<http://doc.qt.io/qt-5/windows-support.html> (на английском)

2. Под linux (дистрибутив Ubuntu)

Для qt4

```
sudo apt-get install libqt4-core libqt4-gui libqt4-dev
```

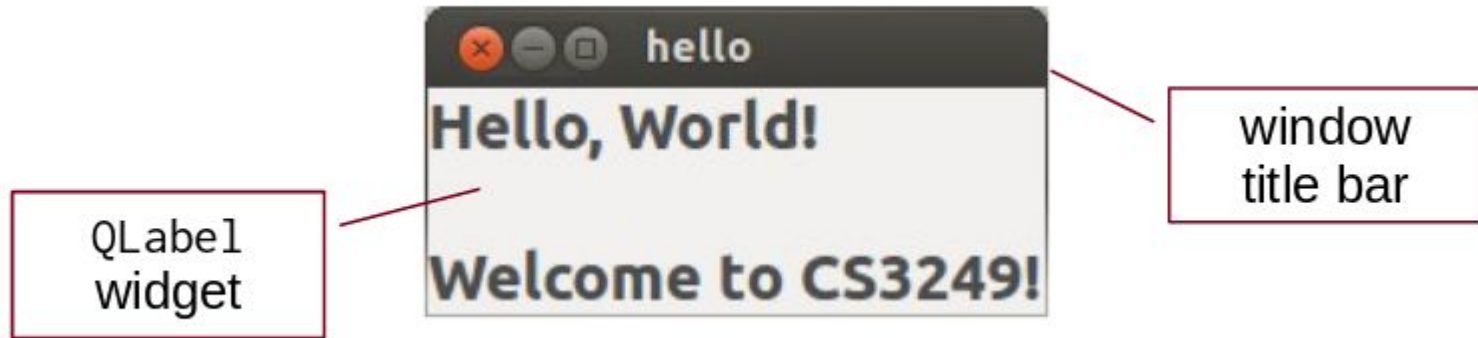
Для qt5

```
sudo apt-get install qtdeclarative5-dev qml-module-qtquick-controls
```

Qt. Hello world

```
#include <QApplication>
#include <QLabel>
int main(int argc, char *argv[ ] )
{
    QApplication app(argc, argv);
    QLabel label("Hello world!");
    label.show(); // По умолчанию виджет не отображается
    return app.exec();
}
```

Hello world – результат



Widget = элемент окна

Виджеты скрыты после создания. Необходимо настроить их до отображения

QApplication - объект программы.

app.exec() старт цикла обработки событий.

qmake

Проект на qt описывается в pro файле.
Пример hello.pro файла.

```
CONFIG += qt  
HEADERS += hello.h  
SOURCES += hello.cpp  
SOURCES += main.cpp
```


Qt. Hello world – сборка

Для того чтобы собрать исполняемый файл
необходимо выполнить следующие команды:

```
qmake -project
```

```
qmake
```

```
make
```

Qt. Сигналы

Qt добавляет в C++ понятие сигналов и слотов.

Перед тем как проект компилируется стандартным компилятором C++ (g++ например), его обрабатывает МОС компилятор, добавляя и подменяя код для соответствия ISO C++.

Сигнал — метод без реализации. Программист пишет только его прототип. Остальное делает МОС. Сигнал срабатывает при вызове: `emit signal_name();`

Qt. Слот

Слот — метод, присоединяемый к сигналу. Фактически обработчик сигнала — реакция на вызов emit.

Логику метода пишет программист.

Пример:

```
class MySlotClass : public QObject {
    Q_OBJECT
public:
    MySlotClass();
public slots:
    Myslot()
    {
        qDebug("SLOT");
    }
};
```

Qt. Соединения

Чтобы связать сигнал и слот используют метод connect класса QObject

```
connect(signaling_class,SIGNAL(its_signal()),slot_class,SLOT(slot_method));
```

пример:

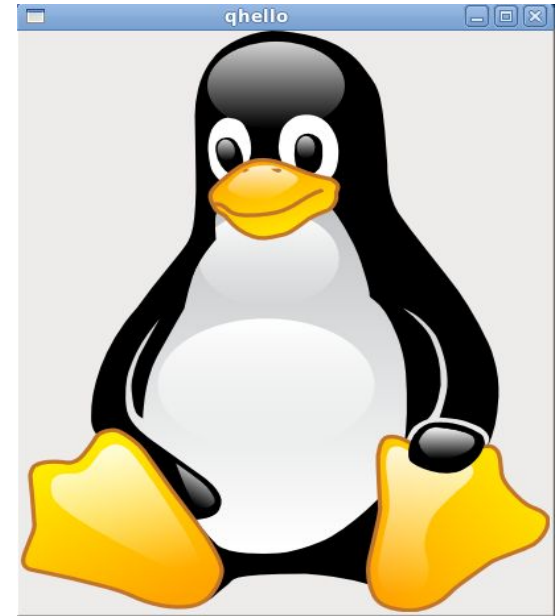
```
...  
MySlotClass example;  
QPushButton button;  
QObject::connect(&button,SIGNAL(clicked()),&example,SLOT(Myslot()));  
...  
// при каждом нажатии кнопки вызывается Myslot
```

Пример с кнопкой.

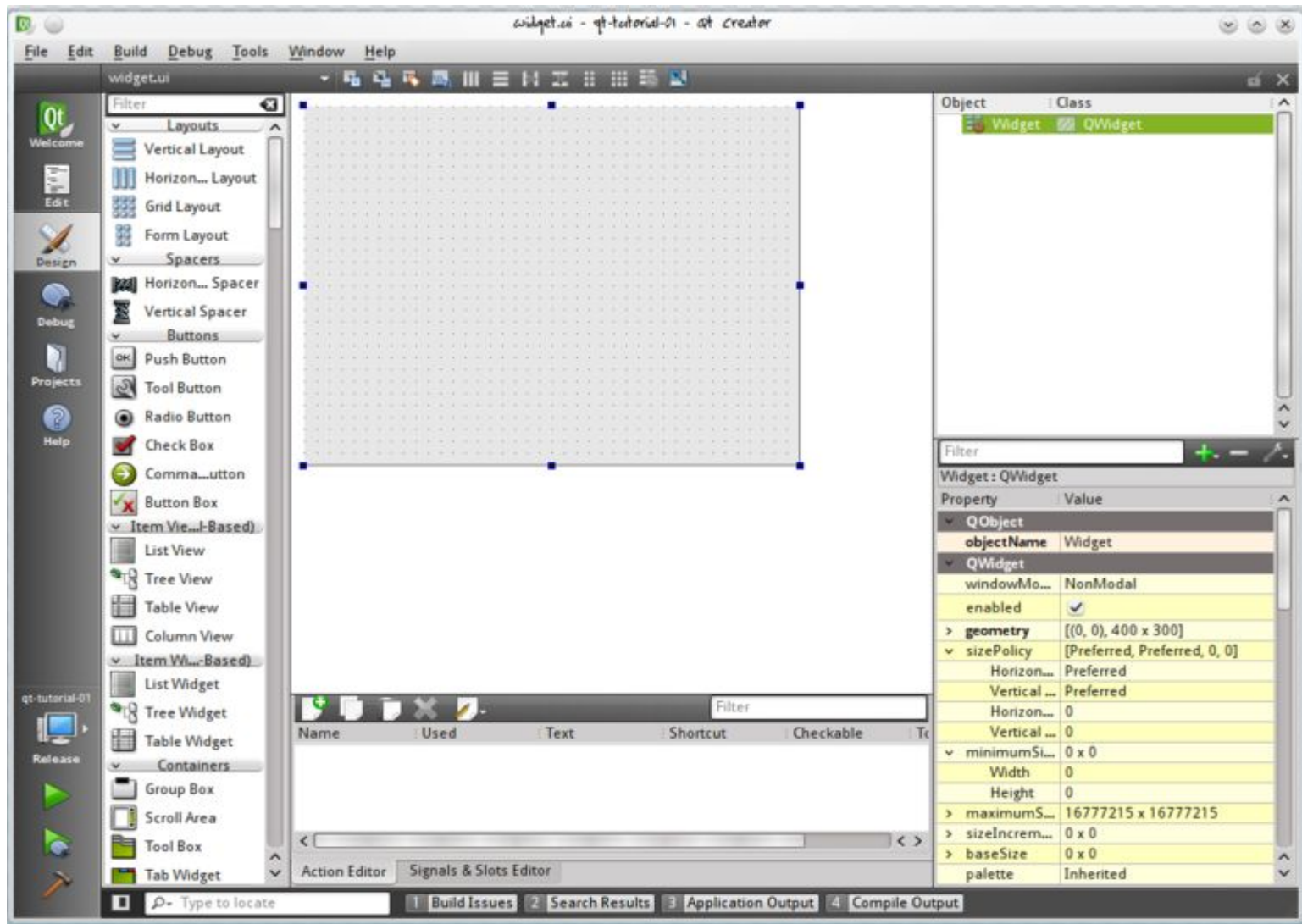
```
#include <qapplication.h>
#include <qpushbutton.h>
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QPushButton *button = new QPushButton("Quit", 0);
    QObject::connect(button, SIGNAL(clicked()),&app, SLOT(quit()));
    app.setMainWidget(button);
    button->show();
    return app.exec();
}
```

Отобразить изображение

```
#include <QtGui/QApplication>
#include <QLabel>
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QImage myImage;
    myImage.load("tux.png");
    QLabel myLabel;
    myLabel.setPixmap(QPixmap::fromImage(myImage));
    myLabel.show();
    return a.exec();
}
```



Создание интерфейса при помощи qtcreator



Pro файл

Файл qhello.pro

```
TEMPLATE = app
```

```
TARGET = name_of_the_app
```

```
QT = core gui
```

```
greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
```

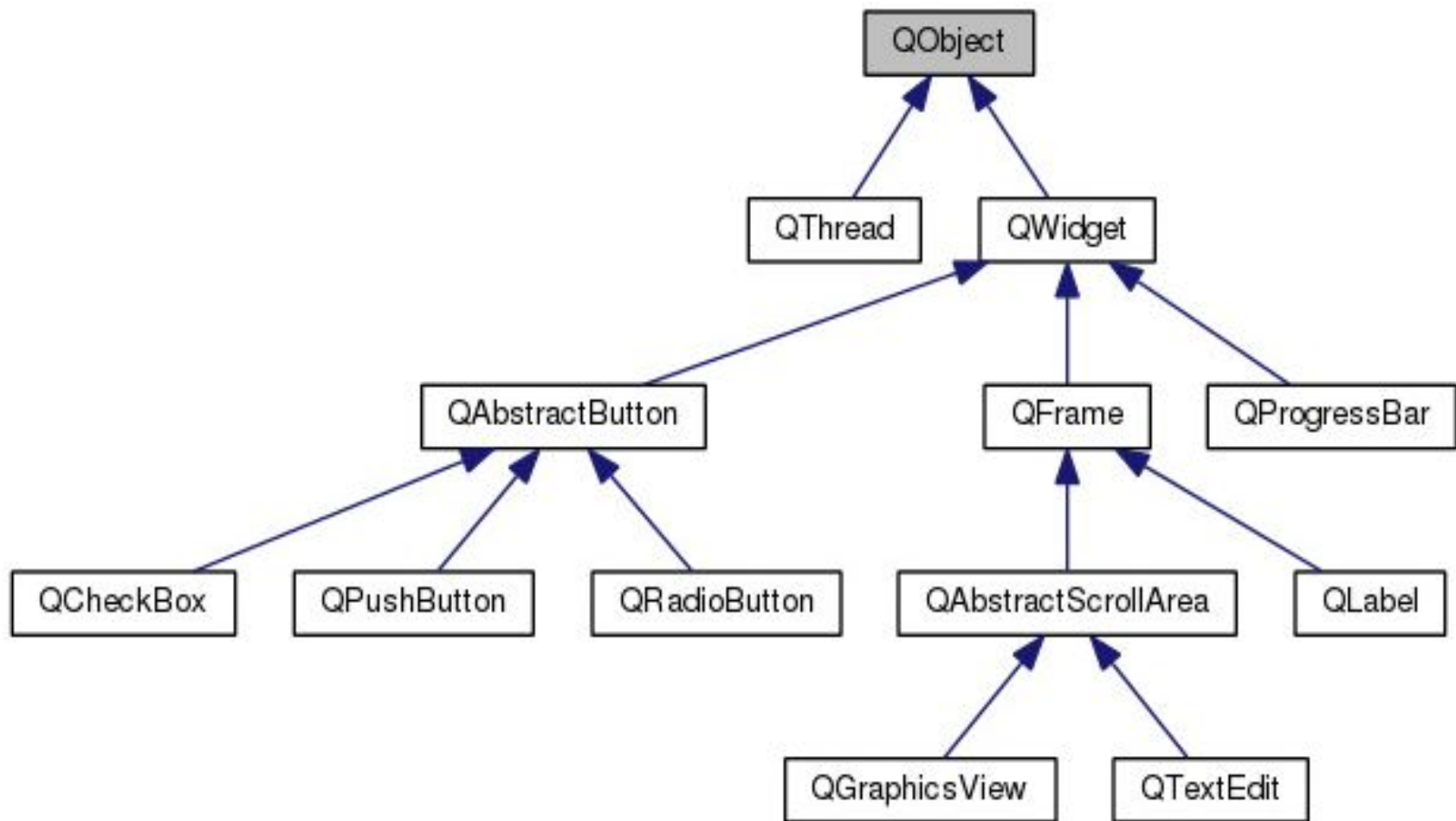
```
SOURCES += main.cpp
```

Как собрать:

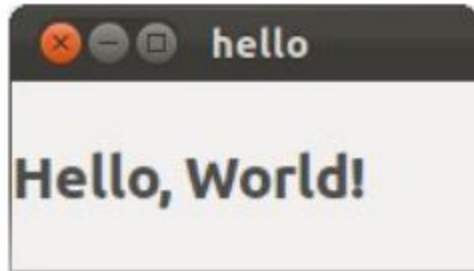
```
qmake -o Makefile qhello.pro
```

```
make
```

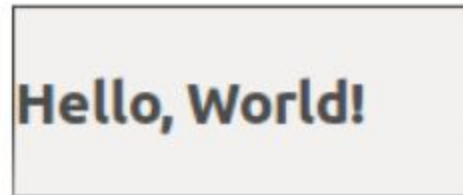

Иерархия виджетов в qt



Window и widget-ы



window



widget

Window — у окна есть заголовок.

Widget - у виджета нет.

Widget без родителя - окно.

Пример

```
#include <QApplication>
#include <QHBoxLayout>
#include <QSlider>
#include <QSpinBox>

int main(int argc, char *argv[]) {
    QApplication app(argc, argv);

    QWidget *window = new QWidget; // Create main window.
    window->setWindowTitle("Enter your age");
    QSpinBox *spinBox = new QSpinBox; // Create spin box.
    spinBox->setRange(0, 130);
    QSlider *slider = new QSlider(Qt::Horizontal); // Create slider.
    slider->setRange(0, 130);
```

Пример

```
QObject::connect(spinBox, SIGNAL(valueChanged(int)),  
slider, SLOT(setValue(int))); // Connect spin box to slider.
```

```
QObject::connect(slider, SIGNAL(valueChanged(int)),  
spinBox, SLOT(setValue(int))); // Connect slider to spin box.  
spinBox->setValue(35); // Initialise value.
```

```
QHBoxLayout *layout = new QHBoxLayout; // Create layout to put widgets in  
place.
```

```
layout->addWidget(spinBox);
```

```
layout->addWidget(slider);
```

```
// Put layout in main window.
```

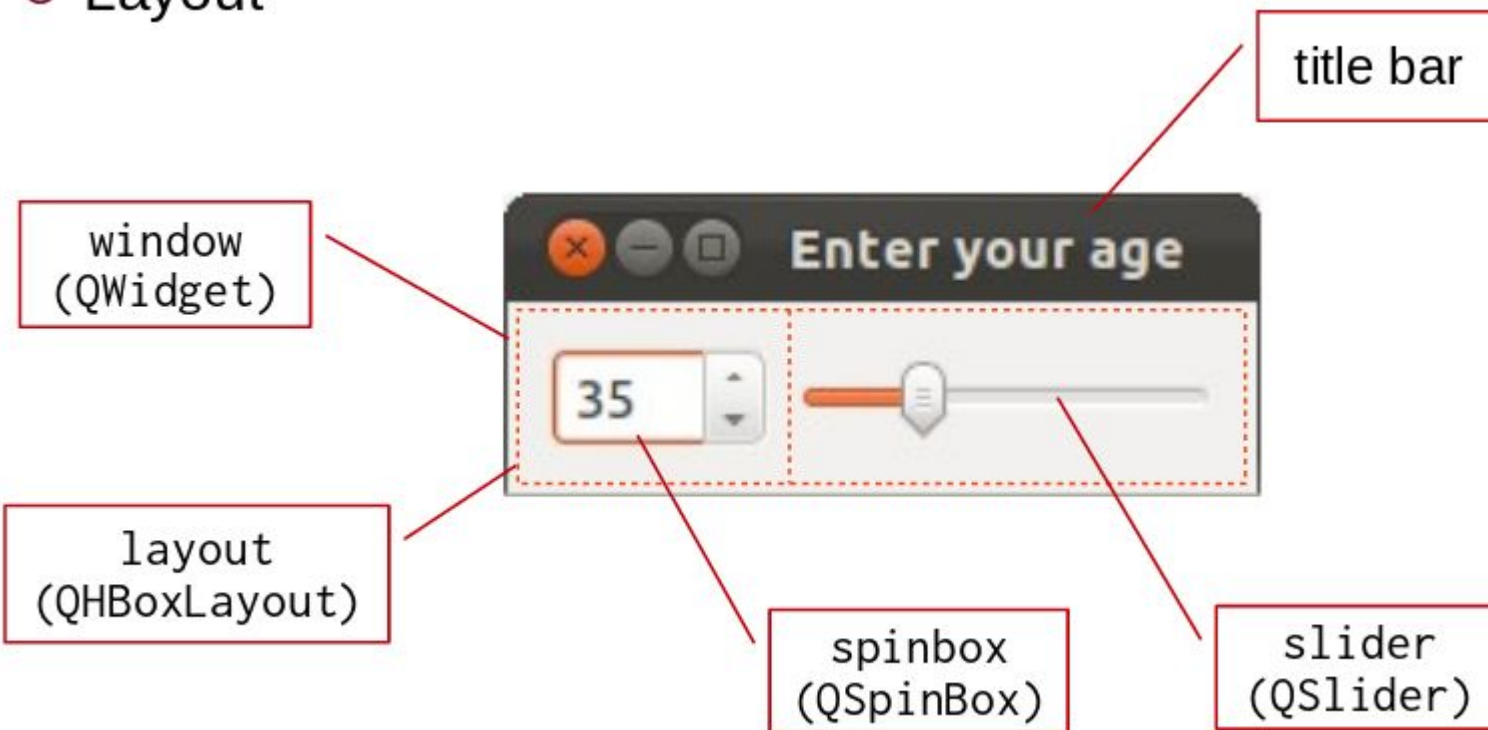
```
window->setLayout(layout);
```

```
window->show();
```

```
return app.exec();
```

```
}
```

Пример



Signal-ы и slot-s

Signals and Slots

When created



setValue(35)



valueChanged(35)



setValue(35)



valueChanged(35)



setValue(35)

no change



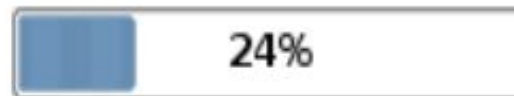
Виджеты

The project has been modified.
Do you want to save the changes?

QLabel



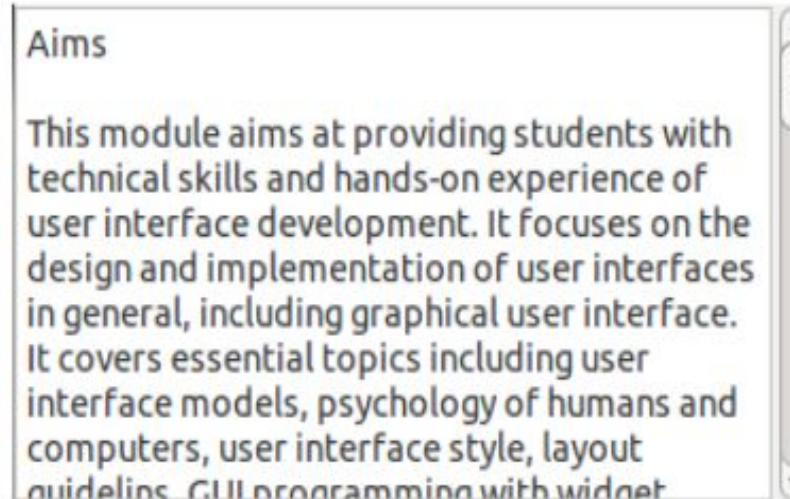
QLCDNumber



QProgressBar

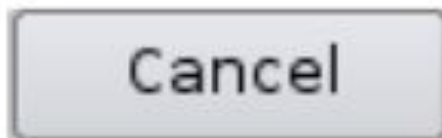


QLabel (image)



QTextBrowser

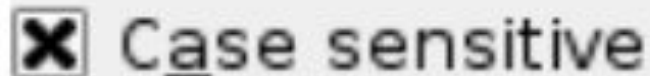
Виджеты



QPushButton



QToolButton



QCheckBox



QRadioButton


Виджеты

QLineEdit

QDateEdit

QTimeEdit

QDateTimeEdit

A horizontal slider widget with a triangular handle on the left side.

QSlider

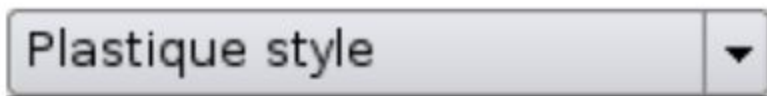
A horizontal scrollbar widget with a central slider and arrow buttons at both ends.

QScrollBar

QSpinBox

QDoubleSpinBox

Виджеты



QComboBox



QFontComboBox



QDial

The **QTextEdit** class provides a widget that is used to edit and display both plain and rich text.

QTextEdit is an advanced *WYSIWYG* viewer/editor that can display images, lists and tables.

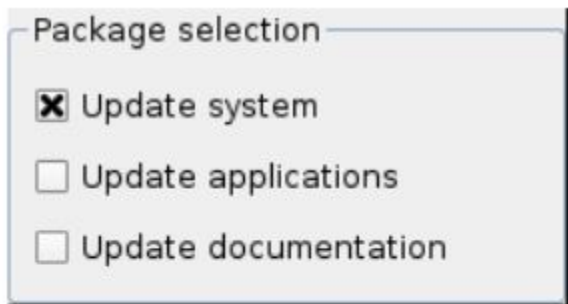
QTextEdit



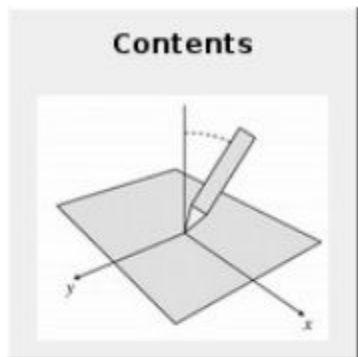
	Sun	Mon	Tue	Wed	Thu	Fri	Sat
22	28	29	30	31	1	2	3
23	4	5	6	7	8	9	10
24	11	12	13	14	15	16	17
25	18	19	20	21	22	23	24
26	25	26	27	28	29	30	1
27	2	3	4	5	6	7	8

QCalendar

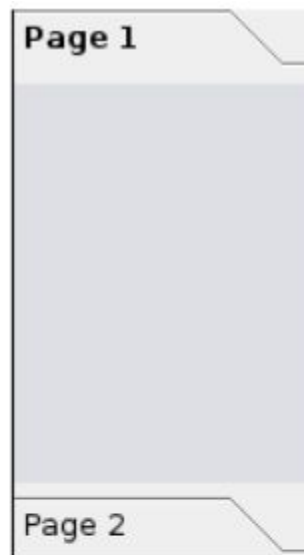
Виджеты



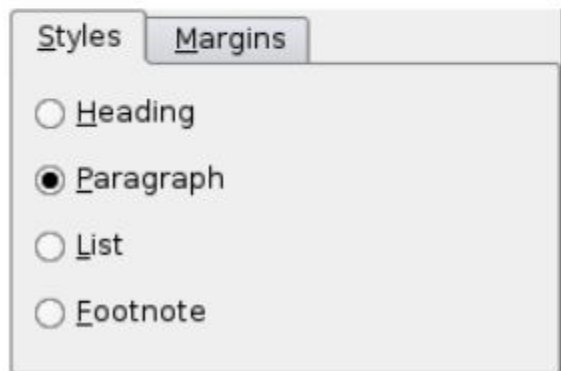
QGroupBox



QFrame



QToolBox

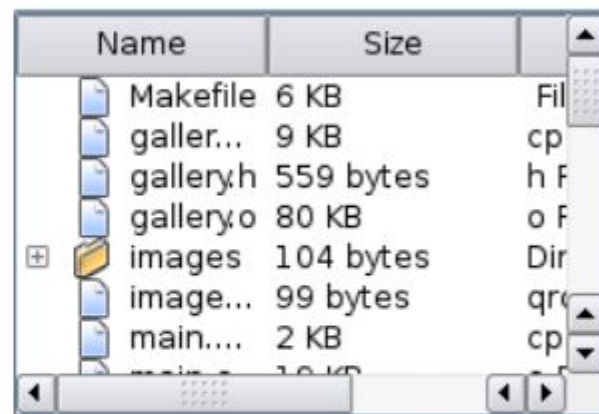


QTabWidget

Виджеты



QListView



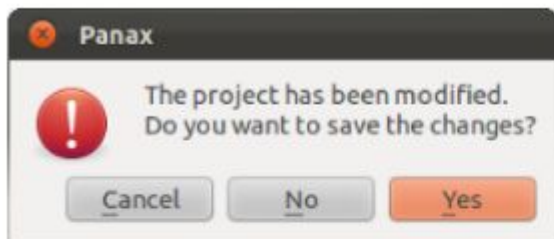
QTreeView

A screenshot of a QTableView widget. It displays a table with three columns: Month, Target, and Action. The data is as follows:

Month	Target	Action
January	6	
February	3	
March	2	
April	3	

QTableView

Виджеты



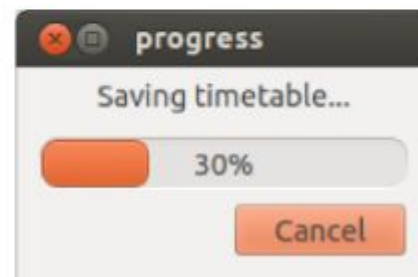
QMessageBox



QErrorMessage

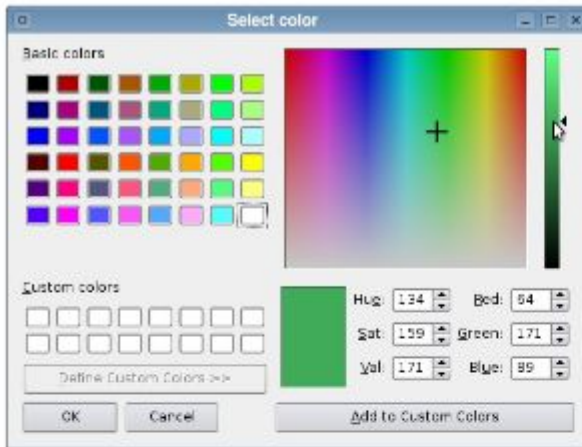


QInputDialog

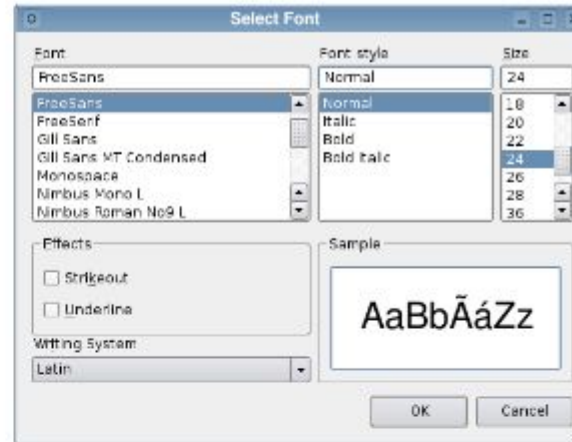


QProgressDialog

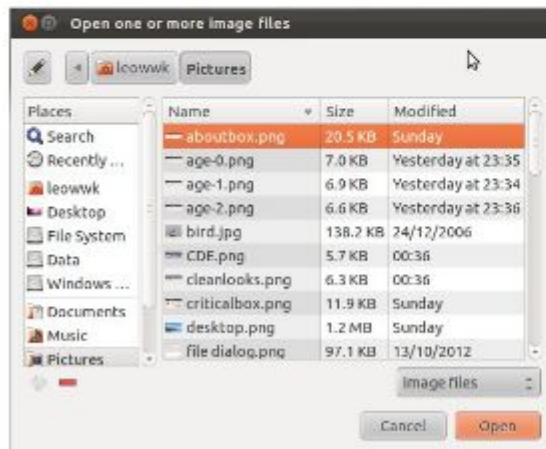
Виджеты



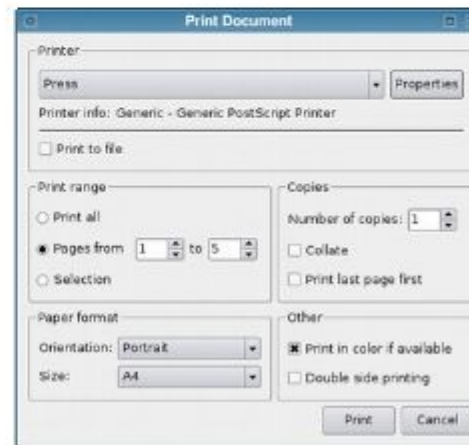
QColorDialog



QFontDialog

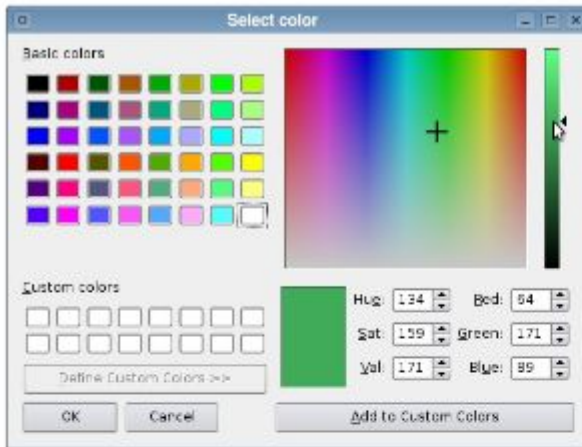


QFileDialog

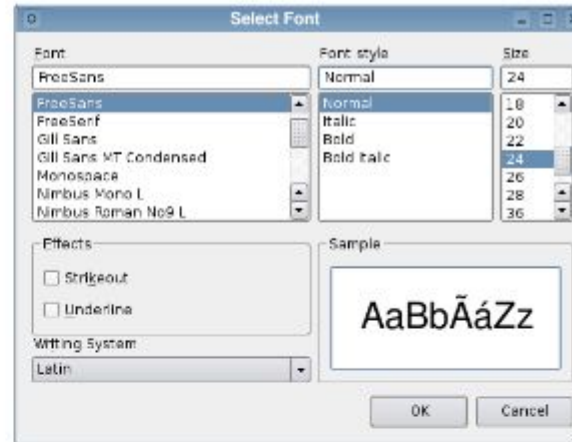


QPrintDialog

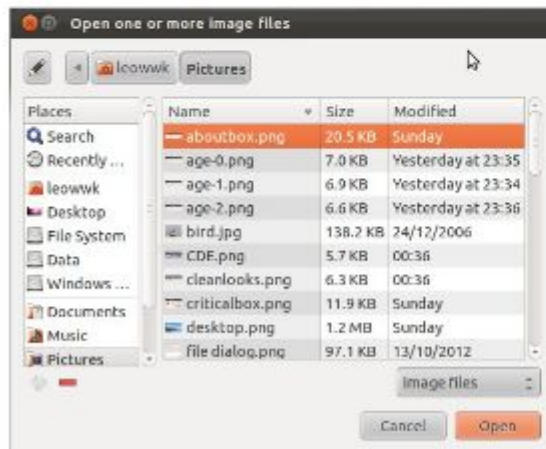
Виджеты



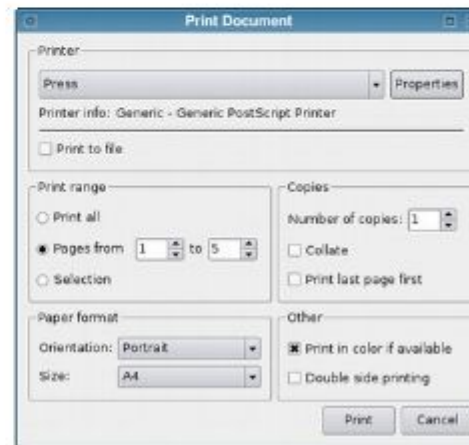
QColorDialog



QFontDialog

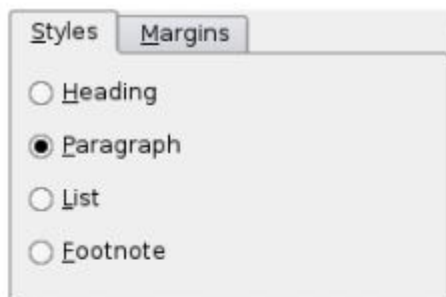


QFileDialog

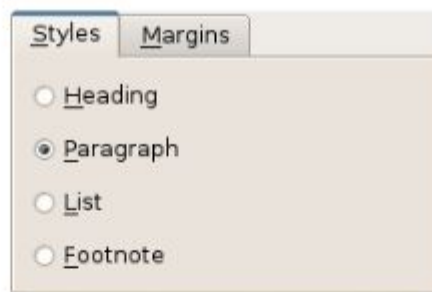


QPrintDialog

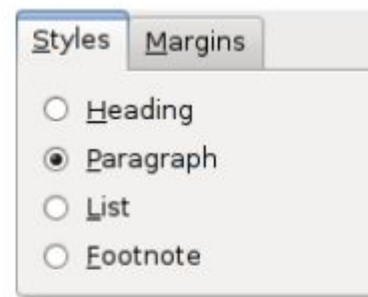
Qt стили



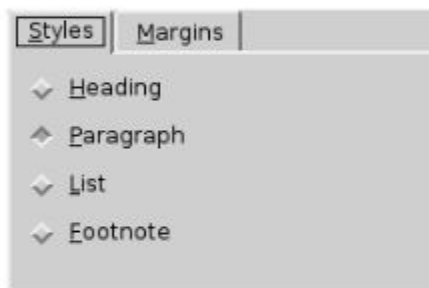
Plastique



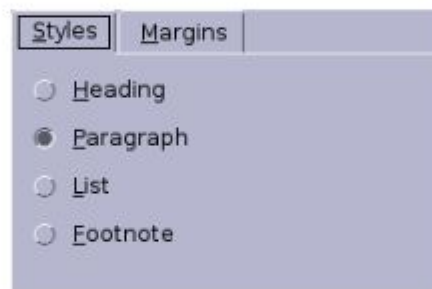
Cleanlooks



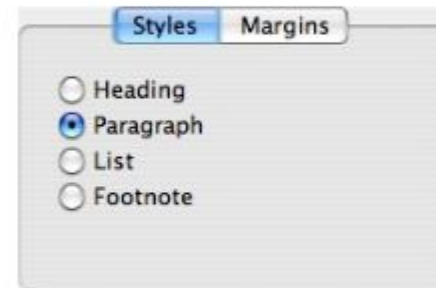
GTK



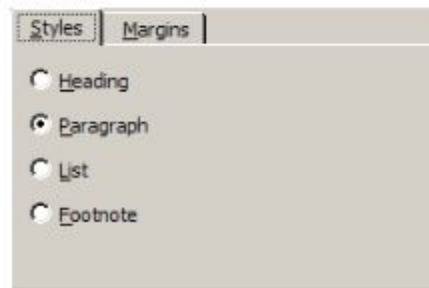
Motif



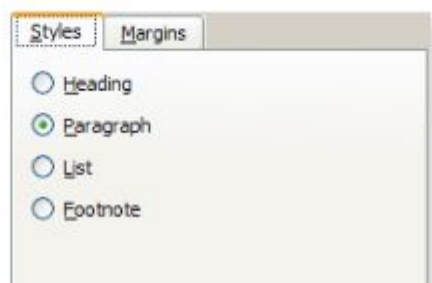
CDE



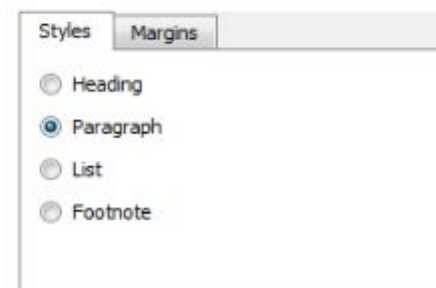
Macintosh



Windows



Windows XP



Windows Vista

Qt лайауты

Можно организовывать виджеты при помощи лайаутов.

Типы лайаутов:

QBoxLayout, QHBoxLayout, QVBoxLayout

QGridLayout

QFormLayout

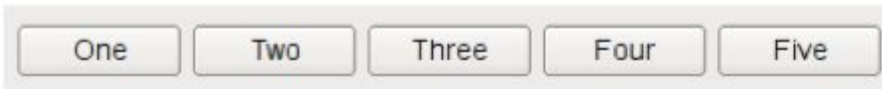
QStackedLayout

Qt лайауты



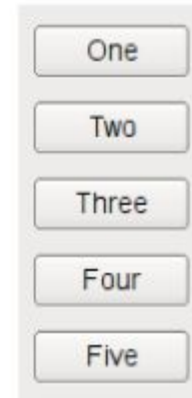
A Qt widget window demonstrating QFormLayout. It contains three labels with corresponding input fields: "Name:" with a text box containing "Gandalf", "Email address:" with a text box containing "gg@troll.no", and "Age:" with a spin box containing "4000".

QFormLayout



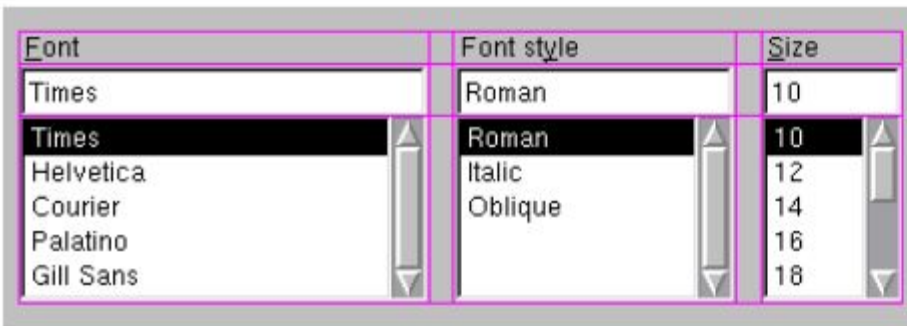
A Qt widget window demonstrating QHBoxLayout. It contains five buttons labeled "One", "Two", "Three", "Four", and "Five" arranged horizontally.

QHBoxLayout



A Qt widget window demonstrating QVBoxLayout. It contains five buttons labeled "One", "Two", "Three", "Four", and "Five" arranged vertically.

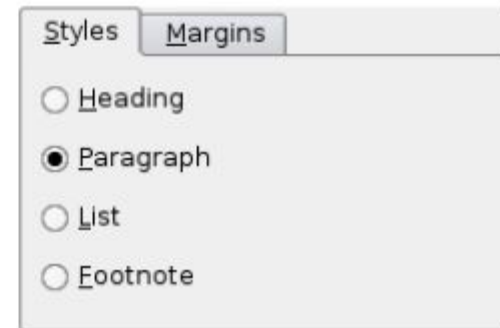
QVBoxLayout



A Qt widget window demonstrating QGridLayout. It contains a table with three columns: "Font", "Font style", and "Size".

Font	Font style	Size
Times	Roman	10
Times	Roman	10
Helvetica	Italic	12
Courier	Oblique	14
Palatino		16
Gill Sans		18

QGridLayout



A Qt widget window demonstrating QStackedLayout. It has two tabs: "Styles" and "Margins". Under "Styles", there are radio buttons for "Heading", "Paragraph" (selected), "List", and "Footnote".

QStackedLayout

Qt лайауты

```
QWidget *window = new QWidget;  
QPushButton *button1 = new QPushButton("One");  
QPushButton *button2 = new QPushButton("Two");  
QPushButton *button3 = new QPushButton("Three");  
QPushButton *button4 = new QPushButton("Four");  
QPushButton *button5 = new QPushButton("Five");  
QHBoxLayout *layout = new QHBoxLayout;  
layout->addWidget(button1);  
layout->addWidget(button2);  
layout->addWidget(button3);  
layout->addWidget(button4);  
layout->addWidget(button5);  
window->setLayout(layout);  
window->show();
```

Литература

1. Jasmin Blanchette, Mark Summerfield. Разработка графического интерфейса с помощью библиотеки Qt3.
2. Бьерн Страуструп. Язык программирования C++.

3. Тут есть несколько неплохих туториалов.

<http://www.cprogramming.com/tutorial.html>

4. Introduction to Programming Concepts in C++ (хорошие базовые лекции по C++ на английском)

<http://staffwww.fullcoll.edu/brippe/csci123/lectures.aspx>

5. <http://rsc-team.ru/index.pl?rzd=2&group=lection&ind=22>

Литература

1. Jasmin Blanchette, Mark Summerfield. Разработка графического интерфейса с помощью библиотеки Qt3.
2. Бьерн Страуструп. Язык программирования C++.

3. Тут есть несколько неплохих туториалов.

<http://www.cprogramming.com/tutorial.html>

4. Introduction to Programming Concepts in C++ (хорошие базовые лекции по C++ на английском)

<http://staffwww.fullcoll.edu/brippe/csci123/lectures.aspx>

5. <http://rsc-team.ru/index.pl?rzd=2&group=lection&ind=22>