2018-2019年度第二学期 00106501

计算机图形学

童 伟 华　管理科研楼1205室

E-mail: tongwh@ustc.edu.cn

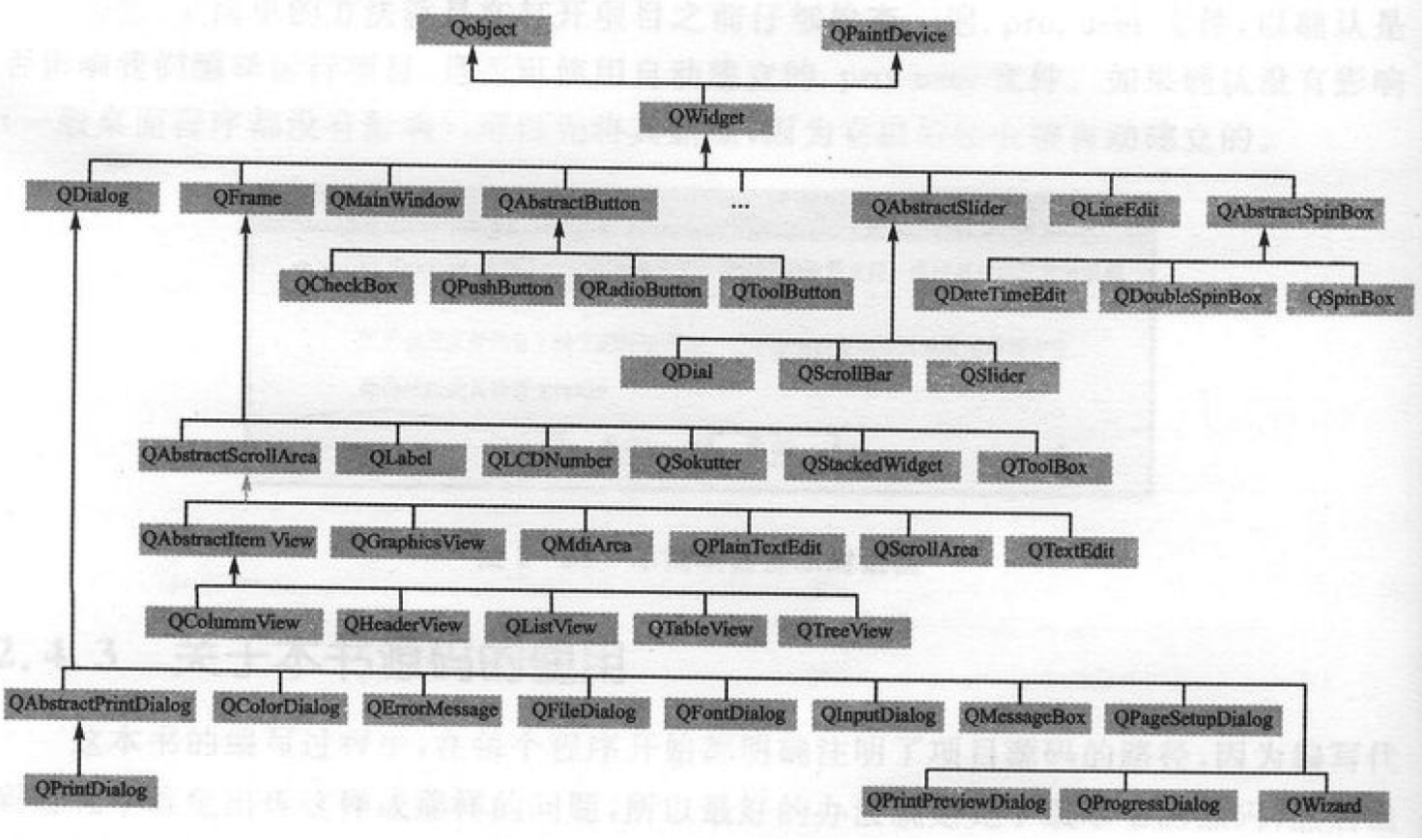
中国科学技术大学　数学科学学院

http://math.ustc.edu.cn/

附讲七　Qt编程（二）

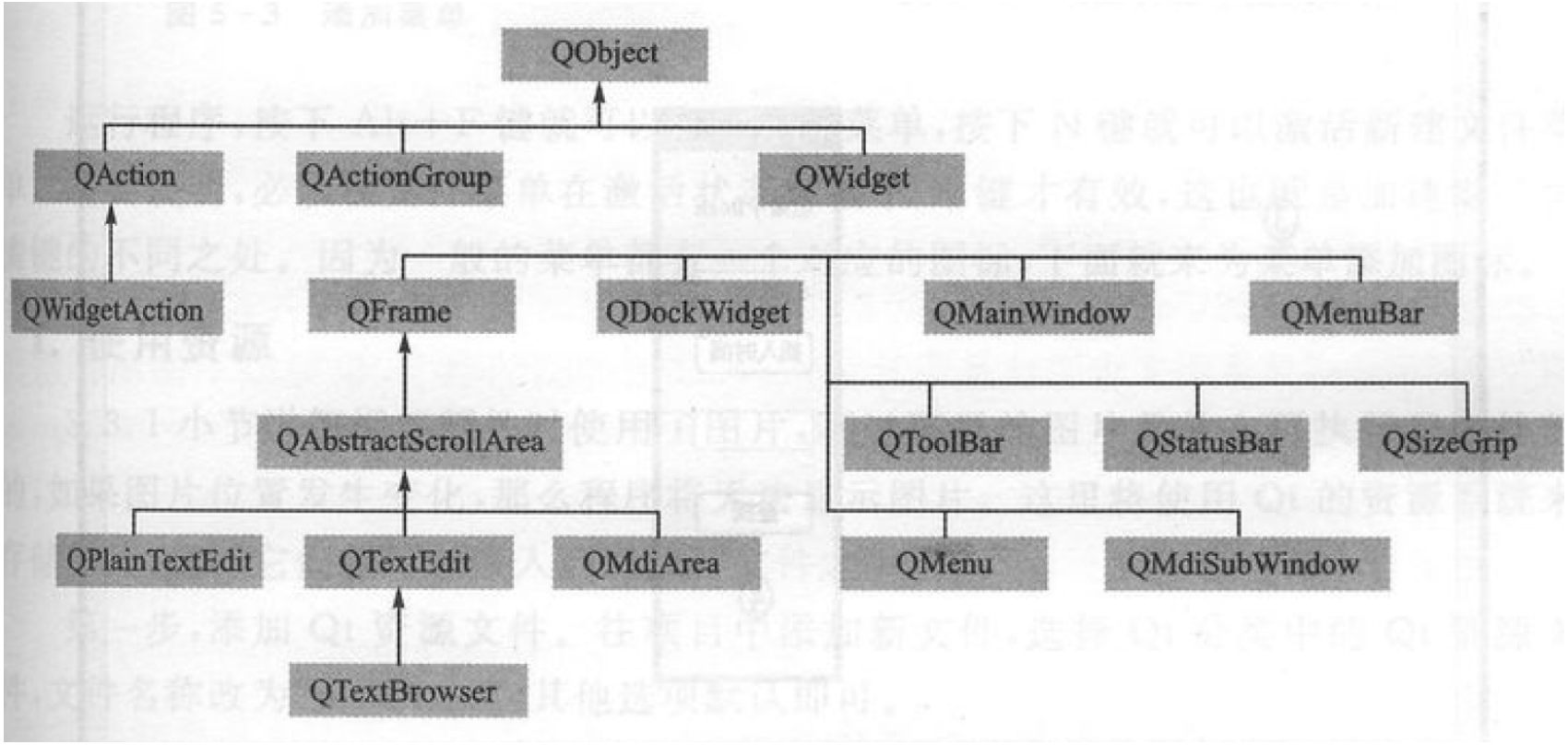
Qt窗口类的简介

- Qt中所有与用户界面相关类的基类：QWidget
  - 继承自QObject，窗口最基础的类
- 与窗口类相关的类关系图



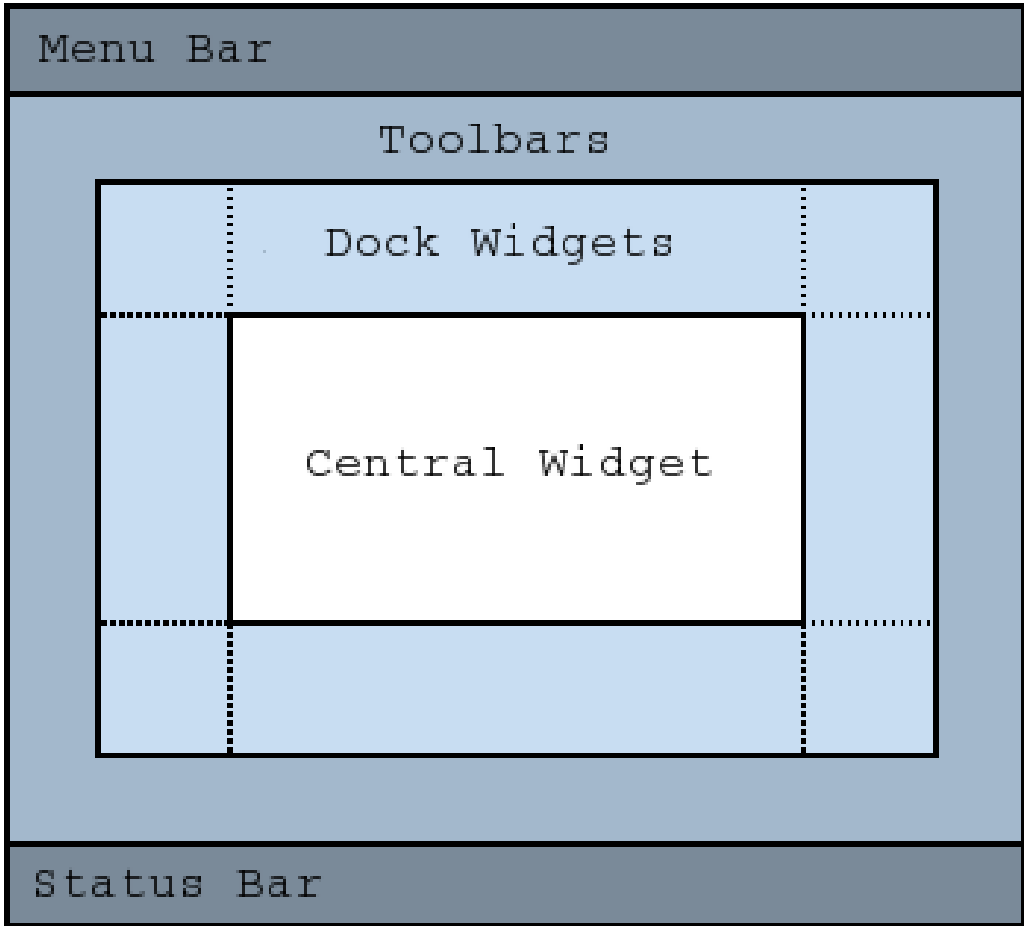
Qt的主窗口类

- 主窗口类：QMainWindow
- 与主窗口类相关的类关系图



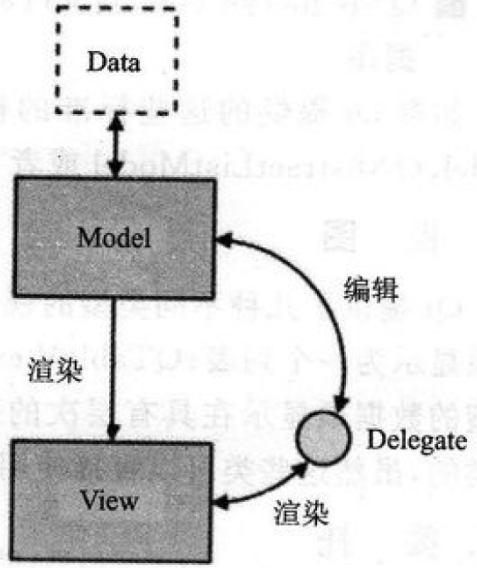
Qt的主窗口类

- QMainWindow的基本组成部分：菜单栏、工具栏、中心部件、可停靠部件、状态栏等



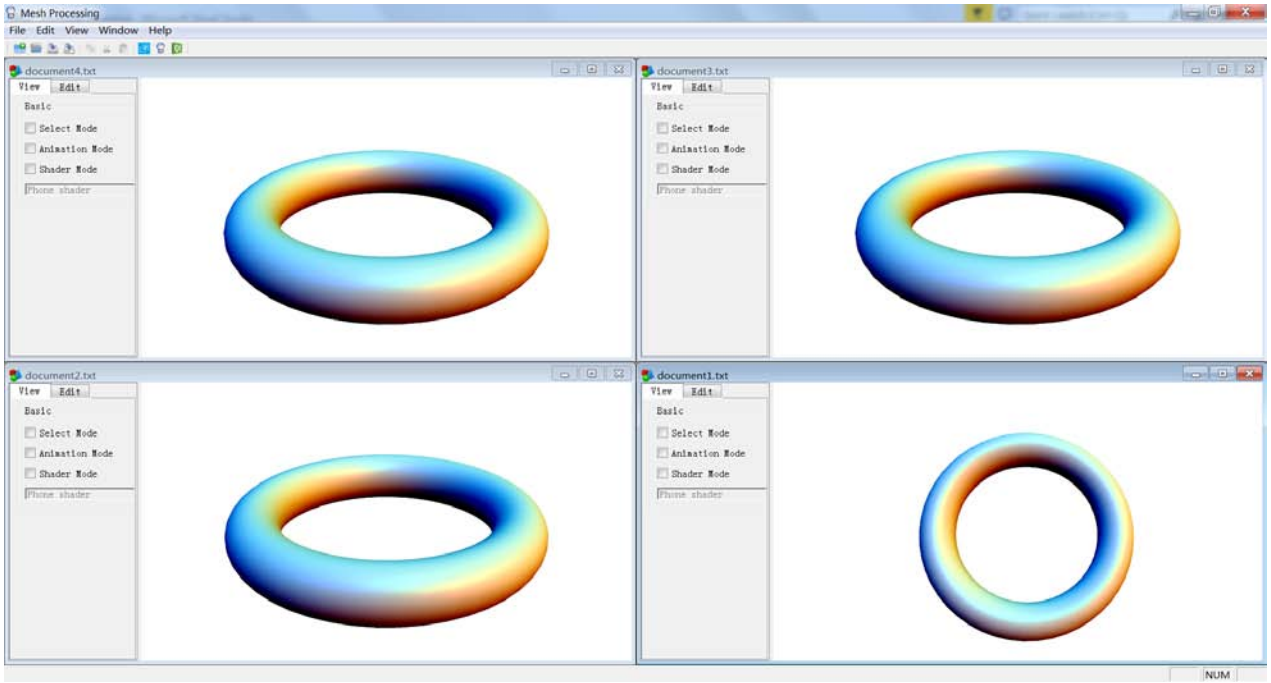
Qt的模型/视图编程

- Qt中的模型/视图框架就是用来实现大量数据的存储、处理及其显示的，起源于Smalltalk的设计模式MVC（model-view-controller），MFC中有类似的编程框架
- 核心思想：数据与界面进行分离，使得相同的数据在多个不同的视图中显示成为可能，而且还可以创建新的视图，而不需要改变底层的数据框架



Qt的单/多文档界面

- 在主窗口的中央区域能够提供多个文档的应用程序就称为多文档界面（Multiple Document Interface）应用程序；否则称为单文档界面（Single Document Interface）应用程序
- 多文档界面使用QMdiArea作为中央窗口部件



Qt的基本界面设计

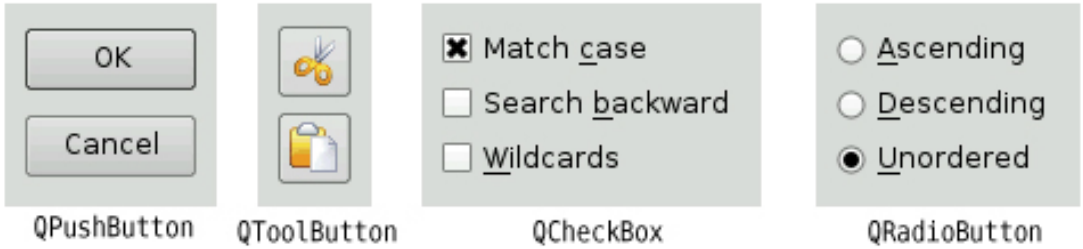
- 菜单（menu）的设计
  - 标准菜单、上下文菜单（自定义菜单）
  - 菜单显示文本，快捷键，状态栏显示文本，工具提示文本等
  - 关键：QAction
  - 使用Qt Design设计或直接用代码生成
- 工具栏（toolbar）的设计
  - 资源文件.qrc，使用rcc编译
  - 图标（icon）
  - 功能与菜单类似
- 对话框（dialog）的设计
  - QDialog类是所有对话框窗口类的基类，模态与非模态对话框
  - 常用的对话框：文件对话框，消息对话框，输入对话框，颜色对话框，字体对话框，进度对话框等

Qt的基本界面设计

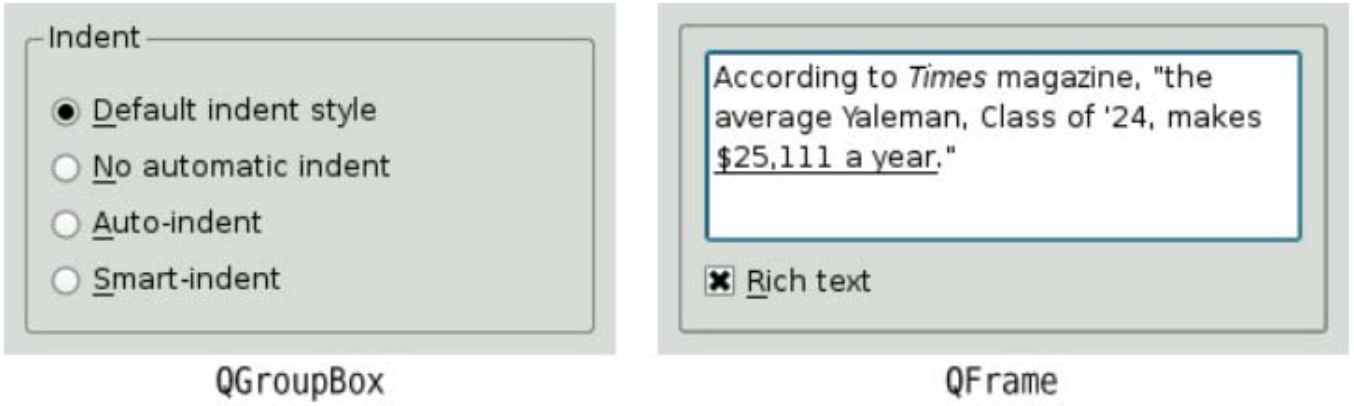
- **常用的窗口部件**
  - QFrame类族（所有带有边框部件的基类）：QLabel（标签部件），QSplitter（分裂器）等
  - QAbstractButton类族（所有按钮部件的抽象基类）：QPushButton，QCheckBox，QRadioButton，QGroupBox等
  - QLineEdit部件：主要用于输入文本、数字等
  - QAbstractSpinBox类族（提供一个数值设定框和一个行编辑器）：QSpinBox，QDoubleSpinBox等
  - QAbstractSlider类族（提供一个区间内的整数值，有一个滑块，可以定位到一个整数区间的任意值）：QSlider，QScrollBar，QDial等
  - 窗口容器类：QTabWidget，QStackedWidget，QListWidget 等
- **设计方式**
  - 图形化设计工具：Qt Design，生成.ui文件，利用uic编译，生成C++代码
  - 或直接编写C++代码

Qt的基本界面设计

- 按钮类部件

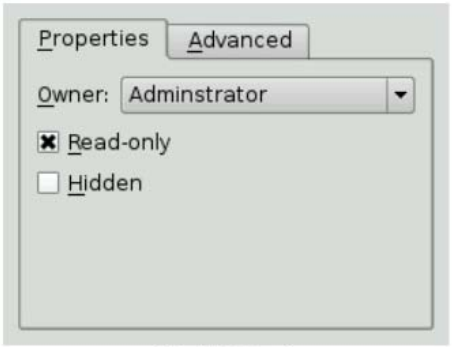


- 单页容器部件



Qt的基本界面设计

- 多页面容器部件

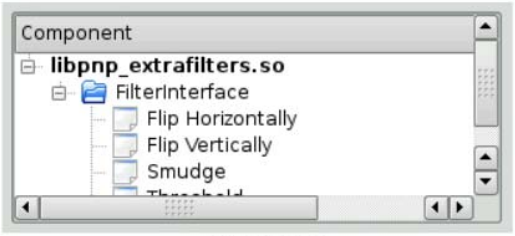


QTabWidget

QToolBox

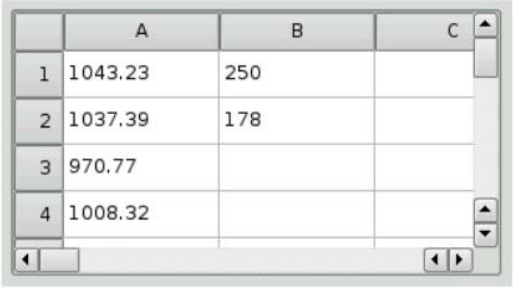
- 项目视图部件

QListView (as list)



QTreeView

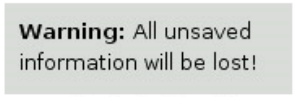
QListView (as icons)



QTableView

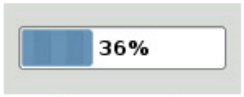
Qt的基本界面设计

■ 显示部件



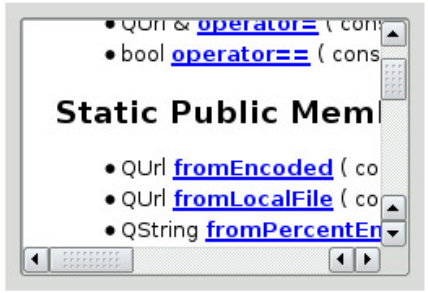
QLabel (text)

QLCDNumber



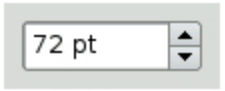
QProgressBar

QLabel (image)

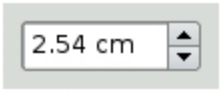


QTextBrowser

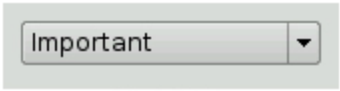
■ 输入部件



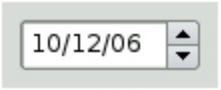
QSpinBox



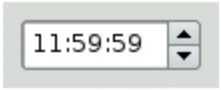
QDoubleSpinBox



QComboBox



QDateEdit

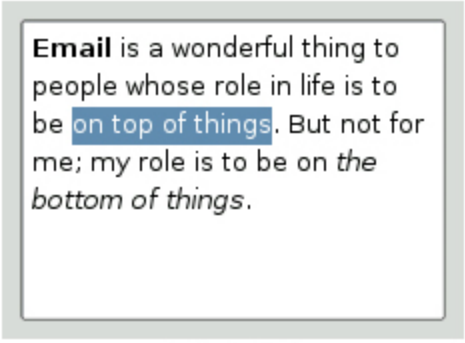


QTimeEdit

QDateTimeEdit

QScrollBar

QSlider



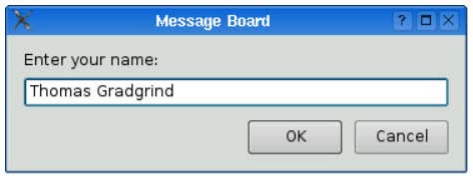
QTextEdit

QLineEdit

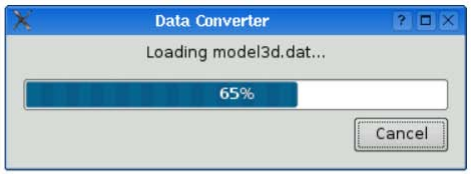
QDial

Qt的基本界面设计

- 消息对话框



QInputDialog

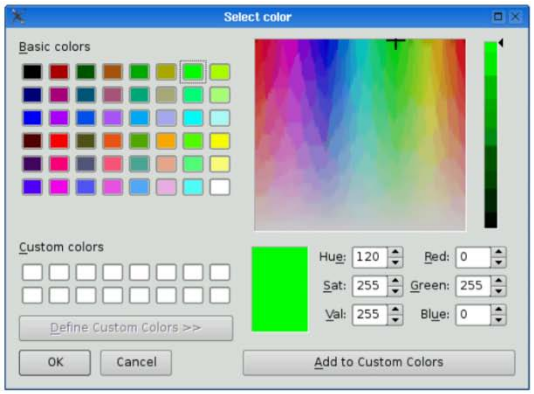


QProgressDialog

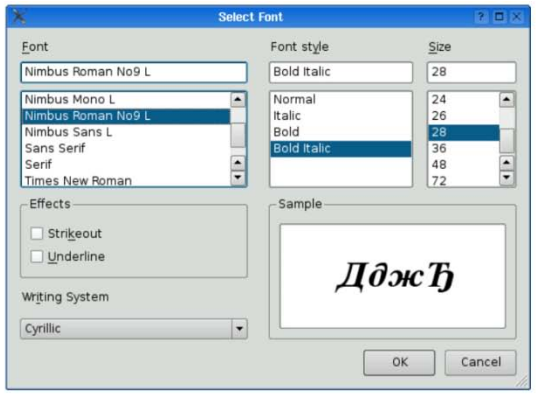
QMessageBox

QErrorMessage

- 颜色和字体对话框



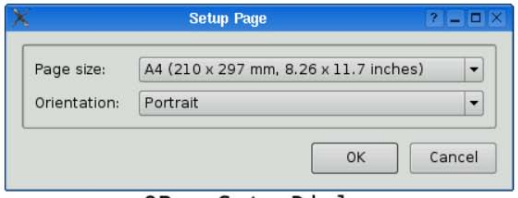
QColorDialog



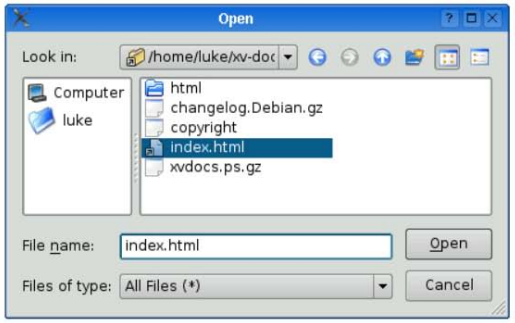
QFontDialog

Qt的基本界面设计

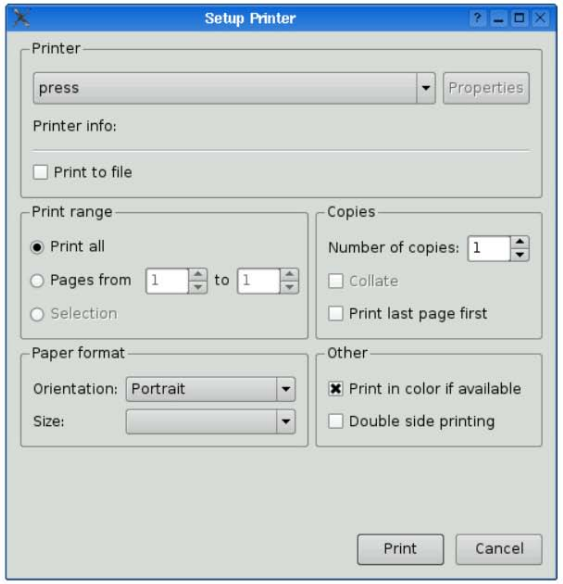
- 文件和打印对话框



QPageSetupDialog

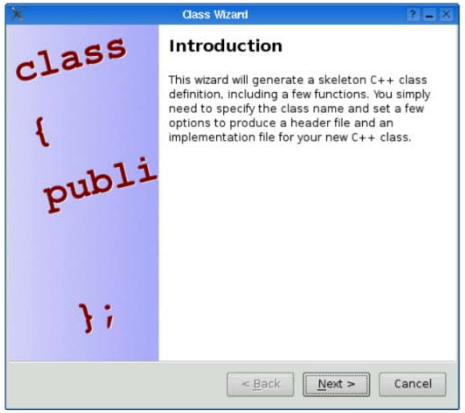


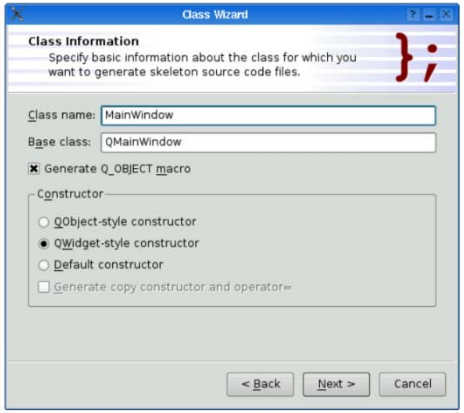
QFileDialog



QPrintDialog

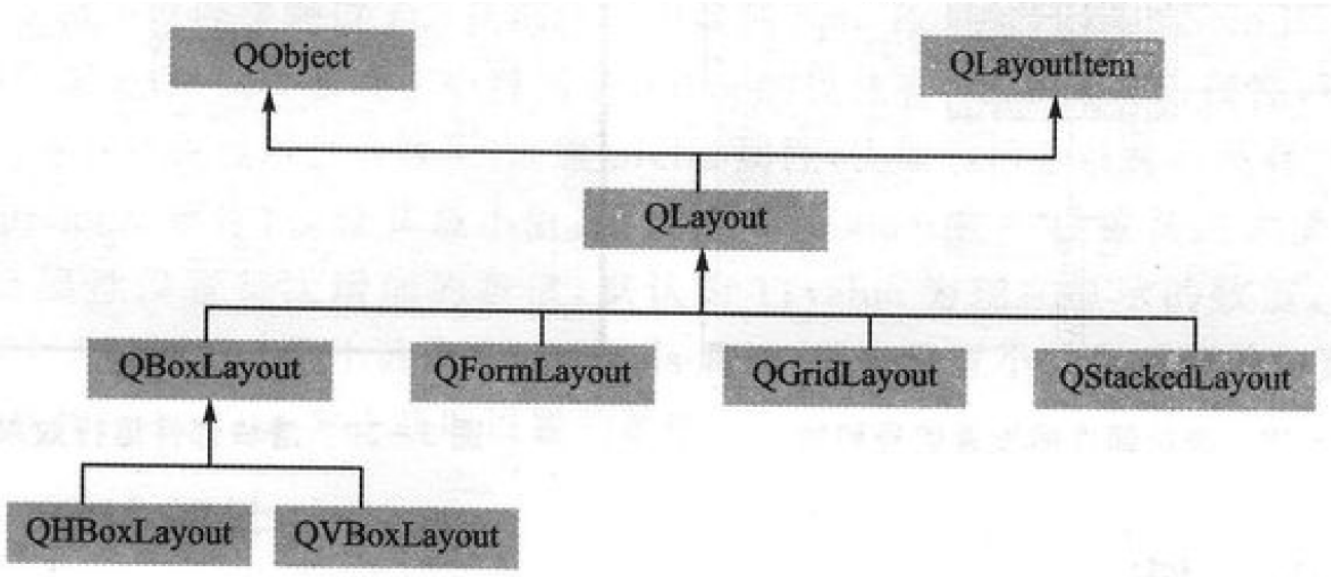
- 向导对话框





Qt的布局管理功能

- Qt的布局管理系统提供了简单而强大的机制来自动排列一个窗口中的部件，确保它们有效地使用空间
- QLayout 类是所有布局管理器的基类，所有QWidget类的子类都可以通过Qwidget::setLayout()函数来设置布局管理器



信号与槽

- Qt的核心特征，也是不同于其他开发框架的最凸出特征
- 作用：对象之间的通讯机制，当一个特殊的事情发生时，源对象可以通过发射一个信号（singal），通知目标对象执行相应的动作（槽，slot，实际上就是一个函数），以这样的方式取代常用的回调（callback）机制
- 关键字：singals，slots，emit
- 限制条件：QObject类及派生的子类才能使用信号和槽

信号与槽

- **基本语法：**

```
QMetaObject::Connection QObject::connect(const QObject * send
er, const char * signal, const QObject * receiver,
const char * method, Qt::ConnectionTypetype =
Qt::AutoConnection)
```

- **使用方式：**
  - 使用QObject类的connect函数
  - 自动关联方式，函数名：on_objectName_singal()（Qt系统利用setObjectName()设置对象名，通过connectSlotsByName()自动相应的信号与槽，常用于一些标准部件的信号处理）
- **方式灵活：**
  - 一个信号可以连接多个槽
  - 多个信号可以连接一个槽
  - 一个信号可以连接另一个信号
  - 连接可以拆除，QObject类的disconnect()函数

信号与槽

- QSignalMapper类：当有多个信号关联到了同一个槽上时，虽然可以利用Qobject::sender()函数返回发送该信号的对象指针，但是需要对每一个信号进行不同的处理，此时可以使用QSignalMapper类
- 作用：实现对多个相同部件的相同信号（没有参数）进行映射，为其添加字符串、数值或窗口参数，然后再发射出去

自定义属性

- Qt提供了基于元对象系统的自定义属性机制
- 基本语法：

```
Q_PROPERTY(type name
           READ getFunction
           [WRITE setFunction]
           [RESET resetFunction]
           [NOTIFY notifySignal]
           [DESIGNABLE bool]
           [SCRIPTABLE bool]
           [STORED bool] [USER bool]
           [CONSTANT]
           [FINAL])
```

- 自定义的属性可在Qt Design与QML语言使用，还可以设置动态属性
- 限制条件：QObject类及派生的子类，Q_PROPERTY宏

对象树与拥有权

- Qt使用对象树（object tree）来组织和管理所有的QObject类及其子类的对象
- 机制：当创建一个QObject类及其子类的对象时，如果使用了其他的对象作为父对象（parent），那么这个对象就会被添加到父对象的孩子（children）列表中，这样当父对象被销毁时，这个子对象也会被自动销毁
- 使用方法：在main函数中，将主窗口部件创建在栈上，而对于其他窗口部件，可使用new操作符在堆上创建，只要指定合适的父窗口，系统就会自动销毁这些窗口对象
- 作用：简化内存管理，降低内存泄露的风险

对象树与拥有权

- 对于窗口部件，父对象还有另一层含义：子窗口部件会显示在它的父对象所在的区域
- 重定义父部件（reparented）：当将一个包含其他部件的布局管理器应用到窗口上，那么该布局管理器和其中的所有部件都会自动将它们的父部件转换为该窗口部件
- 使用children函数获取一个部件的所有子部件的列表

元对象系统

- Qt中的元对象系统（meta-object system）提供了对象间通信的信号和槽机制、运行时类型信息和动态属性系统等
- Qt的关键技术之一
- 要利用元对象系统，需要满足以下条件：
  - 继承自QObject类；
  - 在类的私有声明区声明Q_OBJECT宏
  - 利用moc编译器生成支撑代码（C++）

Thanks for your attention!