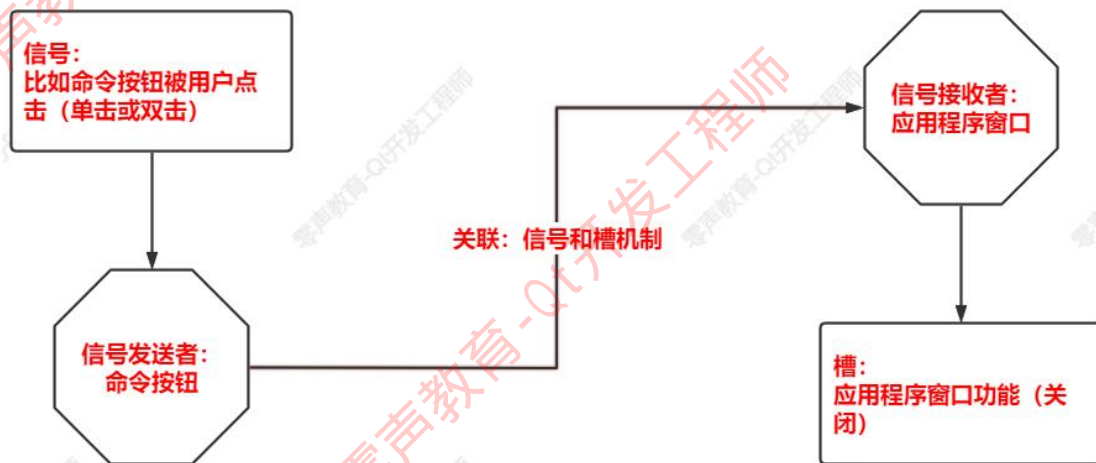


2004--Qt 信号与槽机制实战应用--零声教育 vico 老师

一、Qt 信号与槽机制原理 (Signal & Slot)

1、信号(signal): 所谓信号槽 (观察者模式) , 信号本质是事件。信号展现方式就是函数。当某一个事件发生之后, 则发出一个信号(signal)。

2、槽(slot): 就是对信号响应的函数, 槽就是一个函数。槽函数与普通函数区别: 槽函数可以与一个信号关联, 当信号被发射的时候, 关联的槽函数被自动执行处理。信号与槽关联是使用 QObject::connect()函数进行实现。



信号函数只需要声明 (不需要定义(实现)) , 而槽函数需要定义 (实现) 。

信号和槽机制底层是通过函数之间进行相互调用实现的。每个信号都可以用函数来表示, 称为信号函数; 每个槽也可以用函数表示, 称为槽函数。

槽函数可以使用 public slots/protected slots/private slots 修饰。signals 和 slots 是 Qt 开发当中在 C++ 语言基础之上扩展的关键词, 专门用于指明信号函数和槽函数。

QObject::connect(.....):

```
[static] QMetaObject::Connection QObject::connect(const QObject *sender, const char *signal, const QObject *receiver, const char *method, Qt::ConnectionType type = Qt::AutoConnection);
```

- 参数 1sender: 发出信号的对象;
- 参数 2signal: sender 对象的信号;
- 参数 3receiver: 信号接收者;
- 参数 4method: receiver 对象的槽函数, 当检测到 sender 信号, receiver 对象调用 method 方法。

3、信号与槽机制连接方式

- 一个信号可以跟另一个信号相连;

```
connect(object1,SIGNAL(signal1),object2,SIGNAL(signal1));
```

- 同一个信号可以跟多个槽相连;

```
connect(object1,SIGNAL(signal2),object2,SIGNAL(slot2));
```

```
connect(object1,SIGNAL(signal2),object3,SIGNAL(slot1));
```

- 同一个槽可以响应多个信号。

```
connect(object1,SIGNAL(signal2),object2,SIGNAL(slot2));
```

```
connect(object3,SIGNAL(signal2),object2,SIGNAL(slot2));
```

常用连接方案采用:

```
connect(object1,SIGNAL(signal),object2,SLOT(slot));
```

4、信号和槽机制优势: 松散耦合; 类型案例。一个类若要支持信号和槽, 就必须从 QObject 或 QObject 的子类继承。Qt 信号和槽机制

不支持对模板的使用。

5、信号与槽机制效率：增强对象之间通信的灵活性，但是也会损失一些性能。通过传递一个信号来调用槽函数将会比直接调用非虚函数运行速度慢，主要原因：

- 多线程的时候，信号可能需要排队等待；
- 编组/解组传递的参数；
- 安全地遍历所有的关联；
- 需要定位接收信号的对象。

二、信号与槽实战应用

【运行结果】

