C++&QT知识误区：

基类：

QWidget：空白窗口（触屏）

QMainWindow：带菜单栏的（一般用于pc端）

QDialog：对话框（附加在上面两个上）

This指针：在类未有实例对象前，本身是指向这个类本身的，谁调用，则该函数中的this就 只指向。

构造函数的调用：程序执行到类定义对象的语句时，会执行改类对应的析构函数后，在执行 下一句语句。因此要实现的功能，写在改类的析构函数里。

析构函数在下边3种情况时被调用：（并且谁new谁delete）

1. 对象生命周期结束，被销毁时；
2. 主动调用delete ；
3. 对象i是对象o的成员，o的析构函数被调用时，对象i的析构函数也被调用。

构造函数：

构造函数后加：表示传参初始化列表

函数默认参数：

在函数声明时，参数列表为其赋默认值，如void QActionn**::**triggered(bool*checked* = false)；

局部普通变量超出作用域后会自动释放，看不见效果时，解决方法：

① 将该变量作为类成员，可以指定父类（但是多次重复操作后，可能造成内存占用过多）

② 在局部定义指针，不指定父类，需要手动释放空间（析构函数，或者特有释放函数）

静态成员函数调用：

类名::函数名

函数强制类型转换：（将后面转换为前面），解决函数重载调用二义性问题

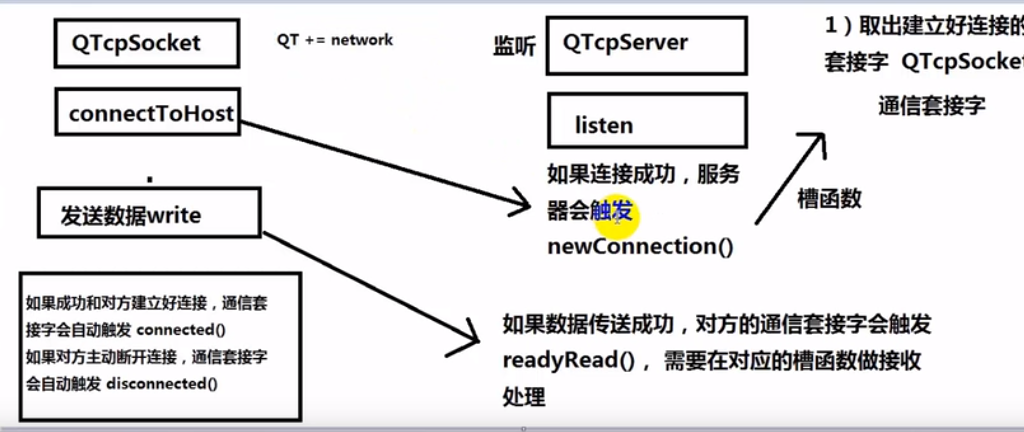
static\_cast<void (类名::\*)(int)>(&类名::函数名)

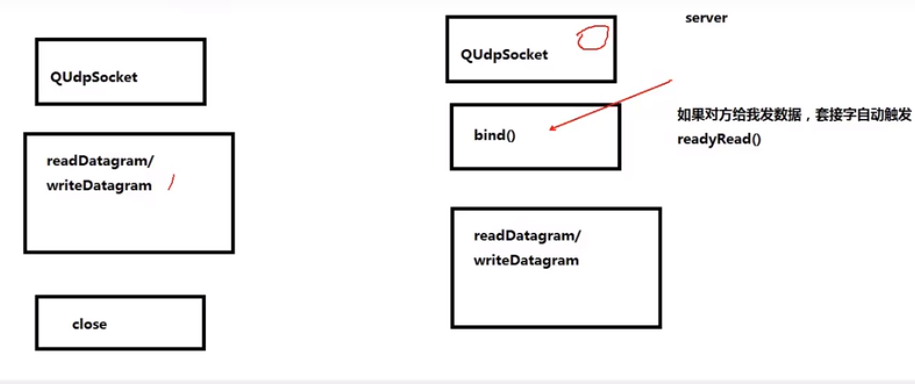
QString用法：

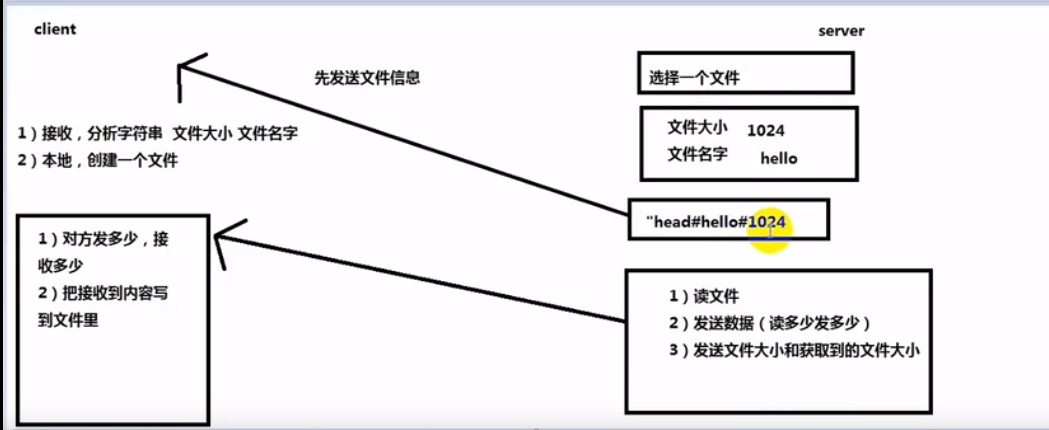
QString str = QString(“abc %1 gopo %2”).arg(342).arg(“guo”);

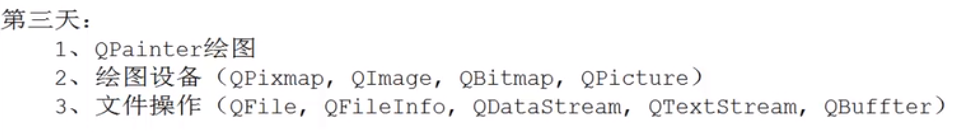
🡪则str内容为：abc 342 gopo guo

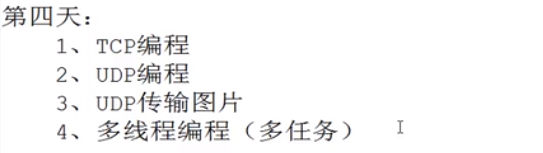


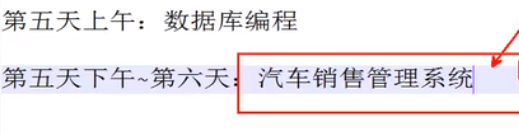


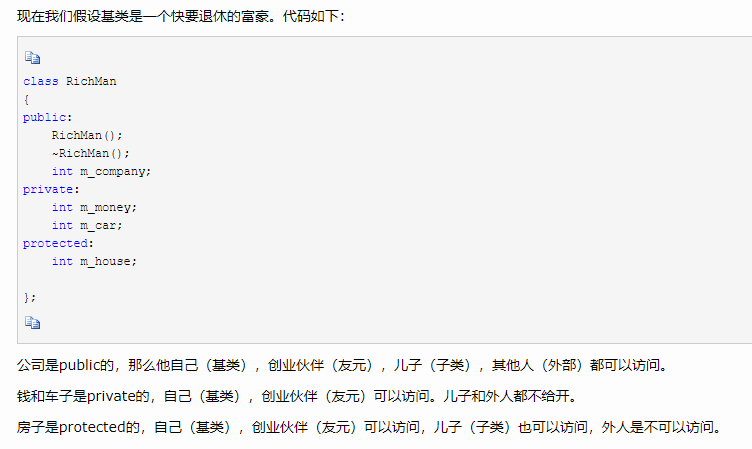












public：自己+友元类+派生类+其他；protect：自己+友元类+派生类；private：自己+友元类；

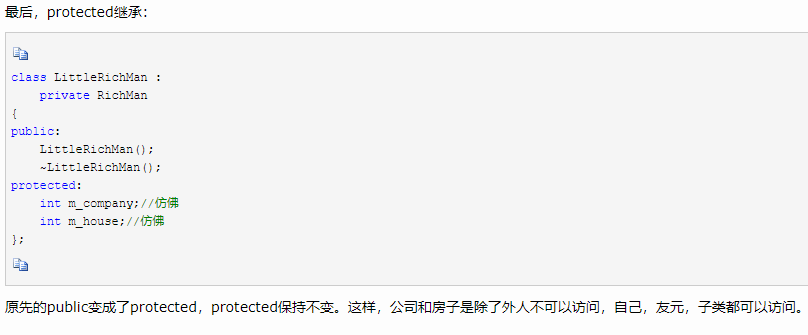
（所以public继承后的派生类不具有访问基类的private的权利）



public继承：只继承基类的public、protected



private继承：只继承基类的public与protected --->子类的private



protected继承：只继承基类的protected与public---> 子类的protected

