1. Qt简介
   1. 跨平台图形界面引擎
   2. 历史
      1. 1991 奇趣科技
   3. 优点
      1. 跨平台
      2. 接口简单，容易上手
      3. 一定程度上简化了内存回收
   4. 版本
      1. 商业版
      2. 开源版
   5. 成功案例
      1. Linux桌面环境 KDE
      2. 谷歌地图
      3. VLC多媒体播放器
      4. …
2. 创建第一个Qt程序
   1. 点击创建项目后，选择项目路径以及 给项目起名称
   2. 名称 - 不能有中文 不能有空格
   3. 路径 - 不能有中文路径
   4. 默认创建有窗口类，myWidget，基类有三种选择： QWidget 、QMainWindow、QDialog
   5. main函数
      1. QApplication a 应用程序对象，有且仅有一个
      2. myWidget w;实例化窗口对象
      3. w.show()调用show函数 显示窗口
      4. return a.exec() 让应用程序对象进入消息循环机制中，代码阻塞到当前行
3. 按钮控件常用API
   1. 创建 QPushButton \* btn = new QPushButton
   2. 设置父亲 setParent(this)
   3. 设置文本 setText(“文字”)
   4. 设置位置 move(宽，高)
   5. 重新指定窗口大小 resize
   6. 设置窗口标题 setWindowTitle
   7. 设置窗口固定大小 setFixedSize
4. 对象树
   1. 当创建的对象在堆区时候，如果指定的父亲是QObject派生下来的类或者QObject子类派生下来的类，可以不用管理释放的操作，将对象会放入到对象树中。
   2. 一定程度上简化了内存回收机制
5. Qt中的坐标系
   1. 左上角为 0 ， 0 点
   2. x以右为正方向
   3. y以下为正方向
6. 信号和槽
   1. 连接函数 ：connect
   2. 参数
      1. 参数1 信号的发送者
      2. 参数2 发送的信号（函数地址）
      3. 参数3 信号的接受者
      4. 参数4 处理的槽函数 （函数的地址）
   3. 松散耦合
   4. 实现 点击按钮 关闭窗口的案例
   5. connect(btn , &QPushButton::click , this , &QWidget::close );
7. 自定义信号和槽
   1. 自定义信号
      1. 写到 signals下
      2. 返回 void
      3. 需要声明，不需要实现
      4. 可以有参数 ，可以重载
   2. 自定义槽函数
      1. 返回void
      2. 需要声明 ，也需要实现
      3. 可以有参数 ，可以重载
      4. 写到 public slot下 或者public 或者全局函数
   3. 触发自定义的信号
      1. emit 自定义信号
   4. 案例-下课后，老师触发饿了信号，学生响应信号，请客吃饭
8. 当自定义信号和槽出现重载
   1. 需要利用函数指针 明确指向函数的地址
   2. void( Teacher:: \* tSignal )( QString ) = &Teacher::hungry;
   3. QString 转成 char \*
      1. .ToUtf8() 转为 QByteArray
      2. .Data() 转为 Char \*
   4. 信号可以连接信号
   5. 断开信号 disconnect
9. 拓展
   1. 信号可以连接信号
   2. 一个信号可以连接多个槽函数
   3. 多个信号可以连接同一个槽函数
   4. 信号和槽函数的参数 必须类型一一对应
   5. 信号和槽的参数个数 是不是要一致？信号的参数个数 可以多余槽函数的参数个数
   6. 信号槽可以断开连接 disconnect
10. Qt4版本写法
    1. connect( 信号的发送者， 发送的信号SIGNAL( 信号) ，信号接受者， 槽函数SLOT(槽函数) )
    2. 优点 参数直观
    3. 缺点 编译器不会检测参数类型
11. Lambda表达式
    1. []标识符 匿名函数
       1. = 值传递
       2. & 引用传递
    2. () 参数
    3. {} 实现体
    4. mutable 修饰 值传递变量 ，可以修改拷贝出的数据，改变不了本体
    5. 返回值 []() ->int {}