2015-04-04 C++ GUI Qt开发入门

A.Qt介绍与跨平台基本环境搭建

列表序号注释:

# **1.** (标题1): 课时标题

## A. (标题2): 课时中的知识点或要点

### i. (标题3): 课件内容要点

### *1(标题4): 知识点点对应讲课内容*

#### 标题4(标题4无列表): 知识点对应介绍

正文(正文): 标题对应讲课内容

目录

1. Qt介绍(课时)

## Qt 基本介绍(知识点)

## Qt历史

## 为什么选择Qt

### Qt包括….

### Qt的优势

## Qt前景

1. Qt学习入门

## 学习Qt需要具备的知识

## 如何学习Qt

1. Windows桌面基本开发环境搭建
2. OS X桌面基本开发环境搭建
3. Linux桌面基本开发环境搭建

# Qt介绍(课时)

本课时我们先会对Qt做一个介绍,让开发者对这个框架有个基本的了解

本课时内容包括:Qt的基本介绍,Qt的历史,我们为什么选择Qt,以及使用Qt开发的前景.

## Qt基本介绍(知识点)

### 什么是Qt:一个跨平台的C++和GUI开发框架

#### 什么是Qt:

#### 首先,Qt不是一个编译器,它是一个基于C++的跨平台应用程序和UI开发框架,由奇趣公司开发,使用的是G++编译器,即GNU工具链.

#### 通过Qt开发的程序,只需开发一次,然后无需重新编写源代码就可以在其他不同的桌面和嵌入式操作系统中部署运行,包括OS X, Windows, Android和iOS

## Qt 的历史

#### 我们先了解一下Qt的相关历史.

### 1991年 Haavard Nord与Eirik Chambe-Eng开始合作开发Qt

#### Qt的诞生也是源自于需求,1991年这两个名字很难念的人接了个项目,项目呢需要有一个能再UNIX,mac,Windows上面跑的界面,于是他们想了,他们需要一个面向对象,而且跨平台的显示系统,然后他们就开始开发了,也就是现在的Qt.

#### 所以说我们程序员是最任性的职业,自己想着要有个上面,就做了.

### 1992年 Eirik提出了”信号与槽”的设想

#### 1992年,他们中的一个人提出了信号槽的设想,这也是一个任性的产物,他们需要有个简单有效而且强大的ui编程规范,于是乎另一个人就把信号槽实现了.

### 1996年 Qt 1.0发布

#### 一直到了1995年,第一个公开发行版本Qt 0.90才发布,当然前面他们已经做了一个能实现部分窗口组件的版本,这个是用来赚钱的.这个Qt当然部署腾讯的那个Qt,很多人以为这是一个缩写,其实Qt就是全称,字母Q也是Qt中类的前缀,这是以为Q在一个字体中很好看(当然这个字体也是他开发的).而后面的t代表的是工具包toolkit

### 2000年 Qt/Embedded Linux发布

#### 2000年的时候发布了Qt的嵌入式版本, 在嵌入式系统上提供了自己的窗口系统,可以作为X11的轻量级替代产品,性能高,占用资源少.Qt在Linux以及嵌入式系统上是具有相当大的优势的,KDE就是利用Qt建立的

### 2008年 Nokia收购TrollTech

#### 2008年的时候诺基亚收购了奇趣公司,用来给他们家的塞班系统开发应用.

### 2009年Qt Creator发布

#### 2009年的时候Qt Creator发布了,这是一个专门为Qt设计,使用Qt开发的IDE,能满足Qt开发的时候的几乎所有需求,包括代码编辑,构建,调试,版本管理等等

### 2010年 Qt Quick发布

#### Qt Quick是Qt上的另一套UI框架,使用了QML描述性语言,对触屏设备的支持更好.具体我们会在后面的课程中再介绍

### 2012年 Digia公司收购Qt

#### 到了2012年,那个时候诺基亚以及开始走下坡路了,没办法,Qt又被卖给了Digia.现在Qt在国内也是有办事处的,主要负责技术支持和企业培训.

## 为什么选择Qt

#### 为什么我们会选择Qt?下面我们介绍的Qt的特性与优势几乎包括了这个问题的所有答案.但是最重要的一点,我认为一个开发工具的成功,应该缘于程序员们对它的喜欢.下面我会介绍Qt的特性与优势,我想在我介绍完之后,你也会认为Qt是一个很适合用来开发的工具,并在使用Qt的过程中逐渐喜欢上它

### Qt包括:一个模块化的C++类库,Qt Creator与Qt Designer, Qt Assistant, Qt Linguist,

#### 说到Qt的特性与优势,首先我们要来说说Qt里面包括了什么,为什么我们说Qt会是一个适合用来开发的工具的工具

#### Qt中提供了丰富的模块化类库,包含了构建高级跨平台应用程序所需的模块,是Qt跨平台开发的基础.通过使用这些模块,开发者们可以在不同的平台上进行开发,而不需要去了解不同平台上大量不同的API

#### 这些类库具有主观化,易学易用,代码可读性高,易维护的特点

#### QtCreator是一个优秀的跨平台IDE开发套件,使用Qt开发,能充分的满足Qt的开发需求.当然它也可以单独作为一个很棒的IDE工具使用.

#### 目前,QtCreator可以在OS X, Windows跟Linux桌面操作系统上运行,通过Qt Creator可以很方便的为不同的桌面操作系统和移动设备创建程序并部署调试.

#### Qt Designer是一个基于Qt的可视化的UI设计工具,既可以在整合了Qt Designer的QtCreator中使用,也可以作为一个独立程序运行.通过Qt Designer, 你可以用一种所见既得的方式生成应用的UI界面,使用起来很方便.

#### 

#### Qt Assistant是Qt的帮助浏览器,具有强大的查询和索引功能.使用它阅读Qt的帮助文档能够比Web浏览器更加快速和容易.

#### 

#### Qt Linguist是一个Qt的翻译工具.Qt提供了对应用程序本地化的支持,使用Qt Linguist可以很方便的对Qt程序的语言文本进行读取和翻译.

#### 

#### (总结)上面我们简单介绍了Qt中几种常用的工具.在开发的过程中配合使用这些工具,可以做到事半功倍的效果.另外,我们也可以说,Qt提供了大部分,甚至可以说是几乎所有我们开发一个应用需要的东西.丰富的类,一个足够好用的IDE,UI设计工具,软件本地化的翻译工具,还有一个包含所有这些的用法的帮助程序.

### 接下来,我们将介绍一下Qt的一些特性,以及使用Qt开发具有怎样的优势.

#### 跨平台:一次编写,到处部署

#### Qt支持的平台

##### Windows

##### OS X

##### Linux

##### Android/iOS/WP等移动设备操作系统

##### 嵌入式操作系统

##### etc. …

#### QtEveryWhere

##### 只需一次开发,就可跨不同桌面和嵌入式系统部署

##### 使用单一源代码库定位多个操作系统

##### 开发者无需在API上花费太多精力

#### (内容)我们知道Qt最显著的优势就是跨平台,一次编写,到处部署.

#### 目前Qt支持的平台包括:……

#### 之前我们介绍过,Qt最早就是为了在Windows与X11直接跨平台而开发的,另外还有Qtopia支持各种嵌入式操作系统,比如Embedded Linux, Windows CE等等.

#### 使用Qt开发的程序,只需要在一个平台开发完成,不需要重新编写源代码,就可以在其他平台上部署允许,我们成为QtEveryWhere, Qt无处不在.上面说过Qt模块化的类库,正是由于这些丰富的类库,我们在用Qt编写软件时无需考虑平台,也不用去研究那些复杂的系统API,特别是这些API在不同平台上都不同.我们只需要专注于开发好的软件(这也是作为一个程序员最大的追求了,能专心开发自己满意,用户喜欢的程序).

#### 丰富易用的开发工具

#### 前面我们介绍了Qt自带的一些简单易用的工具,这些工具也是Qt的优势,它几乎提供了我们在开发时需要用到的所有东西.

##### Qt Creator

#### 作为一个集成开发环境,在QtCreator中,集成了Editor编辑器, Designer可视化UI设计工具, Assistant助手,以及Linguist翻译家等工具

##### Qt Designer

#### 对很多程序员来说,纯粹使用代码来开发整个应用程序很正常,包括应用的界面,很多使用Qt开发的程序员也是这样做的.但是毫无疑问,使用Qt Designer这种可视化方式来设计界面,会比手工敲代码更自然,效率更高.而且这种方式在测试修改的时候也会更加容易.

#### QtDesigner创建的窗体最终仍旧是C++代码,所以可以把它看做是一个传统的开发工具集,可以根据开发者的习惯配合不同的编译器使用.

#### 比如,我可以先用QtDesigner创建一个应用的UI,然后在MSVC中开发程序并调用这个UI.

#### 优美的信号槽机制

#### 正如我们上面所说,Qt使用的信号槽机制是一种简单有效而且很强大的图形界面编程规范,但是我们用了优美来形容它.因为信号槽实在是太漂亮了.

#### 信号槽是一种应用于对象之间的通信的高级接口,它作为Qt的核心特性,也是Qt跟其他开发框架的区别.在其他我们熟悉的GUI工具中,每个窗口widget都有一个回调函数用来响应触发的动作,而回调函数通常是一个指向某个函数的指针.但是在Qt中,使用信号和槽代替了这些凌乱的指针,我们可以用更简洁的方式去处理程序通信,而且信号槽可以携带任意数量和类型的参数,他们完全类型安全.

#### Qt很容易扩展,并且允许真正地组件编程

#### Qt是一个经过完全模块化的开发框架,它内置了插件机制.我们可以很方便的利用它自带的插件机制来实现程序的模块化开发设计,极大的提高开发效率.

#### 随着我们越来越多的使用Qt进行开发,你还会发现Qt在其他方面的优势,比如Qt的私有实现可以保证代码的二进制兼容性,还有元对象与隐式共享等等,这些内容我们会随着课程的深入在后面介绍.

## Qt前景

#### 我们说一个开发框架的生命力是否旺盛,主要是看技术优势,文档,以及使用人数.之前我们说了Qt的技术优势以及它详细的开发文档,而其他的则需要广大Qt开发者去共同维护了.下面我们会说说使用Qt开发的前景.

### Qt VS MFC

#### 说道Qt前景,我想作为一个程序员比较关心的应该是它与其他开发工具相比较优势的优势在哪里.那么我们就与Windows下的MFC做一个比较.MFC也就是微软基础类库,它为程序员调用Windows系统功能提供接口,它跟Qt一样提供了一个面向对象的框架.关于MFC我们不做详细介绍.另外关于开发工具间的优劣其实是一个仁者见仁智者见智的问题,需求不一样适合的工具也不一样.我们这里只是简单介绍一下Qt对比MFC的几个优势与缺点

##### Qt开发效率高

#### 经过多年的发展,MFC具有相当的使用人数,但是它也变得比较冗杂.跟他比起来Qt就显得简洁明了,无论是模块化的类库还是信号槽机制,使用起来都效率很高

##### 容易上手

#### Qt很容易上手,很多机制跟MFC差不多,只需要很基本的C++基础就可以开始学习Qt.另外,它的某些语法跟Java的Swing比较像.所以如果有了解过swing的同学,可以比较容易理解Qt的语法

##### 内存回收机制

#### Qt在一定程度上简化了内存回收机制,而MFC需要自己实现框架才能做到

#### 当然,Qt作为一个还算年轻的框架,或许还会有这样那样的不足,比如说因为它的界面效果依赖于本地引擎,所以运行效率不如MFC之类的原生框架(也正是因为这样所以Qt能更好的实现跨平台),还有Qt的信号槽机制要比回调慢一些,主要损失在类型检查和参数安全上的,这也是为了开发者考虑,以及Qt国际化需要程序员来指定编码等等,但是无法否认Qt正在做得越来越好,现在开始学习使用Qt肯定是明智的.

### 介绍Qt授权协议

##### 开源协议:为开源开发和开发者提供自由

##### 商业授权:适用于不同需求的多种价格模式

#### Qt采用了灵活的双重授权协议,其中开源协议为开源开发和开发者提供服务,不同的模块分别遵循LGPL2.1与LGPL3协议

#### Qt也有商业授权,拥有适用于不同需求的多种价格模式,如果公司或个人购买了商业授权,那么就可以拥有代码所有权,同时得到来自The Qt Company的技术支持

### Qt应用举例

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

#### Qt的应用范围很广,尤其是嵌入式以及UI相关的应用很多都是使用Qt开发的,这里我举几个常用的软件作为例子:PS,Skype,Google Earth,Linux下的KDE桌面,VBox,还有Opera浏览器……

# Qt学习入门

为了能在后面的课程中更好的学习Qt开发,本课时我们会介绍一些与Qt学习相关的内容,包括学习Qt要具备的基本知识,以及如何更好更高效的学习Qt的一些经验

## 学习Qt需要具备的知识

### Qt入门需求

#### Qt的学习很容易上手,无论是0经验的C++初学者,还是你做过多年VC开发,都可以很容易的使用Qt

##### 了解C++

##### 会使用循环,判断,选择,顺序等基本语法

#### 使用Qt进行开发,最低的要求只需要对C++有一个基本的了解,以及会使用循环,判断,选择,顺序等基本语法

### Qt进阶需求

##### 熟悉C++的OOP特性

##### 熟悉C++的封装,继承和多态

##### 熟悉C++模板

##### 会使用容器（vector、set、map等）

##### 熟悉C++内存管理，可灵活使用指针

### Qt高级需求

##### 熟练使用C++

##### 熟悉C++11语法

##### 了解Qt的元对象系统

## 如何更好的学习Qt

### 了解C++

### 了解Qt,知道Qt是什么

### 学会阅读Qt文档

### 熟练使用开发工具

### 最重要的一点:多做,多问

## 关于跨平台

### 为什么要使用跨平台

### Qt与其他跨平台语言的区别

#### 首先,Qt不是一门语言,它是一个C++开发框架,他也没有自己的编译器而是使用G++编译器.

# WIndows桌面基本开发环境搭建

# OS X桌面基本开发环境搭建

# Linux 桌面基本开发环境搭建