

## 第 004 讲 实践 Qt 5 模板库项目中应用

### 零声教育--Vico 老师

- 1、我们为什么要学习 Qt: Qt 跨平台框架（设计、开发、部署）
- 2、目前全球顶尖公司也正用 Qt 打造高端产品：标致汽车数字座舱、LG（智能电视和消费类电子设备）、奔驰汽车数字座舱等等。

#### 【QT 字符串类及窗口类】

##### 1、字符串类（操作字符串）

a.QString 提供一个二元的"+"操作用过组合两个字符串，并提供一个"+="操作符用于一个字符串追加到另一个字符串的末尾，具体案例：

```
QString str1="hello";  
str1=str1+" world"; // str1="hello world"  
QString str2="How do";  
str2+="you do"; // str2="How do you do"
```

b.QString::append()函数具有与"+="操作符同样的功能，实现在一个字符串的末尾追加一个字符串，具体案例：

```
QString str1="ABCD";  
QString str2="XYZ";  
str1.append(str2); // str1="ABCDXYZ"  
str2.append(" ABC"); // str2="XYZABC"
```

c.组合字符串的另一个函数是 QString::sprintf(), 具体案例：

```
QString str;  
str.sprintf("%s", "welcome "); // str="welcome"  
str.sprintf("%s", "to you"); // str="to you"  
str.sprintf("%s %s", "welcome ", "to you"); // str="welcome to you"
```

d.Qt 还提供另一种方便的字符串组合方式，我们使用 QString::arg() 函数，此函数的重载可以处理很多的数据类型。为什么要使用它，因为它类型案例、完全支持 Unicode，具体案例：

```
QString str;  
str=QString("%1 was born in %2.").arg("vico").arg("1978"); // str="vico was born in 1978."
```

e.QString 也提供一些其他组合字符串的方法，包括如下几种：

insert()函数：在原字符串特定的位置插入另一个字符串；  
prepend()函数：在原字符串的开头插入另一个字符串；  
replace()函数：用指定的字符串代替原字符串的某些字符；

f、有时我们去掉一个字符串两端的空白（空白字符包括回车字符"\n"换行字符，制表符"\t"，空格字符" "等等）

QString::trimmed()函数：移除字符串两端的空白字符；  
QString::simplified()函数：移除字符串两端的空白字符，使用单个空格字符" "代替字符串出现的空白字符

## 2、查询字符串数据

a.函数 QString::startsWith()：判断一个字符串是否以某个字符串开头。该函数有两个参数，第一个参数指定一个字符串，第二个参数指定是否大小写敏感，具体案例：

```
QString str="ABC DEF GHI JKL";  
str.startsWith("ABC",Qt::CaseSensitive); // 返回结果 true  
str.startsWith("GHI",Qt::CaseSensitive); // 返回结果 false
```

b.函数 QString::endsWith()类似于 QString::startsWith()，此函数判断一个字符串是否以某个字符串结尾。

c.函数 QString::contains()判断一个指定的字符串是否出现过，具体案例：

```
QString str="ABC DEF GHI JKL";  
str.contains("DEF",Qt::CaseSensitive); // 返回 true
```

## 3、字符串转换

a.QString::toInt()函数：将字符串转换为整型数值，类似函数还有 toDouble()、toFloat()、toLong()等等

```
QString str="125";  
bool bok;  
int hex=str.toInt(&bok, 16);  
int dec=str.toInt(&bok,10);
```

4、QString 提供字符编码集的转换函数将会返回一个 const char\*类型版本的 QByteArray，即构造函数 QByteArray(const char\*)构造的 QByteArray 对象。转换函数有如下几种：

toAscii()：返回一个 ASCII 编码的 8 位字符串  
toLatin1()：返回一个 Latin-1(ISO8859-1)编码的 8 位字符串  
toUtf8()：返回一个 UTF-8 编码的 8 位字符串  
toLocal8Bit()：返回一个系统本地编码的 8 位字符串

## 5、容器类

a.QList 类、QLinkedList 类与 QVector 类

QList 类：QList<T>是目前最常用的容器类，它存储给定数据类型 T 的一系列数值。QList::append，QList::prepend()，QList::insert()。

案例分析如下：

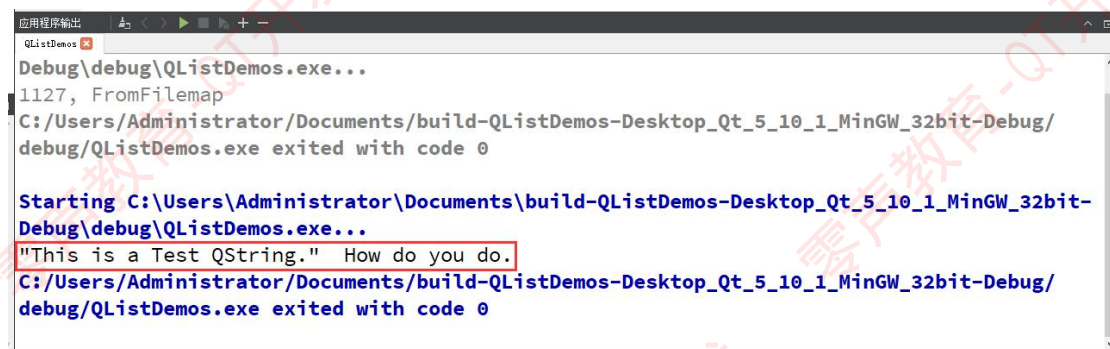
```
#include widget.h  
#include <QApplication>  
#include <QDebug>  
int main(int argc, char *argv[])
```

```

{
    QList<QString> list;
    {
        QString str(This is a Test QString.);
        list<<str;
    }
    qDebug()<<list[0]<< How do you do.;
    return 0;
}

```

运行结果如下：



其实还有 **QLinkedList** 类和 **QVector** 类，具体实际我们可以根据使用的情况决定。**QMap** 类和 **QHash** 类。

**QMap** 类中插入、遍历和修改，具体操作如下：

```

#include widget.h
#include <QApplication>
#include <QDebug>
int main(int argc, char *argv[])
{
    QMap<QString,QString> map;
    map.insert(Beijing,1001);
    map.insert(Shanghai,1002);
    map.insert(TanJing,1003);
    QMapIterator<QString,QString> i(map);
    for(;i.hasNext();){
        qDebug()<< <<i.key()<<-----><<i.next().value();
        QMapIterator<QString,QString> mi(map);
        if(mi.findNext(1003))
            mi.setValue((1005));
        QMapIterator<QString,QString> modi(map);
        qDebug()<< ;
        for(;modi.hasNext();){
            qDebug()<< <<modi.key()<<-----><<modi.next().value();
        }
    }
    return 0;
}

```

}

运行效果如下：



```
Debug\debug\QListDemos.exe...
"Beijing " ----> "1001"
"Shanghai" ----> "1002"
"TanJing " ----> "1003"

"Beijing " ----> "1001"
"Shanghai" ----> "1002"
"TanJing " ----> "1005"
C:/Users/Administrator/Documents/build-QListDemos-Desktop_Qt_5_10_1_MinGW_32bit-Debug/
debug/QListDemos.exe exited with code 0
```

### C/C++开发工程师最基本条件：

- 1、掌握 C++ 语言（面向对象设计分析能力）；
- 2、熟悉 Windows/Linux 编程环境；
- 3、数据结构算法；
- 4、熟悉 C/S 架构的 Qt 跨平台开发；
- 5、熟悉数据库：MySQL、SQLServer、Oracle。

桌面级开发领域、客户端通信系统，VIP 系统学习班里面项目实战：在线网盘系统、即时通信系统。

- 1、C/C++企业级项目实战开发课程
- 2、Linux 服务器后台开发课程
- 3、Linux 内核源码分析课程
- 4、音视频开发课程
- 5、Golang 云原生工程师课程