

2009-常用算法及正则表达式--零声教育 vico 老师

一、【常用算法】

Qt5 常用算法：

1.double c=qAbs(a)：数 qAbs()返回 double 型数值 a 的绝对值；

2.double max=qMax(b,c)：函数 qMax()返回两个数值中的最大值；

3.int bn=qRound(b)：返回一个与浮点数最接近的整数值（四舍五入）；

4.int cn=qSwap(bn,cn)：交换两数的值；

【案例分析】

```
#include "widget.h"
```

```
#include <QDebug>
```

```
Widget::Widget(QWidget *parent)
```

```
    : QWidget(parent)
```

```
{
```

```
    double x=-98.654, y=25.98;
```

```
    double result1=qAbs(x);
```

```
    qDebug()<<"result1="<<result1;
```

```
    double maxresult=qMax(x, y);
```

```
    qDebug()<<"maxresult="<<maxresult;
```

```
    int result2=qRound(y);
```

```
    qDebug()<<"result2="<<result2;
```

```
    int result3=qRound(x);
```

```
    qDebug()<<"result3="<<result3;
```

```
    qSwap(x, y);
```

```
    qDebug()<<endl;
```

```
    qDebug()<<x<<"", "<<y<<endl;
```

```
}
```

```
Widget::~Widget()  
{  
}
```

二、【正则表达式】

正则表达式，又称规则表达式，(Regular Expression ，在代码中常简称为 regex、regexp 或 RE) ，是一种文本模式，包括普通字符（例如，a 到 z 之间的字母）和特殊字符（称为"元字符"）。正则表达式使用单个字符串来描述、匹配一系列匹配某个语法规则的字符串，通常被用来检索、替换那些符合某个模式（规则）的文本。正则表达式描述一种字符串匹配的模式（pattern），可以用来检查一个串是否含有某种子串、将匹配的子串替换或者从某个串中取出符合某个条件的子串等。

正则表达式由表达式(expressions)、量词(quantifiers)、断言(assertions)组成。

（1）最简单的表达式是一个字符。字符集可以使用表达式如"[AEIOU]"，表示匹配所有的大写元音字母；使用"[^AEIOU]"，表示匹配所有非元音字母，即辅音字母；连续的字符集可以使用表达式如"[a-z]"，表示匹配所有的小写英文字母。

（2）量词说明表达式出现的次数，如"x[1,2]"表示"x"可以至少有一个，至多有两个。

正则表达式的量词：

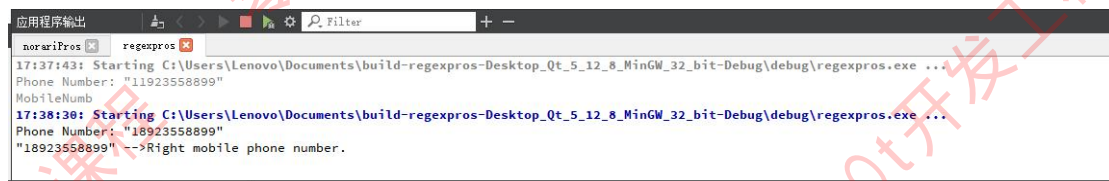
量词	含义	量词	含义
E?	匹配 0 次或 1 次	E[n,]	至少匹配 n 次

量词	含义	量词	含义
E+	匹配 1 次或多次	E[, m]	最多匹配 m 次
E*	匹配 0 次或多次	E[n, m]	至少匹配 n 次，最多匹配 m 次
E[n]	匹配 n 次		

正则表达式的断言：

符号	含义	符号	含义
^	表示在字符串开头进行匹配	\B	非单词边界
\$	表示在字符串结尾进行匹配	(?=E)	表示表达式后紧随 E 才匹配
\b	单词边界	(?!E)	表示表达式后不跟随 E 才匹配

【案例分析】



```

应用程序输出
norariPros  regexpros
17:37:43: Starting C:\Users\Lenovo\Documents\build-regexpros-Desktop_Qt_5_12_8_MinGW_32_bit-Debug\debug\regexpros.exe ...
Phone Number: "11923558899"
MobileNumb
17:38:30: Starting C:\Users\Lenovo\Documents\build-regexpros-Desktop_Qt_5_12_8_MinGW_32_bit-Debug\debug\regexpros.exe ...
Phone Number: "18923558899"
"18923558899" -->Right mobile phone number.
  
```

```
#include "mainwindow.h"
```

```
#include <QDebug>
```

```
#include <regex>
```

```
#include <QString>
```

```
MainWindow::MainWindow(QWidget *parent)
```

```
: QMainWindow(parent)
```

```
{
```

```

/*
 * 通过正则表达式匹配：手机号码
 * 根据目前车家对手机号码规定：11 位数字，其中各段有不同的编码方式；
 * 前三位：网络识别号（中国移动、中国联通、中断电信）
 * 后面第 8 位至 11 位为用户号码
 * 中国移动：134 159 158 188
 * 中国联通：130 133 189 156
 *
 * 相当以 1 开头，第 2 位 3 5 8，共计有 11 位。
 */

QString qMobileNumber="18923558899";
std::regex reg("^1(3|5|8)\\d{9}$");
std::string UserTelIdString=qMobileNumber.toStdString();

QDebug()<<"Phone Number:"<<qMobileNumber;

// 进行匹配
bool bResult=std::regex_match(UserTelIdString, reg);

if(!bResult)
{
    qDebug()<<"MobileNumber"<<"-->Error mobile phone number.";
}
else
{
    qDebug()<<qMobileNumber<<"-->Right mobile phone number.";
}

}

MainWindow::~MainWindow()
{
}

```