基于QT的简易计算器

目　　录

一、需求分析：……3

二、程序的主要功能：……3

三、程序运行平台：……3

四、系统总框架图：……3

五、程序类的说明：　……4

六、模块分析：……6

七、参考资料：……9

八、存在的不足与设计心得……9

九、程序源代码：……10

一、需求分析

为学习可视化编程，特设计此系统。由于 c++程序的可移植性和可维护性较强，且数据比较安全，所以采用 c++进行设计。且将基于跨平台 c++图形用户界面应用程序开发框架 QT 进行开发，便于后期维护及二次开发。

二、程序的主要功能

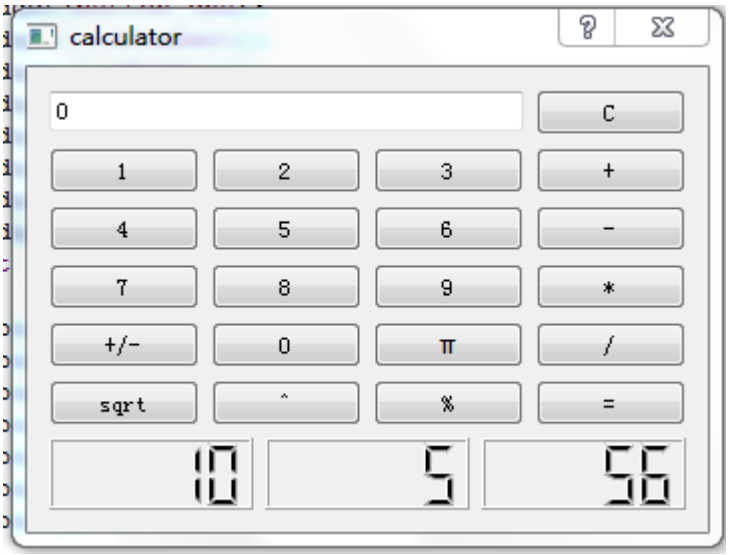
（1）算术型计算器功能——可进行**加（+），减（—），乘（*），除（/），开方（sqrt），取余（%）**等简单运算。

（2）数字时钟功能——为进一步学习 qt，了解其优美的 GUI，以及丰富的 API，加入**数字时钟**模块深入学习。

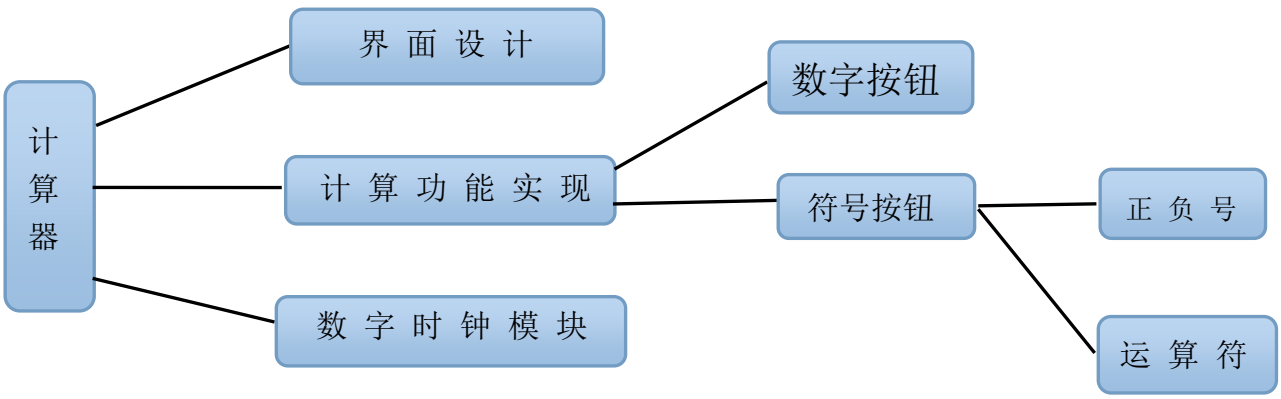
三、程序运行平台

QT 5.1　　IDE:Qt Creator

运行程序“calculator.exe”，进入登录界面如下



四、 系统总框架图



五、程序类的说明

```
class Calculator:public QDialog    //计算模块核心类
{
    Q_OBJECT                    //宏
protected:
    QLineEdit *lineEditor;
    QPushButton *button_0;
    QPushButton *button_1;
    QPushButton *button_2;
    QPushButton *button_3;
    QPushButton *button_4;
    QPushButton *button_5;
    QPushButton *button_6;
    QPushButton *button_7;
    QPushButton *button_8;
    QPushButton *button_9;
    QPushButton *button_pi;
    QLineEdit *time;
    QPushButton *button_sqrt;
    QPushButton *button_chengfang;
    QPushButton *button_mod;
    QPushButton *button_ce;
    QPushButton *button_jia;
    QPushButton *button_jian;
    QPushButton *button_cheng;
    QPushButton *button_chu;
    QPushButton *button__;
    QPushButton *button_dengyu;
    QTimer * timer;
    QLCDNumber * lcdNumber1;
    QLCDNumber * lcdNumber2;
    QLCDNumber * lcdNumber3;

    int num1,num2;
    float result;     //标记第一个数，第二个数和结果
    bool zhenghao;          //数字前面的正负号
    int mark;             //标记是第一个还是第二个数字
    char fuhao;         //运算符号
    QString S;          //显示的字符串
    char a[100];
public:
  //    explicit clock(QWidget *parent = 0);
    Calculator();
private slots:
```

```
    void timeChange();
    void button_0_clicked();
    void button_1_clicked();
    void button_2_clicked();
    void button_3_clicked();
    void button_4_clicked();
    void button_5_clicked();
    void button_6_clicked();
    void button_7_clicked();
    void button_8_clicked();
    void button_9_clicked();
    void button_pi_clicked();
    void button_sqrt_clicked();
    void button_chengfang_clicked();
    void button_mod_clicked();
    void button_ce_clicked();
    void button_jia_clicked();
    void button_jian_clicked();
    void button_cheng_clicked();
    void button_chu_clicked();
    void button__clicked();
    void button_dengyu_clicked();
};

class clock : public QDialog          //数字时钟模块
{
    Q_OBJECT

public:
    explicit clock(QWidget *parent = 0);
    ~clock();

public slots:
    void timeChange();

public:
    QTimer * timer;
    QLCDNumber * lcdNumber;
};
```

六、模块分析

1. 界面设计

因为是第一次接触使用 QT，所以未采用 QT Designer，界面仍然用代码编写。不然我觉得还能更加美观。排版代码如下：

```
QHBoxLayout *Hl1=new QHBoxLayout;          //界面总共 7 行
QHBoxLayout *Hl2=new QHBoxLayout;
QHBoxLayout *Hl3=new QHBoxLayout;
QHBoxLayout *Hl4=new QHBoxLayout;
QHBoxLayout *Hl5=new QHBoxLayout;
QHBoxLayout *Hl6=new QHBoxLayout;
QHBoxLayout *Hl7=new QHBoxLayout;
Hl1->addWidget(lineEditor);                //向每行中添加元件
Hl1->addWidget(button_ce);
Hl2->addWidget(button_1);
Hl2->addWidget(button_2);
Hl2->addWidget(button_3);
Hl2->addWidget(button_jia);
Hl3->addWidget(button_4);
Hl3->addWidget(button_5);
Hl3->addWidget(button_6);
Hl3->addWidget(button_jian);
Hl4->addWidget(button_7);
Hl4->addWidget(button_8);
Hl4->addWidget(button_9);
Hl4->addWidget(button_cheng);
Hl5->addWidget(button__);
Hl5->addWidget(button_0);
Hl5->addWidget(button_pi);
Hl5->addWidget(button_chu);
Hl7->addWidget(lcdNumber1);
Hl7->addWidget(lcdNumber2);
Hl7->addWidget(lcdNumber3);
Hl6->addWidget(button_sqrt);
Hl6->addWidget(button_chengfang);
Hl6->addWidget(button_mod);
Hl6->addWidget(button_dengyu);
QVBoxLayout *V1=new QVBoxLayout;      //程序中只用到一个窗体

V1->addLayout(Hl1);                        //向窗体中添加内容
V1->addLayout(Hl2);
V1->addLayout(Hl3);
V1->addLayout(Hl4);
V1->addLayout(Hl5);
V1->addLayout(Hl6);
```

```
V1->addLayout(Hl7);
```

2. 事件监听

用户界面和运算系统的交互。Qt 中信号和槽用于对象间的通讯。信号/槽机制是 Qt 的一个中心特征并且也许是 Qt 与 其它工具包的最不相同的部分。我们希望任何一类的对象可以和其它对象进行通讯。代码如下：

```
connect(timer, SIGNAL(timeout()), this, SLOT(timeChange()));
                                    //数字时钟模块，通过 timeChange()函数实现动态更新
 /*************************数字按键*********************************/
connect(button_pi,SIGNAL(clicked()),this,SLOT(button_pi_clicked()));
connect(button_0,SIGNAL(clicked()),this,SLOT(button_0_clicked()));
connect(button_1,SIGNAL(clicked()),this,SLOT(button_1_clicked()));
connect(button_2,SIGNAL(clicked()),this,SLOT(button_2_clicked()));
connect(button_3,SIGNAL(clicked()),this,SLOT(button_3_clicked()));
connect(button_4,SIGNAL(clicked()),this,SLOT(button_4_clicked()));
connect(button_5,SIGNAL(clicked()),this,SLOT(button_5_clicked()));
connect(button_6,SIGNAL(clicked()),this,SLOT(button_6_clicked()));
connect(button_7,SIGNAL(clicked()),this,SLOT(button_7_clicked()));
connect(button_8,SIGNAL(clicked()),this,SLOT(button_8_clicked()));
connect(button_9,SIGNAL(clicked()),this,SLOT(button_9_clicked()));
/*************************符号按键*********************************/
connect(button_chengfang,SIGNAL(clicked()),this,SLOT(button_chengfang_clicked()));
connect(button_mod,SIGNAL(clicked()),this,SLOT(button_mod_clicked()));
connect(button_jia,SIGNAL(clicked()),this,SLOT(button_jia_clicked()));
connect(button_jian,SIGNAL(clicked()),this,SLOT(button_jian_clicked()));
connect(button_cheng,SIGNAL(clicked()),this,SLOT(button_cheng_clicked()));
connect(button_chu,SIGNAL(clicked()),this,SLOT(button_chu_clicked()));
connect(button_dengyu,SIGNAL(clicked()),this,SLOT(button_dengyu_clicked()));
connect(button_ce,SIGNAL(clicked()),this,SLOT(button_ce_clicked()));
connect(button__,SIGNAL(clicked()),this,SLOT(button__clicked()));
```

3. 运算模块

此模块不再详述，由于程序意在学习 QT 所以仅需实现两个是的运算，所以涉及算法简单。仅给出部分代码，其他按键代码类似。

```
void Calculator::button_1_clicked(){              //数字按键 1
  S+="1";
  lineEditor->setText(S);
  if(mark==1){
       if(zhenghao){
            num1=num1*10+1;
       }else{
            num1=num1*10-1;
```

```
        }
    }else{
        if(zhenghao){
            num2=num2*10+1;
        }else{
            num2=num2*10-1;
        }
    }
}
void Calculator::button_ce_clicked()                //清屏按键 C
{
    zhenghao=false;
    S="";
    lineEditor->setText("0");
    num1=num2=0;
    mark=1;
}
void Calculator::button_jian_clicked(){             //运算符—
    S+="-";
     lineEditor->setText(S);
    zhenghao=true;
    fuhao='-';
    mark=2;
}
```

4. 时钟模块

由用时钟模块与运算模块在同一界面显示，最后将 Clock 类并入 Calculator 类。QT 提供的库相当完善所以此模块最终并不复杂。涉及代码如下：

```
    timer = new QTimer;
    lcdNumber1 = new QLCDNumber;            //声明三个，分别用于显示时、分、秒
    lcdNumber2 = new QLCDNumber;
    lcdNumber3 = new QLCDNumber;
    lcdNumber1->display(QTime::currentTime().hour());       //调用 currentTime()静态函数，返还时
    lcdNumber2->display(QTime::currentTime().minute());     //分
    lcdNumber3->display(QTime::currentTime().second());     //秒
    timer->start(1000);
                //每 1000ms 发出信号，将调用 timeChange()，达到动态更新效果
  void Calculator::timeChange()             //返还系统时间
{
    lcdNumber1->display(QTime::currentTime().hour());
    lcdNumber2->display(QTime::currentTime().minute());
    lcdNumber3->display(QTime::currentTime().second());
}
```

七、参考资料

1.《Qt C++跨平台图形界面程序设计基础》　清华大学出版社 2014 年版
2《Qt 参考文档》　http://www.kuqin.com/qtdocument/classes.html

八、存在的问题与对策、设计心得

（一） 首先要决定程序是基于 Qt 还是 MFC。简单分析一下：

1、语言本身

QT：跨平台，语法结构简单清晰。面向对象的特性体现的比 MFC 明显。代码写起来比较优雅，也就是说上手会快一点。

MFC：在 Windows 平台地位毋庸置疑，QT 在 windows 下基本属于非主流了。个人初步了解感觉 MFC 相较于 QT 是比较杂乱的，可以看下 MFC 以及 QT 的实例代码，直观上就可以了解了。并且要写 MFC 必须要知道 Windows 的消息循环机制。而 QT 开始时并不需要了解 Windows 底层的东西。

2、学习资源

QT：基本上只有官方的官网和 demo 了，相关的开发论坛比 MFC 的少很多，毕竟敲 MFC 的人要多。

MFC：MSDN 资源，但是 MSDN 比较枯燥，且短时间内未必能独自设计完成，相关论坛和那种问答的资源要多很多，遇到的问题都可以看到前辈们的解决方法，这样会缺少独立思考动力。

3、IDE 以及开发配置

是 windows 系统，对于 MFC，一个 Visual Studio 就足够了。QT，用 QT Creator，配置也不会太麻烦。

决定选用 Qt。

（二）对于 Qt 的学习

1.不同目录下编译结果不同的问题：

把上面的 main.cpp 文件到除 Qt 安装目录下（例如 F:\Qt)下：可以通过 make 命令顺利地得到 exe 文件。

当把相同的文件(main.cpp)复制到 Qt 安装目录下(例如 H:\Qt\4.3.0\examples\tutorial\tt)下时，qmake -project 和 qmake 都通过，make 时却出现下面的问题。

解决办法：当把 Makefile.Release 文件中的 LIBS 项：

LIBS=-L"e:\Qt\5.1.1\lib" -L"c:\Program Files\SQLXML 4.0\bin" -L"h:\Qt\4.3.0\lib" -lmingw32 -lqtmain -lQtGui4 -lQtCore4

改成如下情况时：

LIBS=-lmingw32 -lqtmain -lQtGui4 -lQtCore4 -L"e:\Qt\5.1.10\lib" -L"c:\Program Files\SQLXML 4.0\bin" -L"e:\Qt\5.1.1\lib" （即把后面的几项提到前面去）

或者改成：

LIBS = -L"h:\Qt\4.3.0\lib" -lmingw32 -lqtmain -lQtGui4 -lQtCore4（即把中间的两项去掉）

下面两种情况却可以顺利通过编译。

2.对于信号与槽机制的学习：

信号和槽机制是 Qt 编程的基础。它可以让程序员把一些互不了解的对象绑定在一起。槽和普通的 C++成员函数几乎是一样的——可以是虚函数；可以被重载；可以是公有的、私有的或保护的，并且也可以被其它 C++成员函数直接调用；还有，它们的参数可以是任意类型。唯一不同的

是：槽还可以和信号连接在一下，在这种情况下，每当发射这个信号的时候就会自动调用这个槽。

3.对于界面编写：

由于第一次接触 Qt，时间仓促，并不知道 Qt 设计器，所以用了代码编写界面，给人感觉类似于 java 的图形界面，极易上手。

4.对于已有类的不熟悉：

这就是因为不熟悉了，首次接触导致对于大多数已提供的类的不熟悉。就拿数字时钟来说，原本准备使用 C 语言中方法调用系统时间，导致代码冗长。后经查阅帮助文档发现已经被封装好了可直接调用。短时间内就把此模块写出。

在编程过程中，系统的复习了一遍 c++的知识，还接触了 Qt 开发，发现自己还是有很多不足的地方。永远不要觉得自满，还有同时承认帮助文档是系统学习相应开发语言的最好老师，同时感谢将帮助文档翻译成中文的前辈。这是个 Qt 初学者的学习设计。仍有很多不足，我会继续学习，我会走的更远。