

# 课 程 设 计 报 告

设计题目：QT 计算器

学生姓名：刘铭源

专    业：软件工程

班    级：1 8 — 4

指导教师：张晶

完成日期：2 0 1 9 . 6 . 2 4

合肥工业大学软件学院

## 一、需求分析

实现一个简单的计算器，可以进行算术表达式的计算；操作运算包含加、减、乘、除、括号、与、或、非，即  $21+3*5*(3+4)$ 、 $1!2$ 、 $3\&4$  等计算表达式。后缀表达式严格的从左到右进行计算；假定输入表达式有空格进行简化处理。

计算器需要有良好的操作界面和报错功能，图形界面需要连接计算器进行信号控制，报错需要在计算器进行除数和运算符连续等进行提示，比如在除数为 0 时进行报错题时 error 提示。

## 二、设计

根据上述需求分析，需要设计一个交互良好的计算器，设计图形界面与计算器模型相连接；将图形界面的文本进行传值，进入计算器 mode 进行计算器。

### 计算器 mode:

1. 设立两个堆栈，一个用来保存运算符，另一个用来保存数字。
2. 在表达式首尾添加标记运算符，该运算符运算优先级最低。
3. 从左至右依次遍历字符串，若遍历到运算符，则将其与运算符栈栈顶元素比较，若运算符栈栈顶运算符优先级小于该运算符或者此时运算符栈为空，则将运算符压入堆栈。遍历字符串中下一个元素。
4. 若运算符栈栈顶运算符优先级大于该运算符，则弹出该栈顶运算符，再从数字栈中依次弹出两个栈顶数字，完成弹出的运算符对应

的运算得到结果后，再将该结果压入数字栈，重复比较此时栈顶运算符与当前遍历到的运算符优先级，视其优先级大小重复步骤(3)或步骤(4)。

5. 若遍历到表达式中的数字，则直接压入数字栈。

6. 若运算符堆栈中仅有两个运算符且栈顶元素为我们人为添加的标记运算符，那么表达式运算结束，此时数字栈中唯一的数字即为表达式的值。

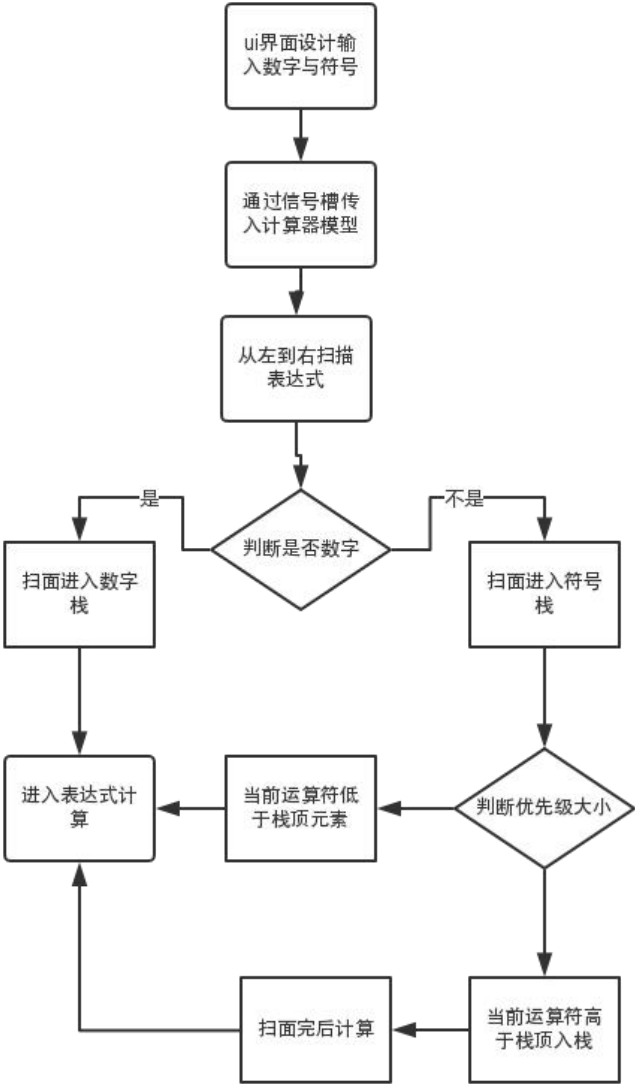
### **图形界面：**

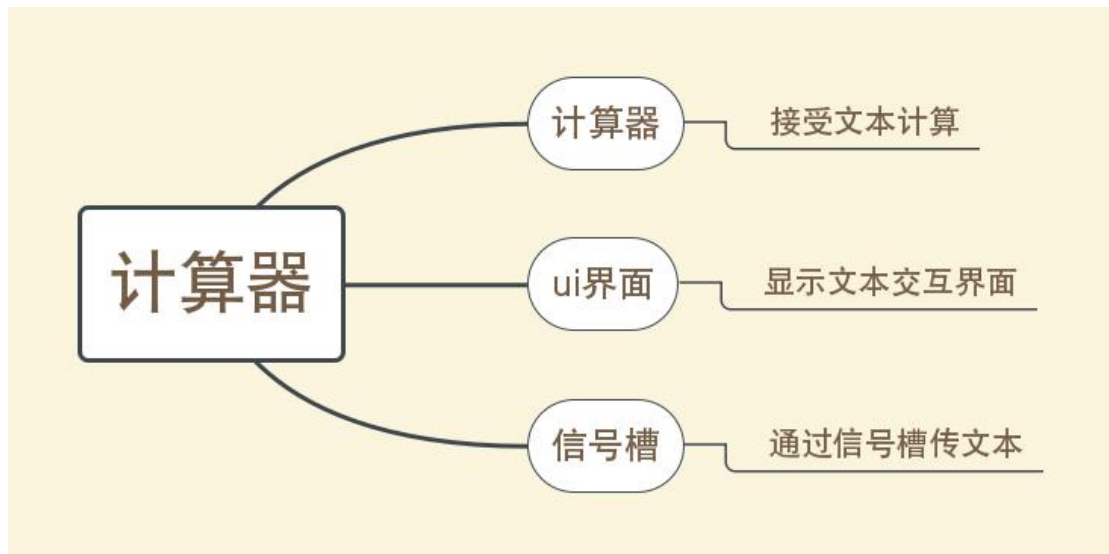
1. 使用 label 进行文本的显示，使用信号槽与计算器连接、数字按键连接，实时将文本显示，传入计算器。

2. 使用 pushbutton 按键进行数字文本输入和符号输入，将文本传入 label 显示和将信息传入计算器中。

3. 使用信号槽进行回传计算值或者进行报错计算。

### 三、系统框图

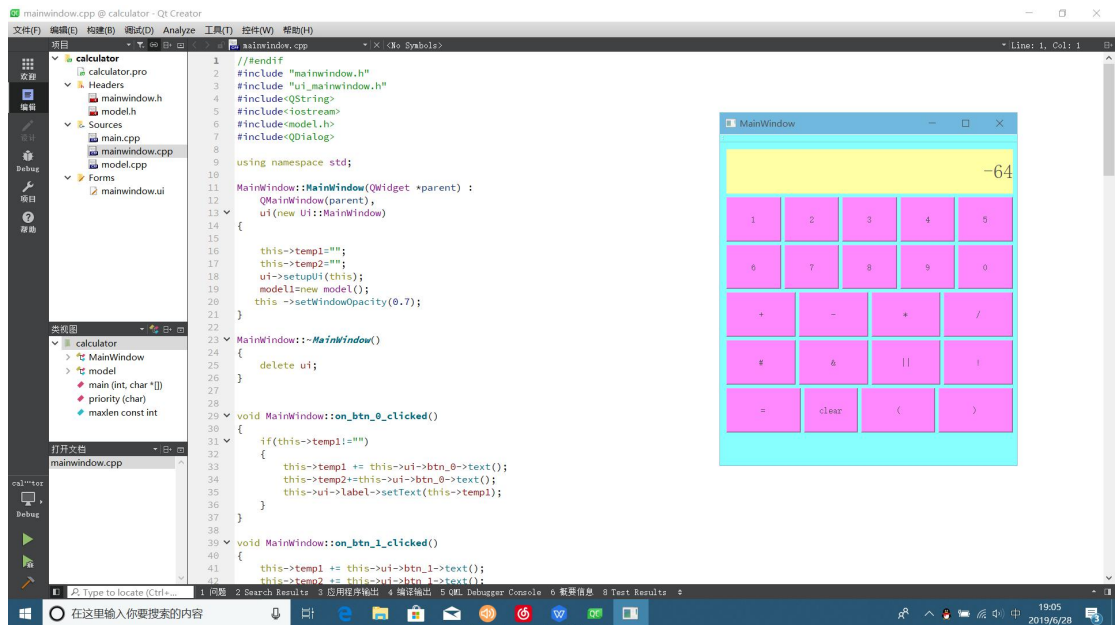
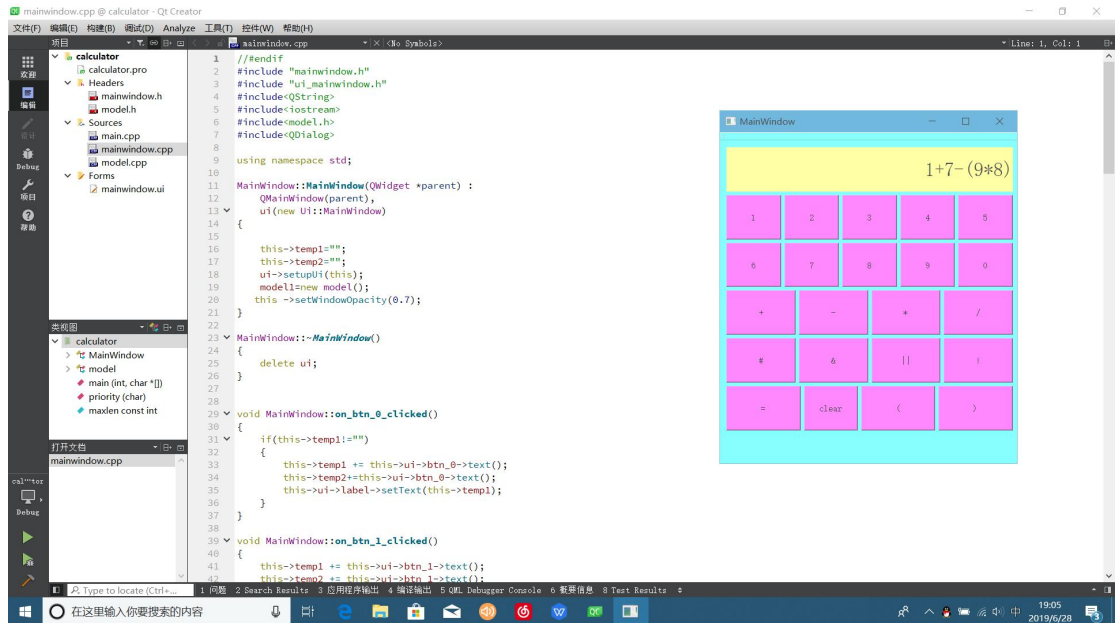




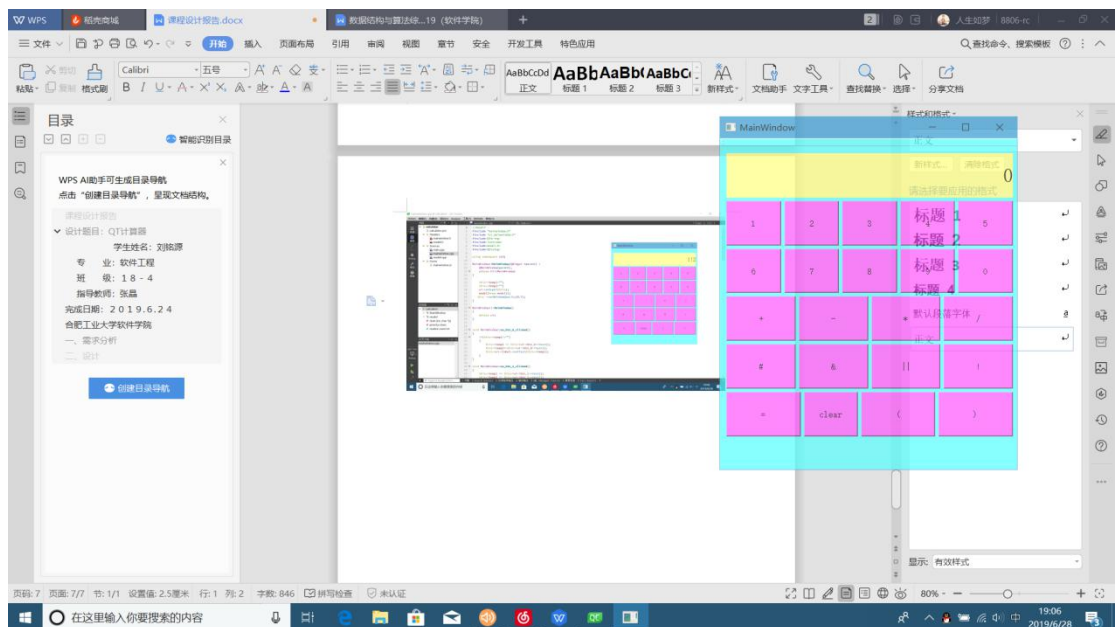
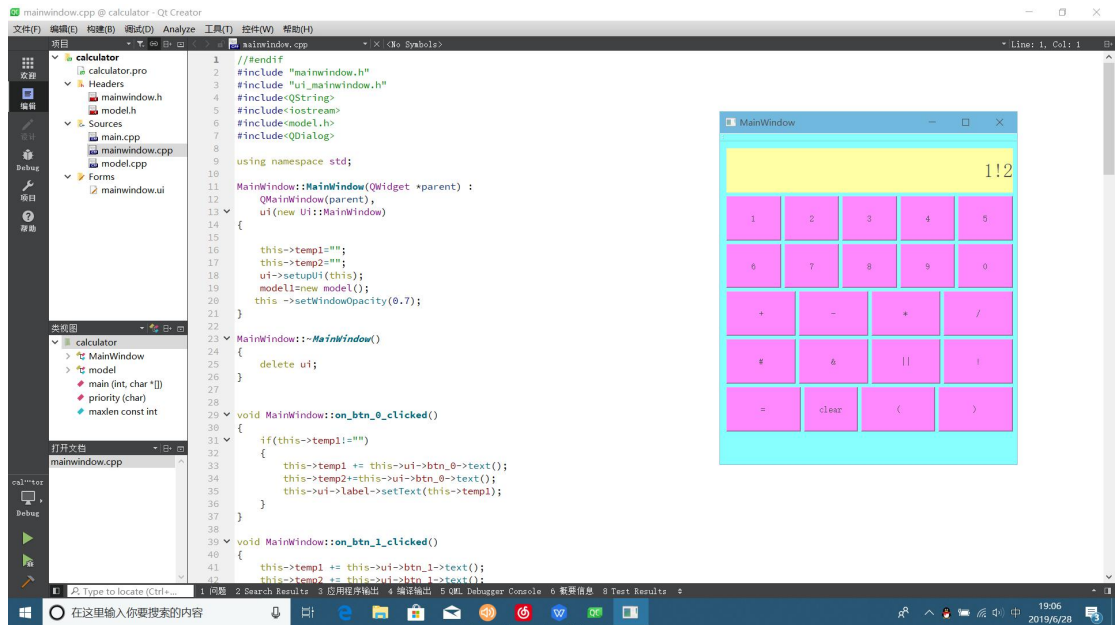
#### 四、属性与方法定义

类	函数名称	作用
Mode	Int Priority	定义优先级
Mode	Void Calculator	计算器函数
Mode	Bool Isnumber	判断数字与字符
Mainwindow	Mainwindow	构造函数初始化
Mainwindow	Void on_btn_clicked	信号槽函数

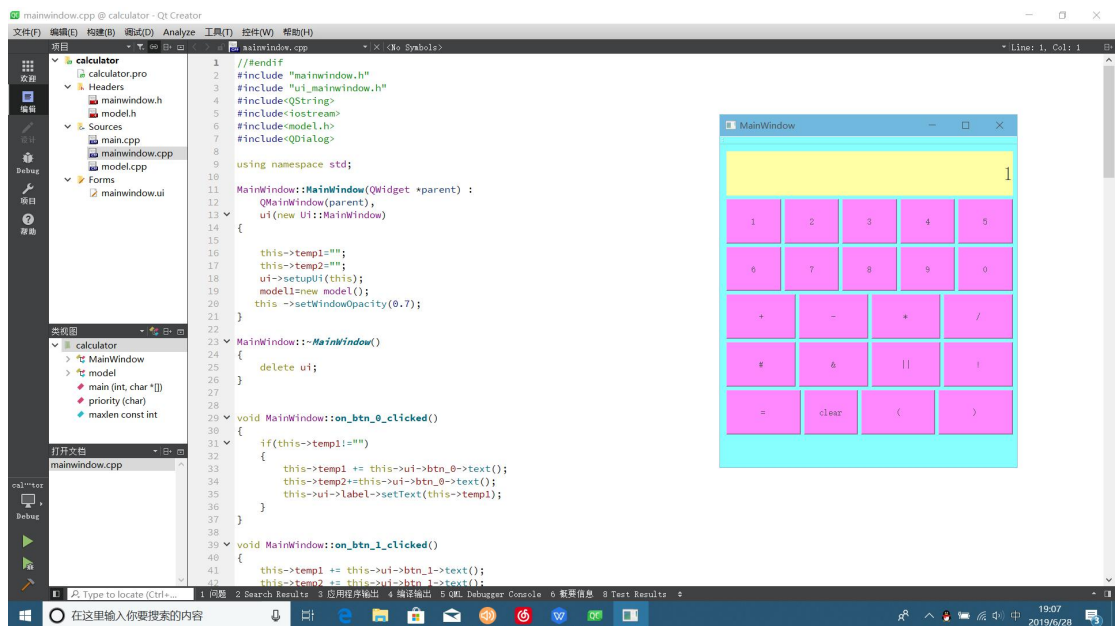
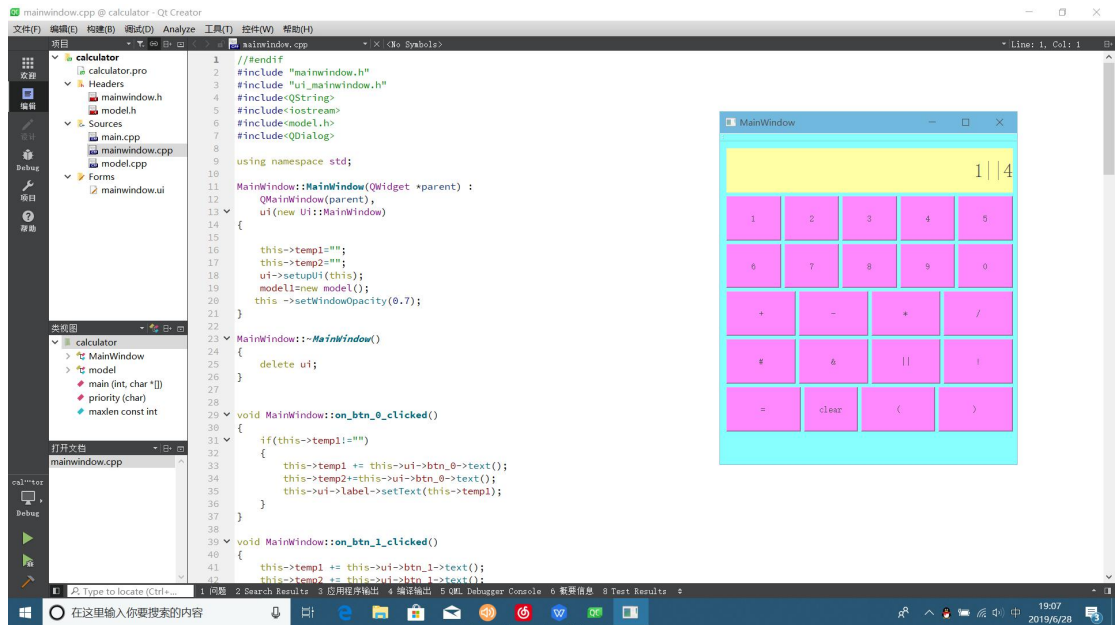
## 五、调试与演示











## 六、参考资料

Csdn--QT 如何实现计算器

Csdn--QT 如何编写图形界面

Csdn--栈的运用

Csdn--如何使用信号槽机制

Csdn--如何改变 QT 的透明度

QT 快速入门

逻辑运算

## 七、感悟

通过这次 QT 计算器的编写，我熟练掌握了 QT 的使用，可以通过后台程序完成对图形界面的设计，通过图形界面向后台程序进行传值计算。通过这次课程设计，我复习了栈的运用，重新完成第二次作业下的计算器编写，加入报错功能，与或非等计算功能，重新温习以前的课程。

## 八、关键源代码

信号槽函数

```
void MainWindow::on_btn_sum_clicked()
{
    this->model1->get_str(this->temp2);
```

```

QString p=this->modell->calculator();

this->templ=p;

this->temp2=p;

this->ui->label->setText(this->templ);

}

```

判断是不是数字

```

bool model::isnumber(char x) {

    if (x >= '0' && x <= '9')return true;

    else return false;

}

```

优先级定义

```

int priority(char x) {

    if (x == '+' || x == '-')

        return 1;

    else if (x == '*' || x == '/')

        return 2;

    else if (x == '!')

        return 3;

    else if (x == '(' || x == ')')

        return -1;

    else if (x == '&' || x == '|')

        return -2;

```

```

        else if (x == '#')

            return -3;

        else return 0;

    }

```

计算器

```

QString model::calculator()

{

    string s;

    s=this->s1.toString();

    QStack<int>number;

    QStack<char>operate;

    char top;

    int a, b;

    s='#'+s+'#';

    for (unsigned int i = 0; i < s.size(); ++i)

    {

        if (isnumber(s[i]))

        {

            int Temp = 0;

            string temp;

```

```

temp += s[i];

while (isnumber(s[++i]))

    temp += s[i];

for (unsigned int j = 0; j < temp.size(); ++j)

{

    Temp = Temp * 10 + temp[j] - 48;

}

number.push(Temp);

i--;

}

if (!isnumber(s[i]))

{

    if (s[i] == '!')

    {

        i++;

        if (isnumber(s[i]))

        {

            int Temp = 0;

            string temp;

            temp += s[i];

            while (isnumber(s[++i]))

```

```

        temp += s[i];

        for (unsigned int j = 0; j < temp.size();
++j)

        {

            Temp = Temp * 10 + temp[j] - 48;

        }

        number.push(Temp);

        i--;

    }

    a = number.top();

    number.pop();

    a = !a;

    number.push(a);

}

else if (operate.empty())

{

    operate.push(s[i]);

}

else

{

    top=operate.top();

```

```

        if (priority(s[i])>priority(top) || s[i] ==
' (')

        {

            operate.push(s[i]);

        } //入栈高优先级运算符。

        else

        {

            while (priority(s[i]) <= priority(top))

            {

                if (top == '#' && s[i] == '#')

                {

                    return

QString::number(number.top());

                }

                else if (top == ' (' && s[i] == ')') {

                    ++i;

                }

                else if(top==s[i])

                {

                    return "erro";

                }

                else {

```

```
    a=number.top();  
    number.pop();  
    b=number.top();  
    number.pop();  
}
```

```
if (top == '+') {  
    b += a;  
    number.push(b);  
}  
else if (top == '-') {  
    b -= a;  
    number.push(b);  
}  
else if (top == '*') {  
    b *= a;  
    number.push(b);  
}  
else if (top == '/') {  
  
    if(a!=0)
```



```

        {

            b=b/a;

        }

        else {

            return "erro";

        }

    }

    else if (top == '&') {

        b = a && b;

        number.push(b);

    }

    else if (top == '|') {

        b = a || b;

        number.push(b);

    }


    operate.pop();

    top=operate.top();

}

```

```
operate.push(s[i]);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
}
```