



# 计算机综合课程设计项目报告

COMPUTER INTEGRATED CURRICULUM DESIGN PROJECT REPORT

设计题目： 简易计算器

报 告 人： 钟源 学号： 04022212

## 1.最终效果:

本实验的选题是简易计算器，共有 39 个按钮、2 个编辑框、1 个标题栏及一张背景图片，最后 exe 文件在 5MB 左右，创建的主要类是 CcalculatorDlg 类，其中核心函数编写在 calculate.cpp 文件中。

项目最终实现的功能有：

- (1) 可以输入整数和浮点数——既可以通过按钮输入，也能够直接在编辑框中输入；
- (2) 可以逐位回退数据——按 DEL 键；
- (3) 可以清空数据——按 AC 键；
- (4) 带有记忆功能：计算结果可以加入下一次计算——按 ANS 键；
- (5) 计算时判断输入数据的正确性，输入有误的给出提醒；
- (6) 计算功能包括但不限于：加、减、乘、除、正弦函数、余弦函数、正切函数、余切函数、反正弦函数、反余弦函数、反正切函数、倒数、平方根、立方根、平方、立方、log、log10、ln、幂；
- (7) 能够识别运算的优先级，其中 log 运算具有最高优先级；
- (8) 遇到浮点数，根据其位数进行输出，若是无限位，则输出 9 位。
- (9) 可输入  $\pi$ （取 3.14159265358979323846）和  $e$ （取 2.71828182845904523536）





图 1. 主界面



图 2. 记忆功能



图 3. 给出错误提示

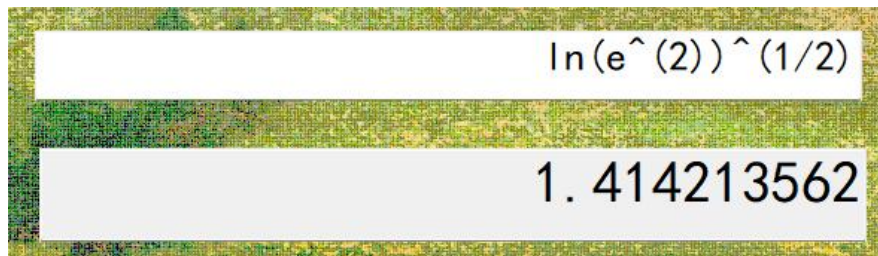


图 4. 函数运算



图 5. 识别运算优先级



图 6. 识别浮点数位数-1



图 7. 识别浮点数位数-2

## 2.算法实现:

(1) 在头文件里添加相应成员变量（如 mStr 记录输入字符，result 记录输出结果等），用于底层运算。

(2) 完成前端设计，并添加相应按钮函数，并写好相应函数内容。

部分函数代码（以添加 $\pi$ 的按钮函数为例）如下：

```
void CcalculatorDlg::OnBnClickedButtonpai()
{
    UpdateData(TRUE);
```

```

    if (flag)
    {
        CcalculatorDlg::OnBnClickedButtonac();
        flag = false;
    }
    mStr += L"π";
    UpdateData(FALSE);
}

```

(3) 自定义一些算法函数，为之后的输出结果准备。

部分函数代码（以识别浮点数位数的函数为例）如下：

```

int WeiShu(double num) {
    int result = 0;
    int temp = floor(num);
    num *= 10;
    temp *= 10;
    int xiaoshu = num - temp;
    while (xiaoshu > 0 && xiaoshu < 10 && result <= 12) {
        result++;
        temp = floor(num);
        num *= 10;
        temp *= 10;
        xiaoshu = num - temp;
    }
    return result;
}

```

(4) 定义核心结构体——PARENTHESIS\_LEFT和NUMBER。前者用于储存左括号位置、绑定的函数、匹配的右括号的位置；后者则将算式抽象化为链表，链表的节点记录了当前运算数的数值、运算数后的运算符以及该运算符的优先级。

相关代码如下：

```

struct PARENTHESIS_LEFT
{
    int location;
    CString function;
    int location_right;
};
.....
struct NUMBER
{
    //

```



```

double num;
CString opt;
int precedence;//优先级
std::shared_ptr<NUMBER> next;//链表的节点定义
};

```

(5) 编写核心函数——operate(匹配运算符对应的运算)、function(匹配各个函数对应的运算)、check(检查括号是否错误)、paired(子字符串的括号绑定函数)、calculate(计算结果的函数)。

部分函数代码（以 calculate 的部分代码为例）如下：

```

if (mStr_[i] == L'(')
{
    NUMBER num_now;
    int j = 0;
    for (j = 0; j < left_parenthese; j++)//获得当前所对应的括号
    {
        if (parenthese_left_[j].location == i)
        {
            break;
        }
    }
    num_now.num = function(parenthese_left_[j].function, calculate(mStr_, i + 1,
    parenthese_left_[j].location_right - i - 1, ans));
    i = parenthese_left_[j].location_right + 1;
    if (i == mStr_.GetLength())
    {
        num_now.opt = L""; //末尾运算符为空，优先级为0
        num_now.precedence = 0;
        all_num.push_back(std::make_shared<NUMBER>(num_now));//@将结果加到链表
        中
        break;
    }
    else
    {
        if (mStr_[i] == L'+')
        {
            num_now.opt = L"+";
            num_now.precedence = 1;
        }
        if (mStr_[i] == L'-')

```

```

    {
        num_now.opt = L"-";
        num_now.precedence = 1;
    }
    if (mStr_[i] == L'*)
    {
        num_now.opt = L"*";
        num_now.precedence = 2;
    }
    if (mStr_[i] == L'/')
    {
        num_now.opt = L"/";
        num_now.precedence = 2;
    }
    if (mStr_[i] == L'^')
    {
        num_now.opt = L"^";
        num_now.precedence = 3;
    }
    if (mStr_[i] == L'l')
    {
        num_now.opt = L"log";
        num_now.precedence = 4;
    }
    all_num.push_back(std::make_shared<NUMBER>(num_now));//@将结果加到链
    }
}

```

(6) 在=键对应函数OnBnClickedButtonequal()中调用各函数，最终实现结果输出。

相关代码如下：

```

void CcalculatorDlg::OnBnClickedButtonequal()
{
    UpdateData(TRUE);
    if (!mStr.IsEmpty()) //空字符串不做处理
    {
        if (check()) //先检查
        {
            ans = calculate(mStr, 0, mStr.GetLength(), ans);//再计算
            result = strCreate(ans);
        }
    }
}

```

```
        //默认的12位输出精度，多了有误差  
        flag = true;  
    }  
}  
UpdateData(FALSE);  
}
```