

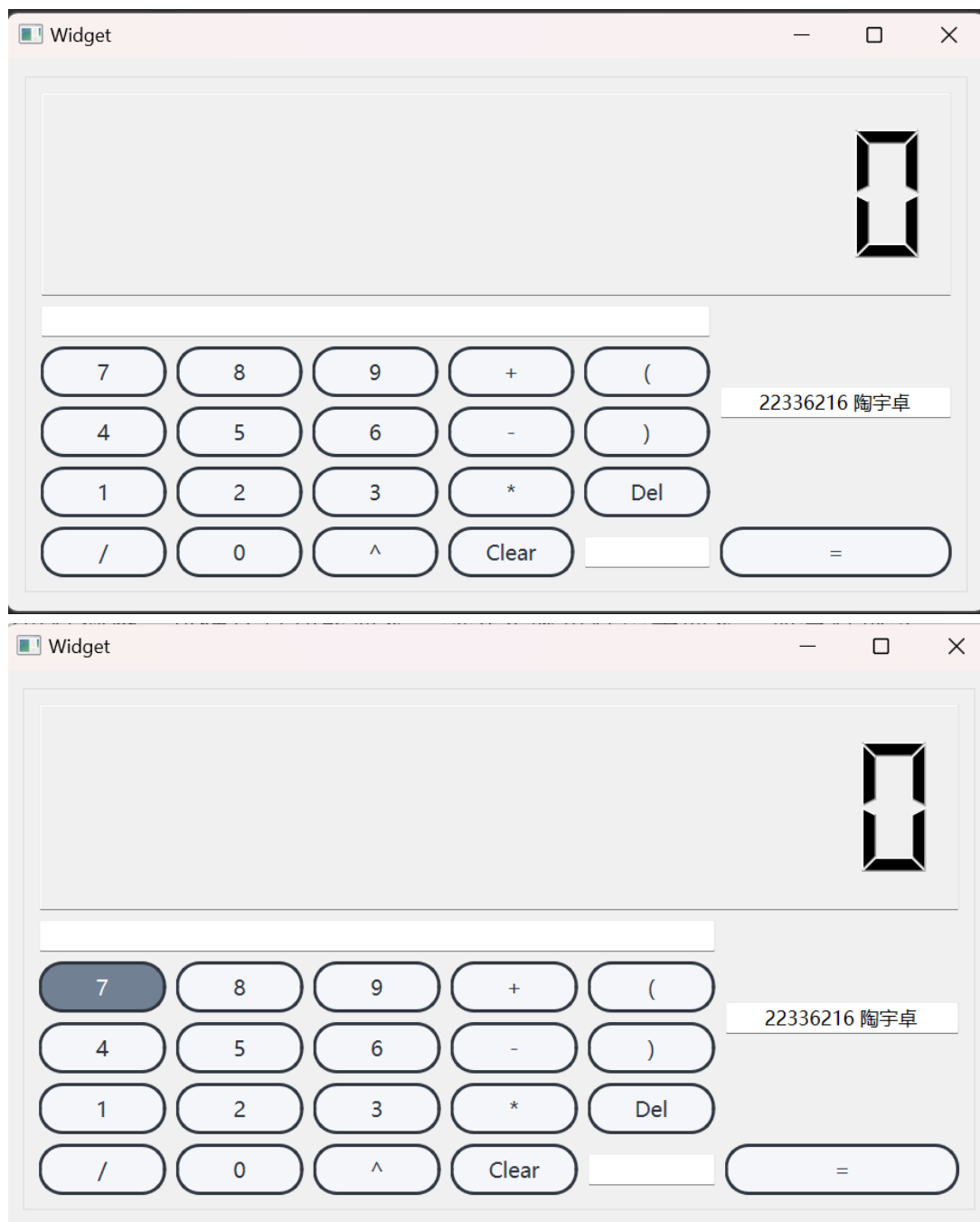
Project2 实验报告

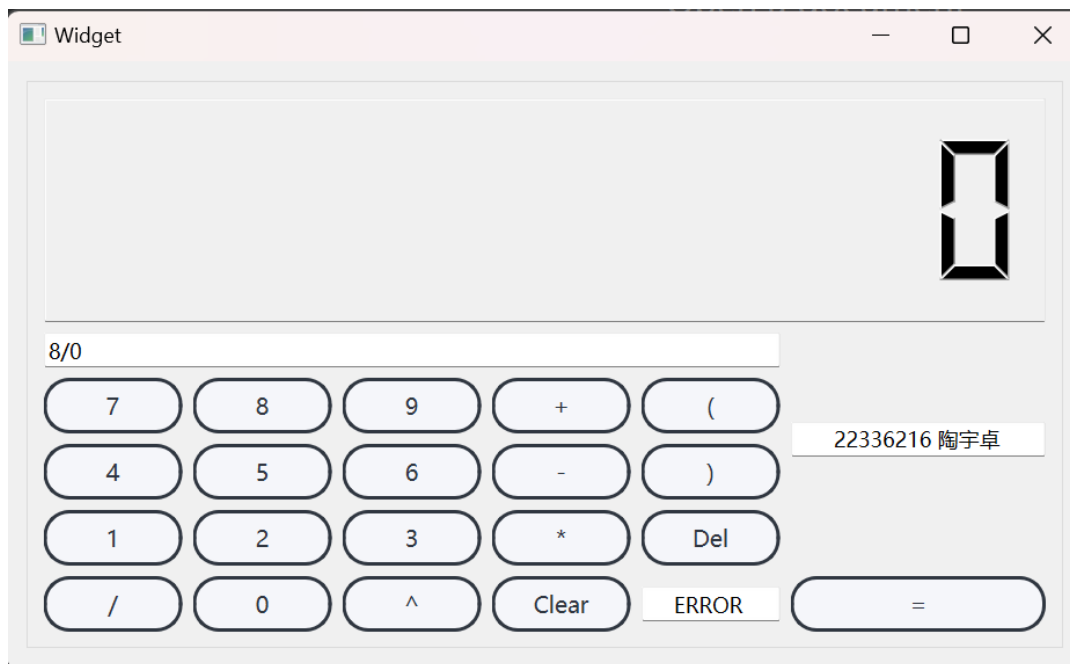
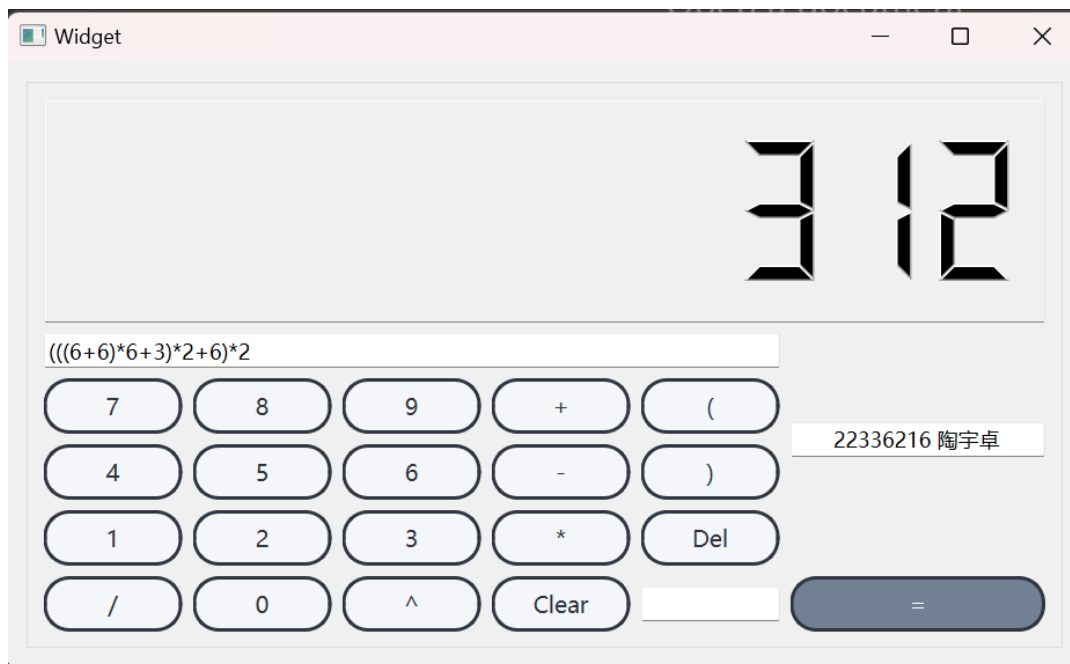
22336216 陶宇卓

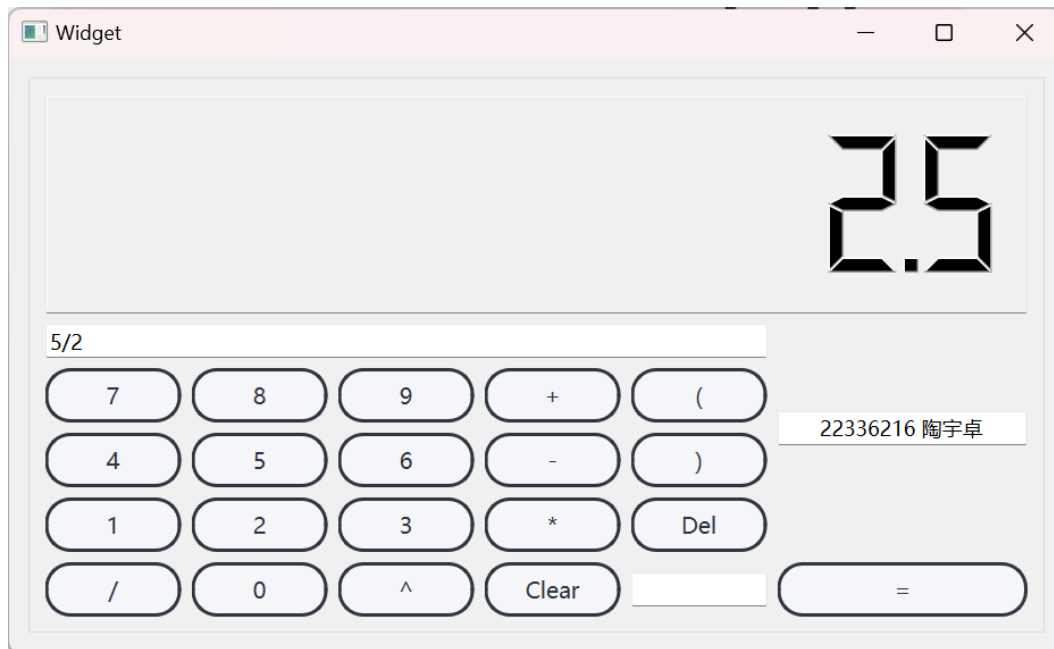
1、 程序功能简要说明。

本程序使用 Qt6.2.4 编译，可以对输入的表达式进行求值，并支持除法报错和结果以小数形式输出。

2、 程序运行截图，包括计算功能演示、部分实际运行结果展示、命令行或交互式界面效果等。







3、部分关键代码及其说明。

```
void Widget::on_pushButton_0_clicked()
{
    s+="0";
    ui->lineEdit->setText(s);
}
```

按下普通的 0-9 以及 ‘-’ ‘+’ 等符号时，将这个字符压入 s 字符串里，并在 lineEdit 显示。

```
void Widget::on_pushButton_eqL_clicked()
{
    int begin=0;
    s=ui->lineEdit->text();
    double result=calculate(s,begin);
    if(correct==false){
        ui->lineEdit_2->setText("ERROR");
        return;
    }
    ui->lineEdit_2->setText("");
    ui->lcdNumber->display(result);
}
```

在按下等于号按钮时调用 calculate()函数，将结果计算出来并展示到 lcdNumber 上。如果不合法的话，打印“ERROR”到 lineEdit_2 上。

```

double Widget::calculate(QString str,int &i){
    char operation = '+';
    std::string s=str.toString();
    std::stack<double> nums;
    double num = 0.00;
    double res = 0.00;
    correct=true;
    for (i; i < s.size(); i++)
    {
        if (s[i] >= '0' && s[i] <= '9')
        {
            num = num * 10 + (s[i] - '0');
        }
        if (s[i] == '(')
        {
            num = calculate(str, ++i); //从i的下一个开始计算, 进入递归
            i++; //计算完之后的i指向) 所以再++
        }
        if (((s[i] < '0' || s[i] > '9') && s[i] != ' ') || i >= s.size() - 1) // 继续计算
        {
            double pre = 0.00;
            switch (operation)
            {
                case '+': nums.push(num);
                    break;
                case '-': nums.push(-num);
                    break;
                case '*':
                    pre = nums.top();
                    nums.pop();
                    nums.push(pre * num);
                    break;
                case '/':
                    pre = nums.top();
                    nums.pop();
                    if(num==0){
                        correct=false;
                        nums.push(-1);
                    }
                    else nums.push(pre / num);
                    break;
                case '^':
                    pre = nums.top();
                    nums.pop();
                    nums.push(pow(pre,num));
                    break;
            }

            operation = s[i];
            num = 0.00;
        }
        if (s[i] == ')') //遇到) 回到上一级递归
        {
            break;
        }
    }
    while (!nums.empty())
    {
        res += nums.top();
        nums.pop();
    }
    return res;
}

```

计算时，遍历字符串，遇到数字时将结果统计在 num 中，遇到 ‘(’ 递归计算括号内部的值，遇到 ‘)’ 返回上一层。提前储存一个操作符 operation ‘+’，遇到运算符时，按照当前储存的字符进行操作，如压栈等，具体如图。最后

将所有结果相加。

4、程序运行方式简要说明。

其实 Qt 也是 C++语法，他具体的逻辑和运行方式我还没有太熟悉。通过 `push_button` 函数能够将按下的按钮转化为字符压入字符串 `s` 中。真正的核心代码还是 `Calculate` 函数，能实现计算出结果并返回到 `ui` 中的 `lcdNumber` 中。