**测试报告**

**By：211906601-蔡维恒**

1. 测试用例：Task6601\_U1

测试内容：正常使用

输入数据：12+5=清空88\*2=清空34-41=清空42/2=

预期结果：17、176、-7、21

实际运行效果：17、176、-7、21

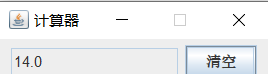
1. 测试用例：Task6601\_U2

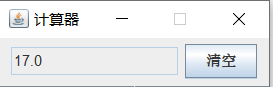
测试内容：连续计算

输入数据：12+2=34/2=

预期结果：14、17

实际运行效果：





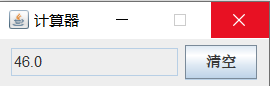
1. 测试用例：Task6601\_U3

测试内容：连续计算

输入数据：12+2=+32

预期结果：14、46

实际运行效果：



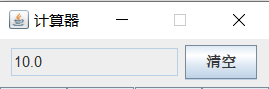
1. 测试用例：Task6601\_U4

测试内容：连续计算

输入数据：4+3+6+4

预期结果：10

实际运行效果：



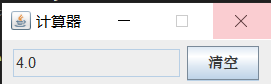
1. 测试用例：Task6601\_U5

测试内容：没有第二个数

输入数据：4+=

预期结果：4（4+0）

实际运行效果：



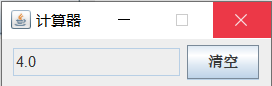
1. 测试用例：Task6601\_U6

测试内容：没有第二个数

输入数据：4\*=

预期结果：4（4\*1）

实际运行效果：



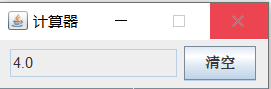
1. 测试用例：Task6601\_U7

测试内容：没有第二个数

输入数据：4\*=4-1

预期结果：4（4\*1）、4（4-0）

实际运行效果：



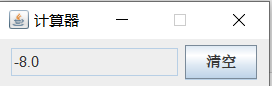
1. 测试用例：Task6601\_U8

测试内容：没有第一个数

输入数据：-8=

预期结果：-8（0-8）

实际运行效果：



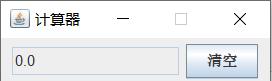
1. 测试用例：Task6601\_U9

测试内容：没有第一个数

输入数据：/8=

预期结果：0（0/8）

实际运行效果：



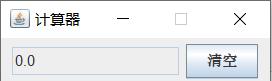
1. 测试用例：Task6601\_U10

测试内容：直接 =

输入数据：=

预期结果：0（0+0）

实际运行效果：



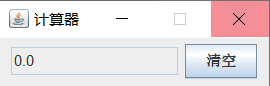
1. 测试用例：Task6601\_U11

测试内容：直接 /=

输入数据：/=

预期结果：0（0/1）

实际运行效果：



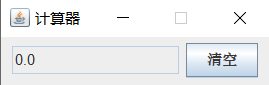
1. 测试用例：Task6601\_U12

测试内容：直接 /=

输入数据：/= +=

预期结果：0（0/1）0（0+0）

实际运行效果：



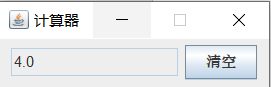
1. 测试用例：Task6601\_U13

测试内容：特殊输入

输入数据：1+2+4+=

预期结果：4（4+0）

实际运行效果：



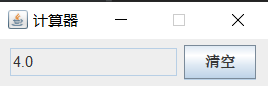
1. 测试用例：Task6601\_U14

测试内容：特殊输入

输入数据：1+2+4\*=

预期结果：4（4\*1）

实际运行效果：



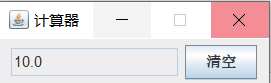
1. 测试用例：Task6601\_U15

测试内容：特殊输入

输入数据：2+-\*5=

预期结果：10（2\*5）

实际运行效果：



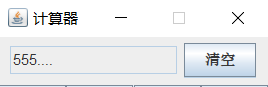
1. 测试用例：Task6601\_U16

测试内容：特殊输入

输入数据：555......+=

预期结果：555......

实际运行效果：



输入的数据中，点的数量超出1个即为非法输入，必须清空，否则将无法进行运算

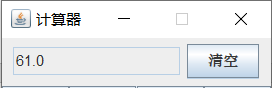
1. 测试用例：Task6601\_U17

测试内容：累加计算

输入数据：55+3==

预期结果：58（55+3）、61（58+3）

实际运行效果：



个人小结：

基本的运算逻辑皆参照了 calc.jar

本次实验只是实现了运算的逻辑，尚且存在诸多需要修改的地方，待后续会一一完善之

需要改进的地方：

1. 用了大量的 Boolean 进行特殊输入的判断
2. 显示的样式，如果结果非浮点数，则输出整型
3. 连续运算的逻辑 1 + 2 + 3 +4 实行累加，而不是替换，可以省去大量的判断工作。即 1 + 2 + 3 + 4 = 10，而非 7
4. 可以输入 000023 或自动识别成 23，个人认为 这也是一种非法的输入，可能无伤大雅，但是不太友好。如果以0为开头 则不能在输入0，如果在0后面输入了数字则替换掉0，即 0123 显示 123即可