Calculator 功能case总结

+-\*/：四则运算 %平方开方倒数：单点运算

注意两个细节：

1.大屏幕上只会出现数字button，小屏幕的最后一个字符一定不是数字button。（不考虑系统错误）

2.%没有其运算表达式，直接呈现运算结果。

三个最主要的全局变量：

1.当前大屏幕数据

2.当前小屏幕数据

3.最后一个四则运算的符号

需要的全局bool变量：

1.大屏幕是否含有小数点

2.大屏幕是否含有负号

3.=是否被清除

4.小屏幕是否含有四则运算

5.是否激活纯单点运算

需要的栈数据结构：（可能不需要找，只需要一个全局变量就足够了）

1.上一次操作类型

2.上一次大屏幕更新数据

3.小屏幕最后一个数据

三种类型的错误：

1.除数为0：（3/0）（0倒数）（Cannot divide by zero）

2.负数求根：（5负号求根）（Invalid input）

3.数值溢出：（5一直平方）（Overflow‬）

出现错误后任何一个错误后单点运算符、四则运算符、小数点、正负号都将被禁用，再进行有效点击后被禁用功能又恢复使用。

（在触发此错误后移去这些button的click事件，有效点击后再全部激活）

（在剩下的按钮实现方法的最开始绑定一个检验错误的方法）

数字按钮的点击限制：

1.数字0的限制：（有效数字法则）

大屏幕的数字从左往右数，遇到第一个非0的数字前（包括小数点）的数字0的个数最多只能有一个。

（每次进行输入数字前判断一下）

2.小数点的限制：大屏幕最多只能存在一个小数点。

（可以用一个全局bool变量来表示其状态）

3.正负号的限制：

纯数字0不给给予正负号，其它情况正负号来回切换不予叠加显示。（正号默认不显示）

（可以用一个全局bool变量来表示其状态）

四则运算按钮的点击限制：

大屏幕不显示符号，小屏幕不允许连续两个四则符号相邻，采取后者迭代前者的方案措施。

（可通过上一次操作的操作类型进行判断）

注意：部分按钮的情况受情形顺序的影响！（主要是=未清除的情形）

不单独考虑C、DEL、CE状态，将其转化为其它已知类型！！！

C：清空小屏幕，大屏幕置0。（全部数据初始化）（为0的数字button类型）

DEL：三种情况：

1.上一次操作是数字button时，小屏幕不变，清空大屏幕最后一位数字，如果是各位数（不包括正负号），直接归0。（5\*65DEL）

（数字button类型，即不改变原始类型）

2.归0后或者上一次操作是四则运算或者单点运算时，显示屏结果不发生改变。

（5\*0DEL）（5+DEL）（5^DEL）

（直接return）（不改变原始类型）

3.如果上一步操作是=或者使用=后未手动清除=即只使用单点运算或者=，小显示屏置空，大显示屏不变。（5+6==DEL）（5+6=平方倒DEL）

（用一个全局bool变量来表示=的清除状态）

（=类型，不改变原始类型，只不过=已被清除）

CE：三种情况：（大屏幕始终置0）（即大屏幕相关数据全部初始化）

1.在四则运算与单点运算的混合使用，并且上一步操作是单点运算时，小屏幕清空上一次的单点运算。（9+6平方CE）

（为0的数字button类型）

（用一个bool变量来表示是否含有四则运算，回退功能，使用栈来实现）

2. 如果上一步操作是=或者使用=后未手动清除=即只使用单点运算或者=，则小屏幕也置空，类似C按钮的效果。（5+6==CE）

（为0的数字button类型）

3.其余情况，小屏幕不做改变。（5+6平方+CE）\

（为0的数字button类型）

+/-：两种情况（大屏幕始终满足正负号限制）

1.（小屏幕无显示文本或者小屏幕有显示文本）但上一步操作的数字button时，

小屏幕显示无变化。（5+6+/-）

2.其余情况，一旦小屏幕上有文本显示，小屏幕会negate()叠加显示。（5++/-）

（用一个栈来存储最后一个小屏幕数据的记录，取出操作然后再放入栈中）

（如果上一步是单点运算，基准就是小屏幕最后一个数据，否则基准就是大屏幕数据）

.：两种情况（即满足小数点限制）（最后一位是小数点默认忽略小数点）

1.如果上一步操作是数字button时，满足小数点限制规则。

2.其余情况，大屏幕内容替换成0.。

3.如果上一步操作是=或者使用=后未手动清除=即只使用单点运算或者=，小屏幕清空。（5+6=.）(CE.)

4.在四则运算与单点运算的混合使用，并且上一步操作是单点运算时，小屏幕清空上一次的单点运算。（5+6平方.）(CE.)

5.其余情况，小屏幕不变。

数字0-9：六种情况

1.如果上一步操作是数字button，则大屏幕上会进行数字叠加，小屏幕显示不变。（数字0需满足数字0的限制法则）（5+65）

2.如果上一步操作是四则元素或者是纯单点运算，大屏幕数字会迭代，小屏幕显示不变。（6+5）（6平方5）

（用一个bool全局变量来代表纯单点运算）

3. 在四则运算与单点运算的混合使用，并且上一步操作是单点运算时，小屏幕会清空上一次单点运算，大屏幕迭代。（类似CEnumber）（9+6平方5）

（参照CE1）

4.如果上一步操作是=或者使用=后未手动清除=即只使用单点运算或者=，大屏幕数字会迭代，小屏幕内容置空。（类似DELnumber）（5+6=平方3）

（参照DEL3）

5.如果上一步操作是C或者CE时，跳转情形1.

6.如果上一步操作DEL，如果是DEL2情形，则忽略；如果是DEL1情形，则跳转情形1；如果是DEL3情形，则跳转情形4.

=：六种情况

1.如果上一步操作是数字button且小屏幕不含四则运算符时，大屏幕不变，小屏幕显示 “大屏幕内容=”。（6=）

2.如果上一步操作是数字button但小屏幕上含有四则运算符时，大屏幕会直接显示运算结果，小屏幕内容显示之前的内容+“大屏幕内容=”。（6+5=）

3.如果上一步操作是四则运算符时，跳转情形2。（类似number=）（6+=）

4.如果小屏幕含有单点运算时，当作正常的数字button进行相应跳转情形1或情形2。（近似直接当成数字button来处理）（6平方=）（6平方+5=）

4.+ 如果上一步操作是单点运算，大屏幕直接显示运算结果，小屏幕显示为前小屏幕结果+“=”。

5.如果上一次操作是=，如果小屏幕含有四则运算符，跳转2继续运算；如果小屏幕不含四则运算符，跳转1。如果含有单点运算符，则全部转换成其对应的数字结果。

6.如果上一次操作是C、CE、DEL，按照小屏幕是否有四则运算进行情形1和2的相应跳转。

四则运算：六种情况（和=情形极为相似）

1.如果上一步操作是数字button且小屏幕不含四则运算符时，大屏幕不变，小屏幕显示 “大屏幕内容+”。（6+）

2.如果上一步操作是数字button但小屏幕上含有四则运算符时，大屏幕会直接显示运算结果，小屏幕内容显示之前的内容+“大屏幕内容+”。（6+5+）

3.如果上一步操作是单点运算，小屏幕上直接拼接对应的四则运算符，而大屏幕的内容按照小屏幕上是否已经有四则运算符来跳转情形1或情形2。

（6平方+）（6+5%+）

4.如果上一次操作符是四则运算符，则遵循四则运算按钮的点击限制。

5.如果上一步操作是=或者使用=后未手动清除=即只使用单点运算或者=，按照是否含有单点运算符来相应跳转情形1或者情形31。（6+5=+）（6+5=平方+）

(DEL+)

6.如果上一次操作是C、CE、DEL，按照小屏幕是否有四则运算进行情形1和2的相应跳转。

单点运算：五种情况（不考虑%的特殊情形）

（=运算后的下一次任何操作，之前存在的单点运算表达式都会消失，拿个bool类型表示是否开启变回数字）

1.如果上一步操作是数字button或者四则运算，大屏幕直接显示单点运算的结果，小屏幕显示拼接大屏幕单点元素的符号表达式。（5平方）（5+6平方）

2.如果上一步是单点运算，大屏幕直接显示单点运算的结果，小屏幕显示单点运算符号表达式的复合拼接。（5+6平方开根号）

3.如果上一步是%，直接跳转情形1. （5+6%平方）

4. 如果上一步操作是=，大屏幕直接显示单点运算的结果，小屏幕显示被大屏幕单点元素的符号表达式迭代。（5+6=平方）

5.如果上一次操作是C、CE、DEL，跳转情形1.

%：五种情况

1.如果小屏幕里没有四则运算符，则大屏幕与小屏幕都置0。（65%）（5平方%）

2.如果上一次操作运算符是+或者-，那么大屏幕显示的结果为上次大屏幕显示的结果\*当前大屏幕显示的结果/100，而小屏幕直接拼接此结果。（5+7负号-9%）

3.如果上一次操作运算符是\*或者/，那么大屏幕显示的结果为当前大屏幕结果/100，小屏幕直接拼接此结果。（5+7负号\*9%）

4.如果上一次操作是=或者使用=后未手动清除=即只使用单点运算或者=，按照是否有四则运算符以及四则运算符的种类相应跳转情形1，情形2，情形3.

(小屏幕迭代显示)（5=%）（8+4==%）（5\*6==%）

5.如果上一次操作是C、CE、DEL，根据具体情况跳转情形1，情形2，情形3。