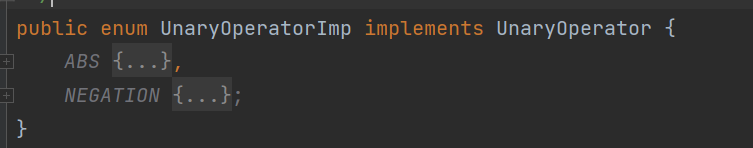
实习三

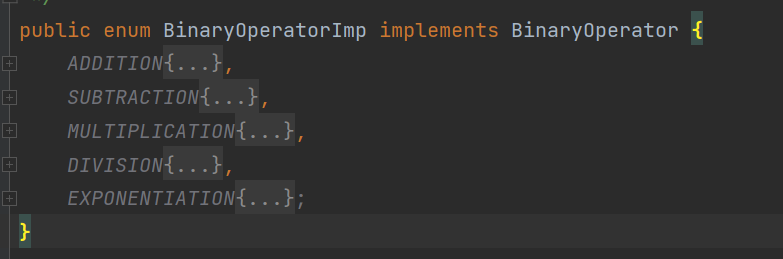
**第1部分:使用多态性实现计算器操作**

1. 实现单目运算符和二目运算符的实例类

单目运算符：绝对值、相反数



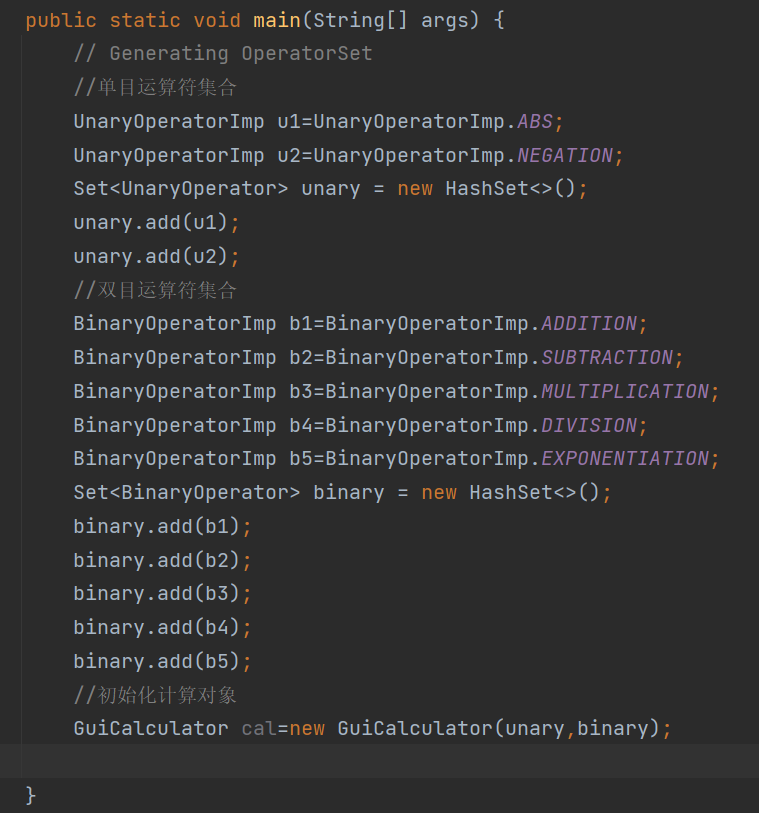
双目运算符：加、减、乘、除、指数函数



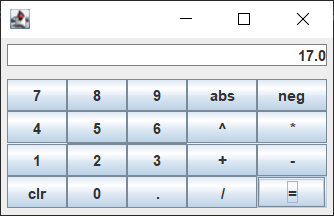
由于运算符实现很简单，所以就不详细的解释实现过程了；

1. 在src.lab3.guicalc.Main中完成该程序，即将运算符当作参数传递给src.lab3.guicalc.GuiCalculator的构造函数并启动计算器。

初始化程序：

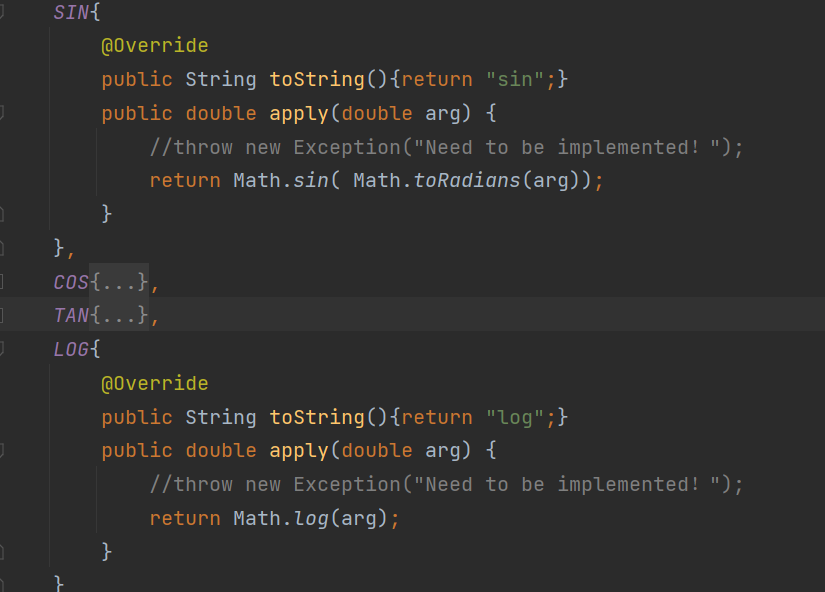


程序运行：

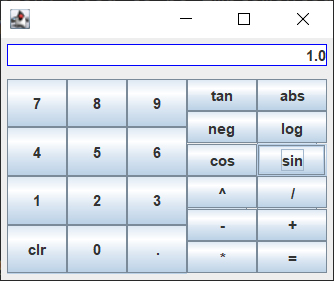


添加sin、cos、tan、log运算：

在这里自己由于对可视化不是很熟悉，所以踩坑了，去看了很久的CalculatorController类，然后后来发现直接在单目运算符枚举类里面加运算就可以，然后在初始化的时候加入set就行，然后分别实现sin和cos和tan的角度运算，log的运算；

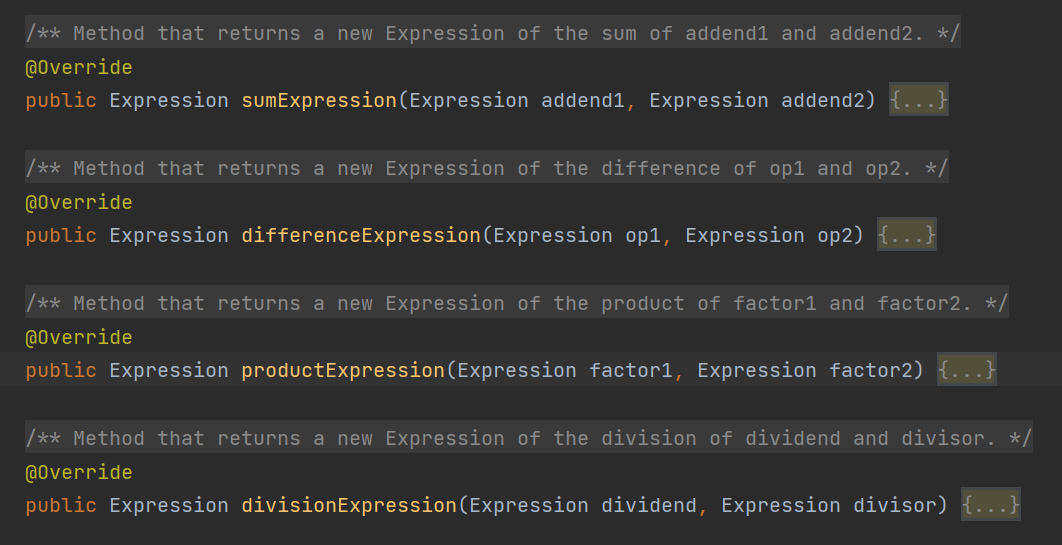


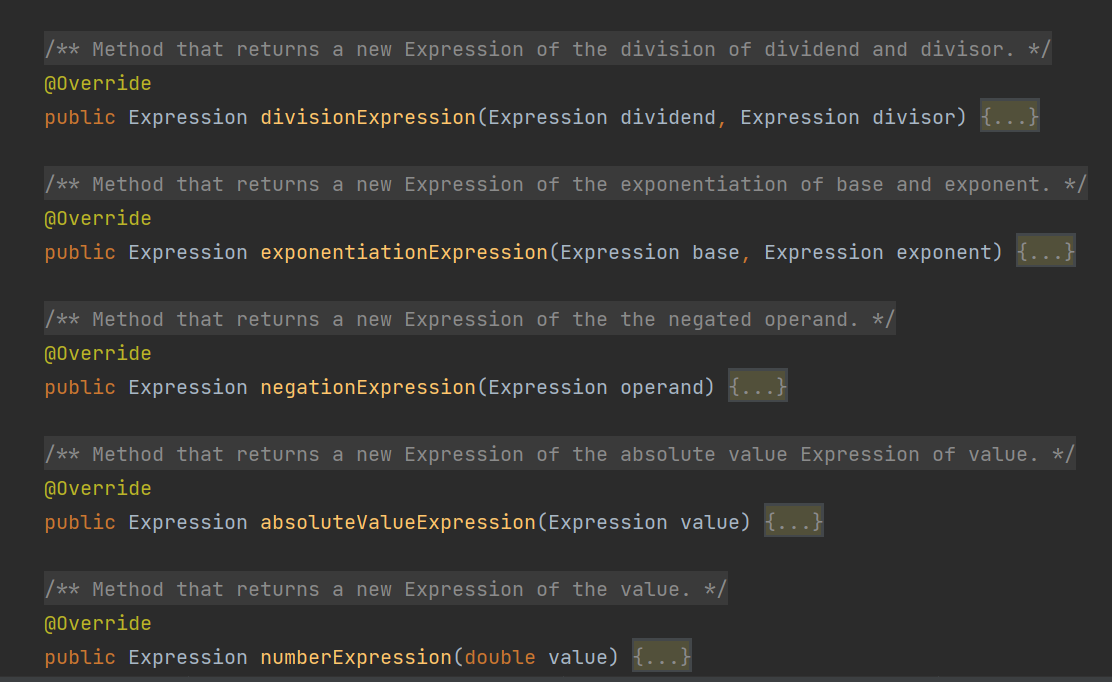
运行结果：



**第2部分:实现计算器表达式**

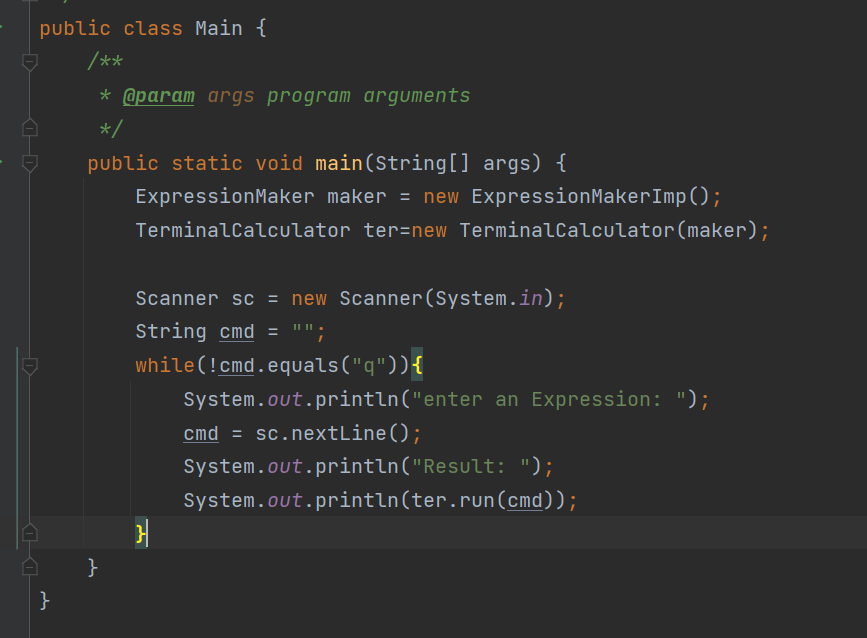
1. 实现表示数字、一元运算符表达式和二元运算符表达式的类（数字，2个一元运算符，5个二元运算符；共8个类）



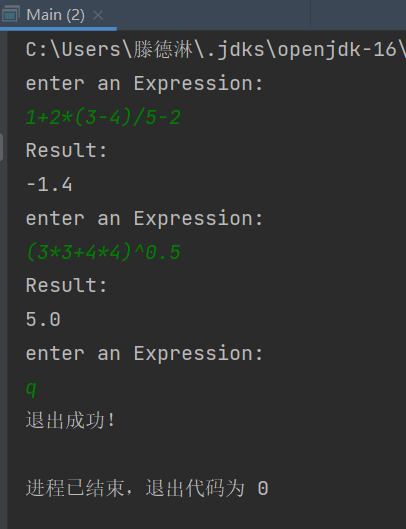


这些函数实现主要是基于Math库的实现，经过这个终端的计算器联系，我了解到接口的使用，还有接口类的实现；

1. 编写主函数：



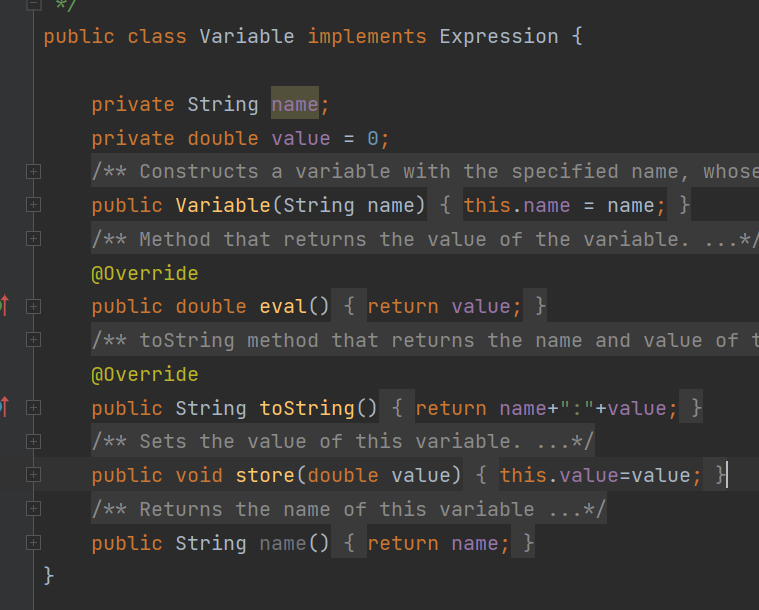
运行结果：



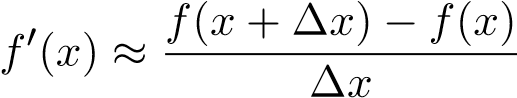
**第3部分:函数式编程:导数与牛顿法**

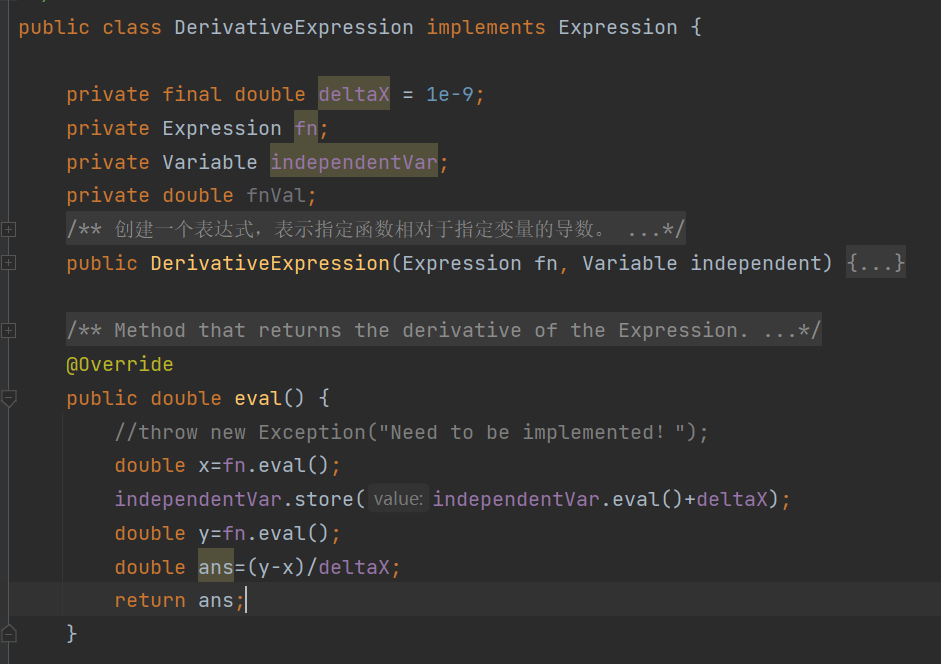
1. **导数**

这个部分需要先完成变量类src.lab3.expression.Variable的完善：这个部分很简单，几乎只是初始化一个变量而已。



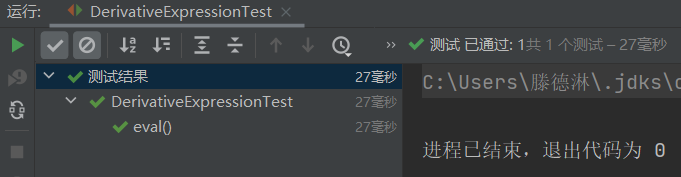
然后完成导数类src.lab3.expression.DerivativeExpression的补充：这里就是根据提供的公式进行求导数，然后返回导数的值，便于进行单元测试；





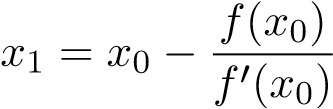
最后，编写测试案例：

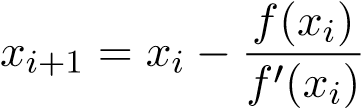




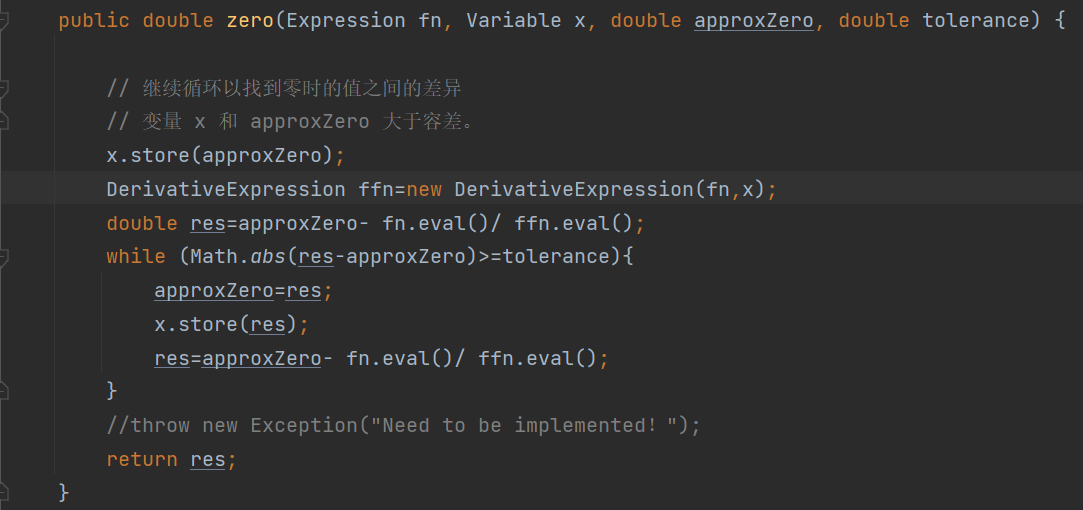
1. **牛顿法求积分**

与前面类似，编写src.lab3.expression.NewtonsMethod类中的函数，然后在测试用例中调用，进行求积分；主要是根据提供的公式进行计算值；





根据公式，很容易想到用一个循环，然后每次计算一次，最后得到零点。



编写测试用例：



