**高级计算机模拟**

**目 录**

**一 前言**

**1、程序设计问题描述**

**2、程序开发工具描述**

**二 需求分析**

**三 程序设计与实现**

**1、界面设计与实现**

**2、总体程序设计**

**3、计算程序设计**

**四 程序调试与运行**

**1、运行界面**

**2、运行结果**

**3、运行问题与不足**

**五 总结与分析**

**一 前言**

**1、程序设计问题描述**

本次所选课程设计主题：高级计算器模拟程序

功能要求：该程序显示 GUI 用户界面，能实现整数的加、减、乘、除、乘方、立方、n次方、开平方、开n次方、倒数、三角函数、反三角函数运算。

界面基本要求：用图形界面实现。

**2、程序开发工具描述**

本次课程设计涉及相关程序开发工具：Java语言；NetBeans IDE 8.2；JavaFX Scene Builder 2.0。

**二 需求分析**

为了呈现并实现计算器系统的各项功能，首先需要实现计算机界面的呈现，其次需要解决各项计算的问题，另外，还要实现主要实现各项计算程序与计算器界面连接，即程序要监听计算器按键，确保计算器能正常实现并显示计算式子与结果。

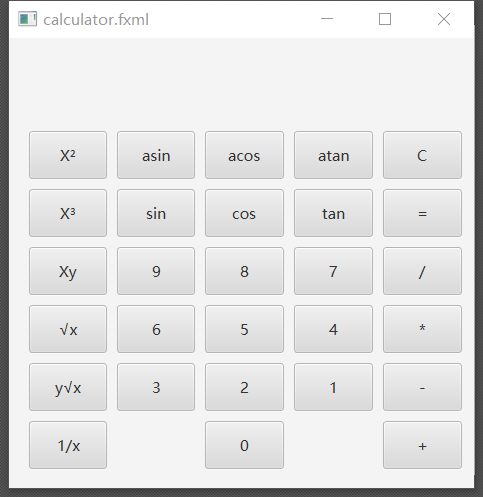
**三 程序设计与实现**

创建一个JavaFX fxml应用程序的项目，并将项目下的三个文件分别改名为Calculator.java ; CalculatorController.java ; Calculator.fxml。它们会在以下的论述中会出现。

**1、界面设计与实现**

计算器的界面设计是通过JavaFX实现的，计算器的功能需要能实现整数的加、减、乘、除、乘方、立方、n次方、开平方、开n次方、倒数、三角函数、反三角函数运算，再加上10个数字、一个等号、一个清零，总共需要28个按键，以及一个显示屏。在JavaFX Scene Builder 经过多次排版布局、调整大小，

双击Calculator.fxml可直接到达JavaFX Scene Builder的界面，通过谨慎地排版布局、多次调整大小以及更改按键显示文本，大概可以做出计算器界面，预览图如下：

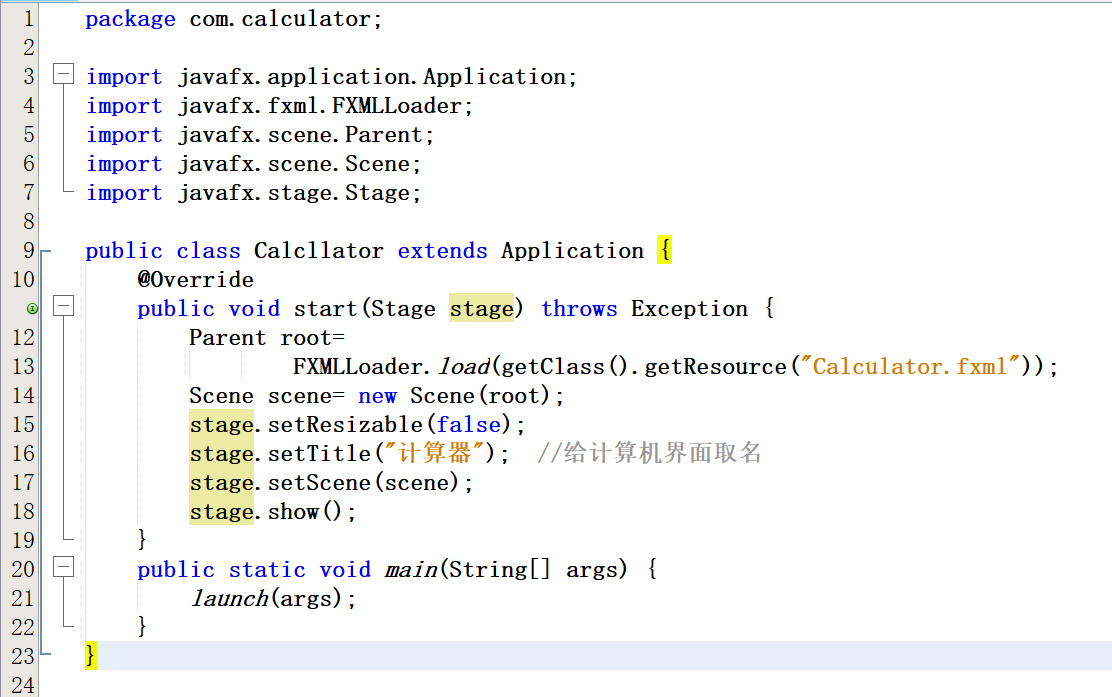


保存后，对应的Calculator.fxml文件发生改变，还可得到初步的CalculatorController.java文件。

**2、总体程序设计**

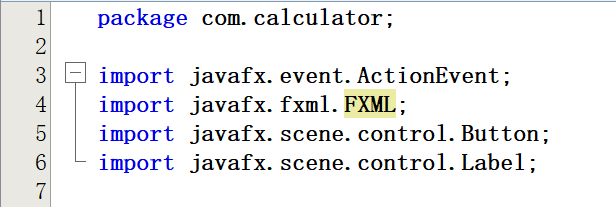
**·Calculator.java 文件**

在原有程序的基础上给计算机界面取一个名字：stage.setTitle("计算器")。可得到以下程序。

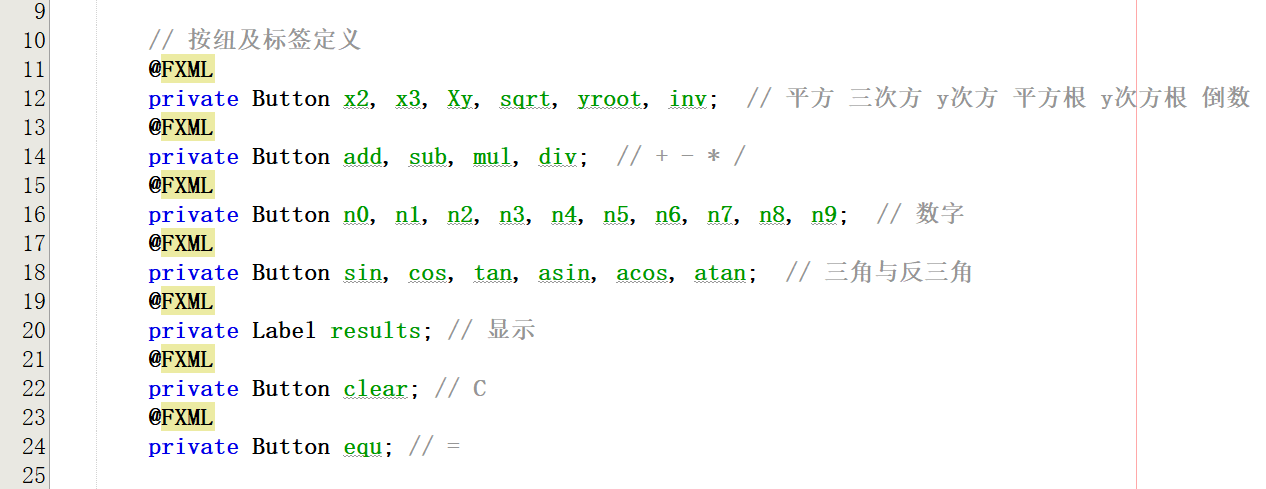
****

**·CalculatorController.java 文件**

**用到的工具包：**

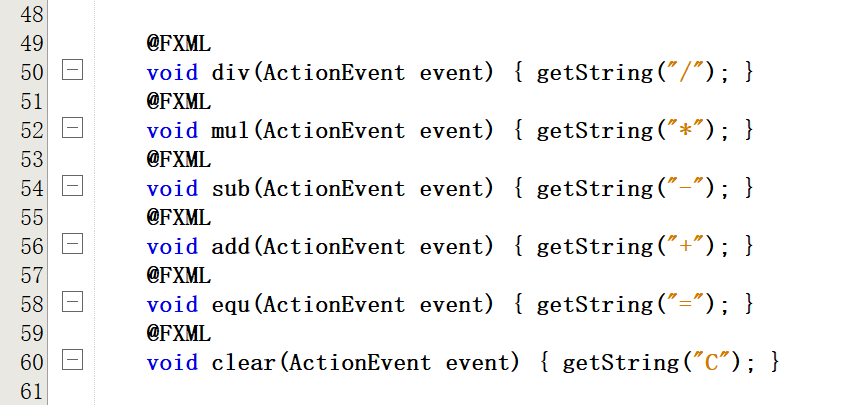


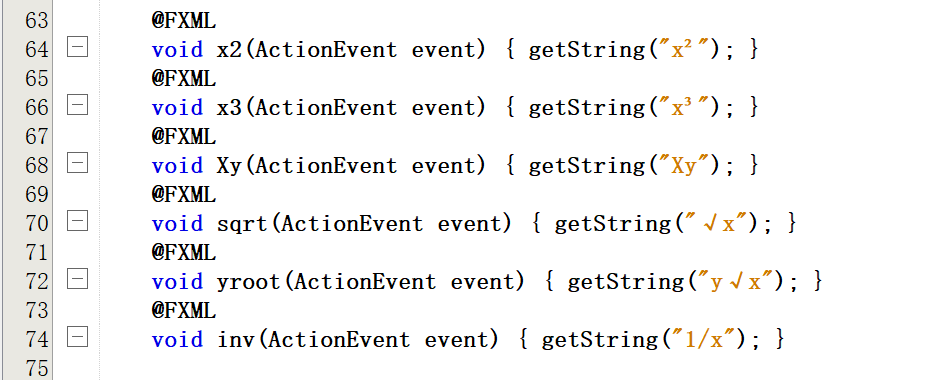
**定义按钮及标签：**

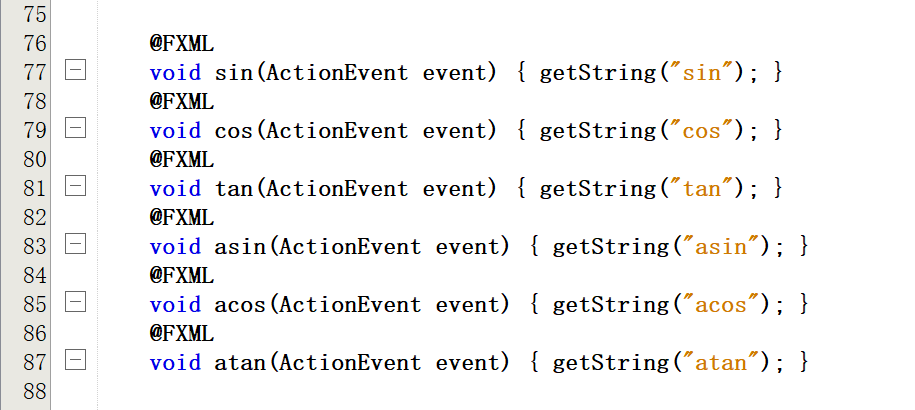


**给按钮添加事件，调用显示整体式子与结果的函数（即getString(String s)函数），各个按钮对应该内容的字符串：**





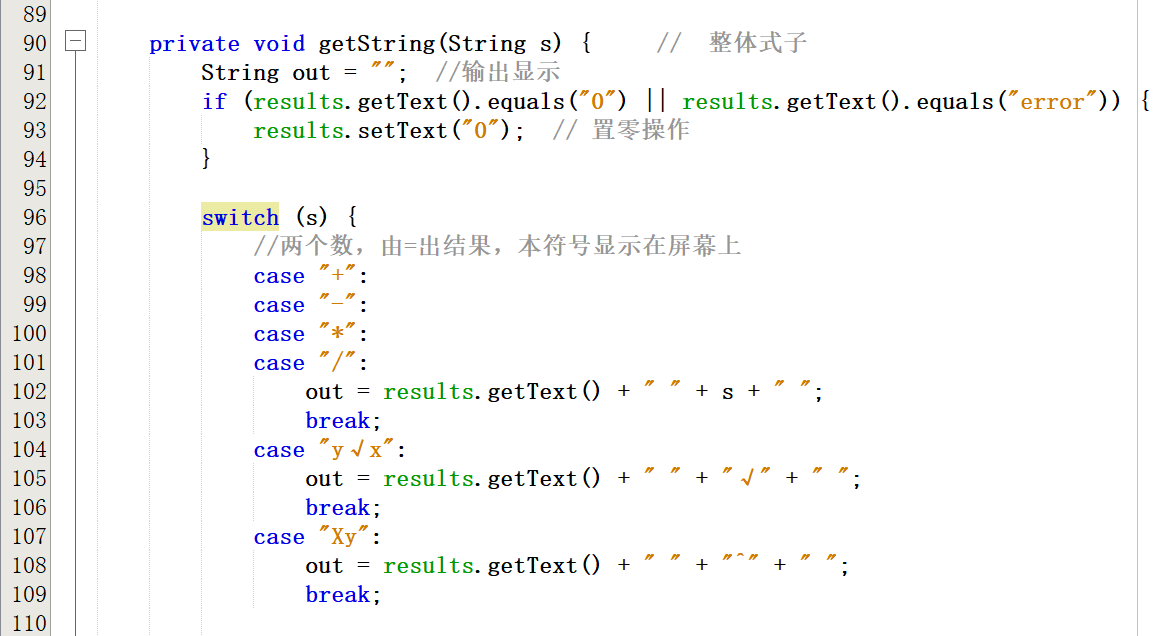




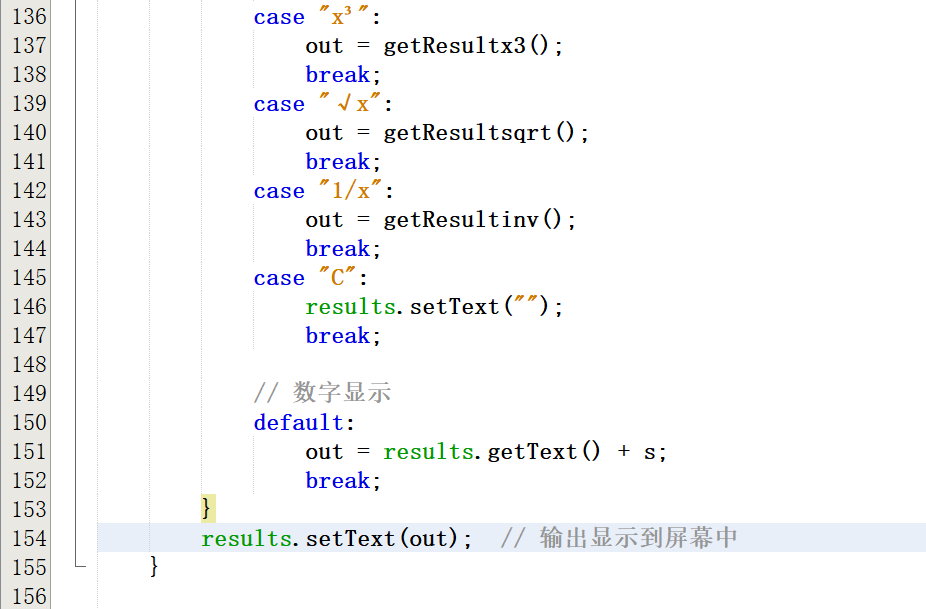
**式子与结果显示（即getString(String s)函数）：**

分析可知，“+”、“-”、“\*”、“/”、“y√x”、“Xy”这些按钮，对应符号计算涉及两个数，不可以由自身直接显示结果，需要后续用“=”显示，故它们需要在屏幕上显示；而“=”、“sin”、“cos”、“tan”、“asin”、“acos”、“atan”、“x²”、“x³”、“√x”、“1/x”以及“C”，这些按钮对应符号，不管是只涉及一个数还是两个数，亦或者置空（“C”），他们都是由该符号直接显示结果，无需在屏幕上显示该符号。当然，各自符号的运算结果需要设计各自的函数，以供调用。

代码设计如下：







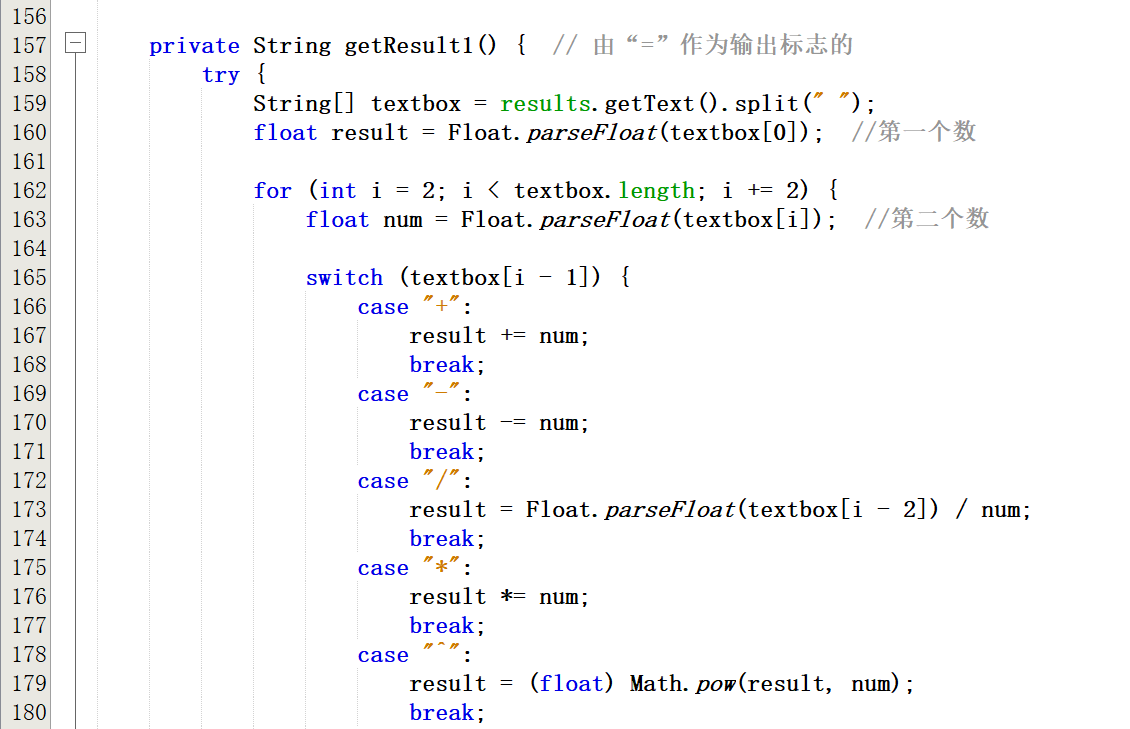
**3、计算程序设计**

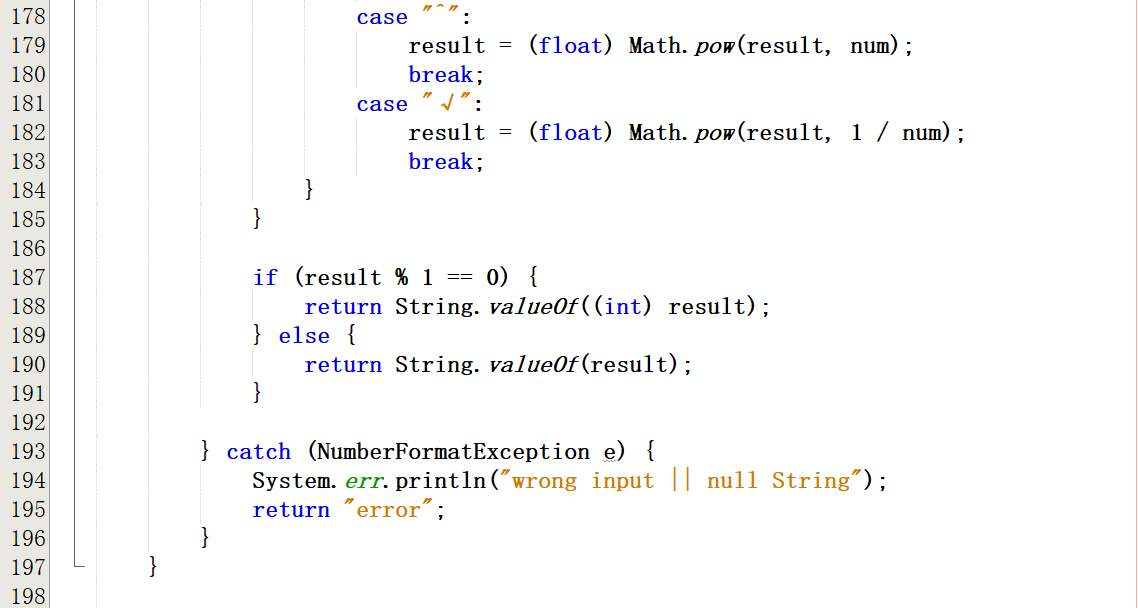
由于“+”、“-”、“\*”、“/”、“y√x”、“Xy”的按钮对应符号都是由“=”作为式子结束输出结果，所以他们的输出显示在一个函数中；其他符号要么输入或输出不具有一致性，要么有自己独有的条件（如分母不为0），所以单独构建函数。

部分复杂的数学函数可以用Math包。

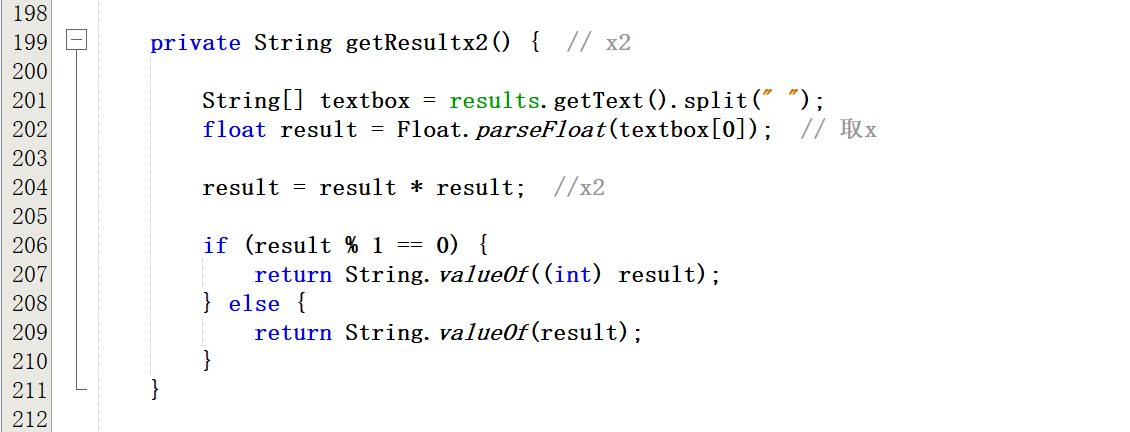
具体代码如下：

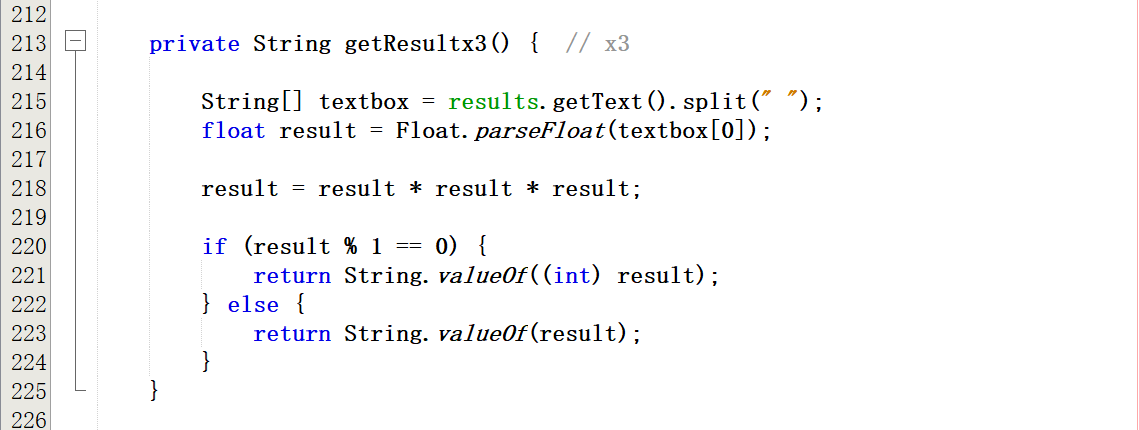
由“=”作为输出标志的：

****

****

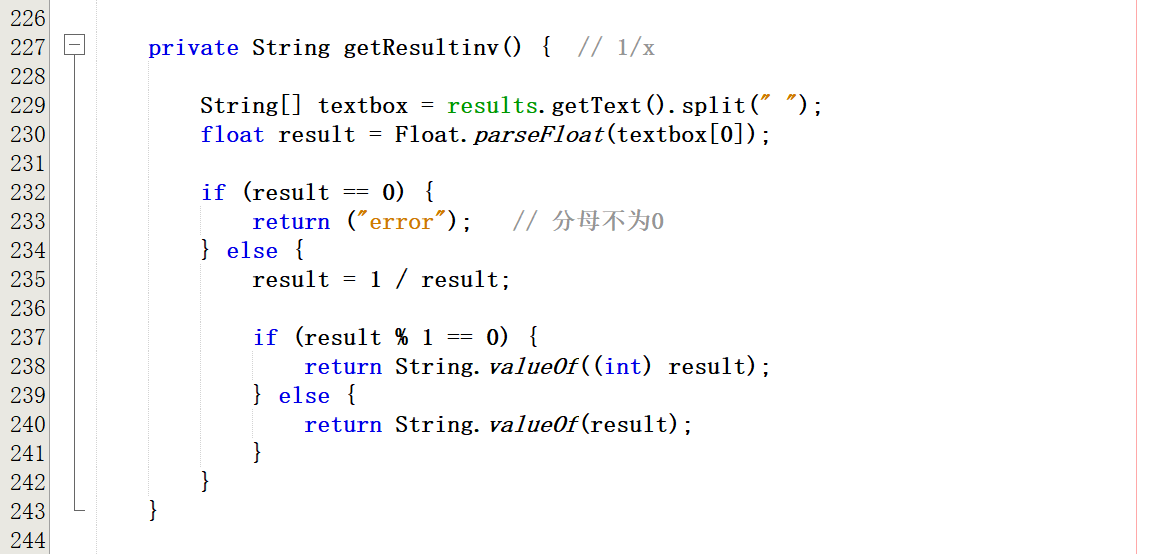
平方：



三次方：

倒数：

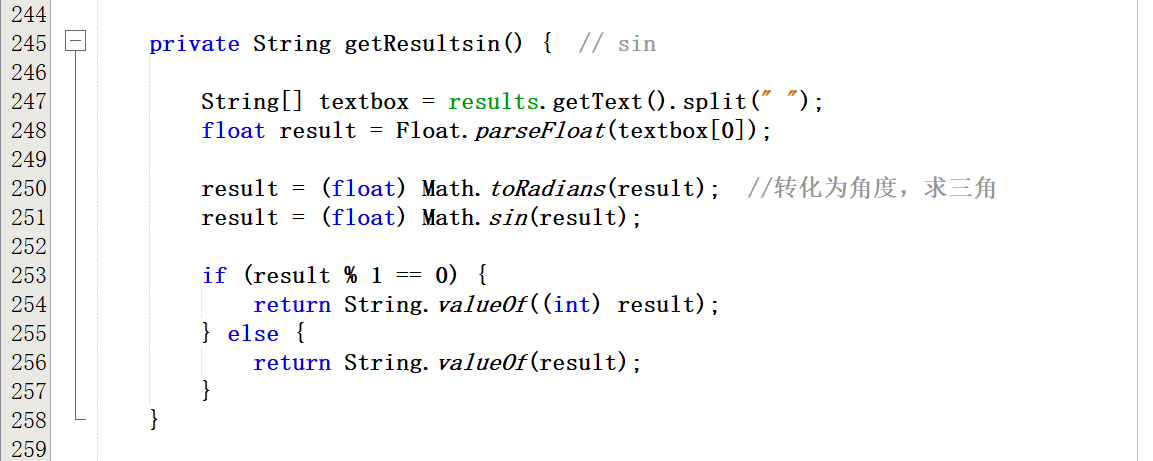
注意：分母不可为0

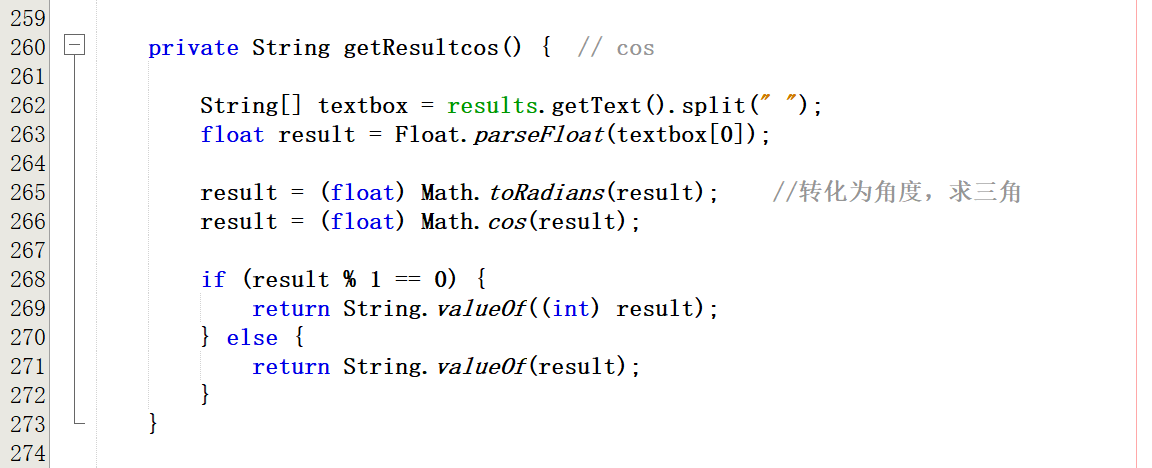


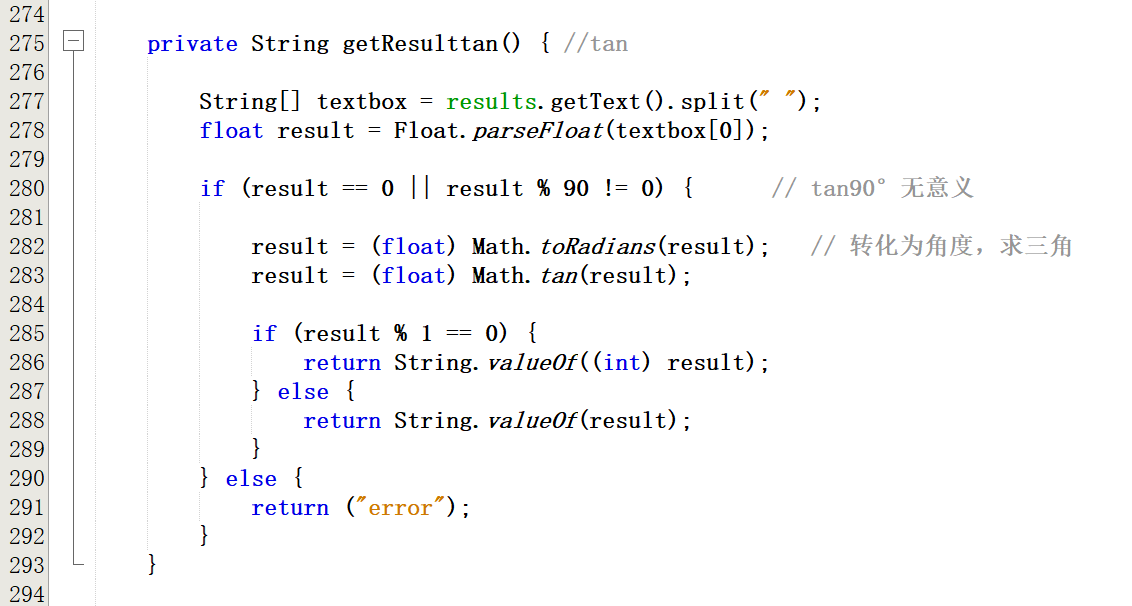
三角函数：

注意：1、result要转化为度数，才可调用三角函数

2、正切函数不可有90度及90度的倍数，会无意义



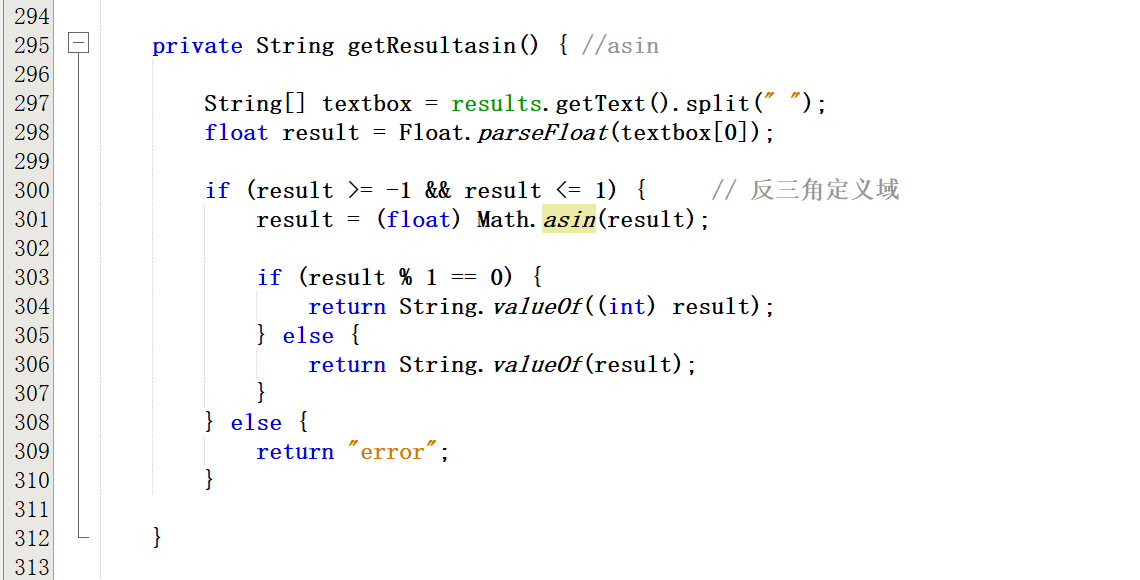


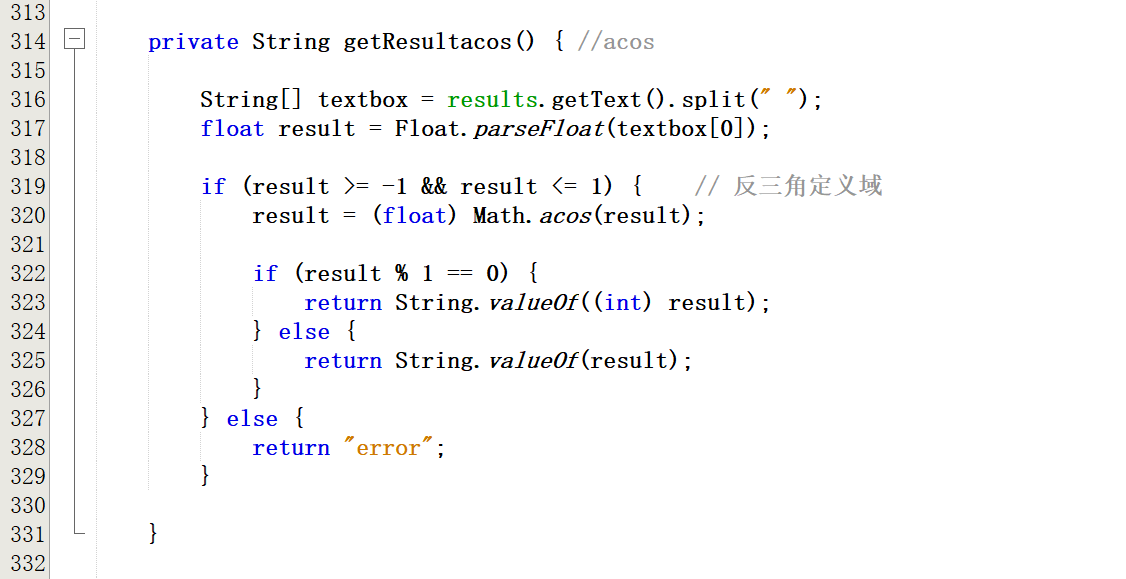


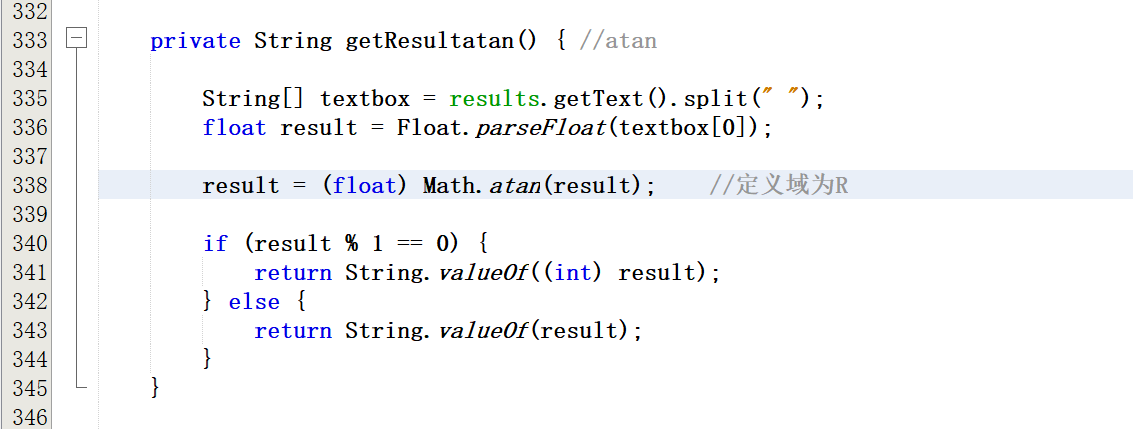
反三角：

注意：1、反三角函数无需转化为度数

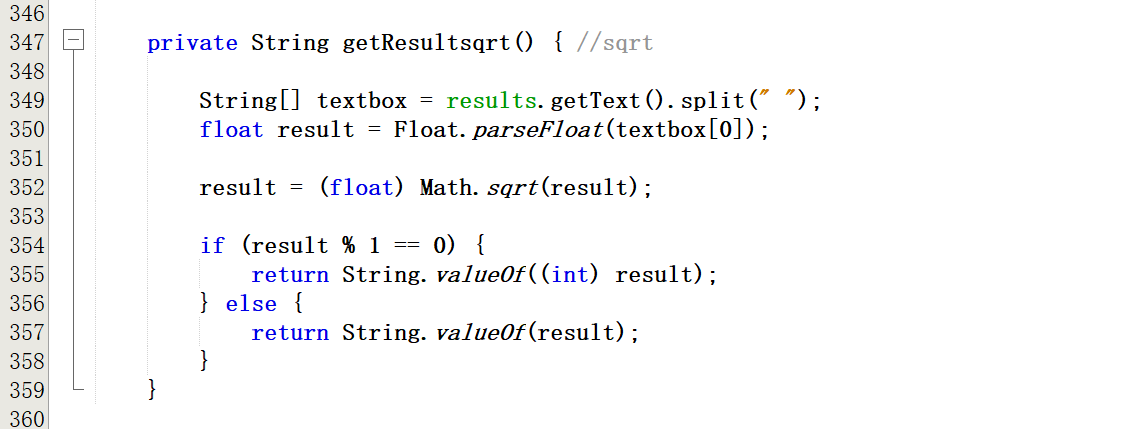
2、反三角函数的定义域







平方根：



**四 程序调试与运行**

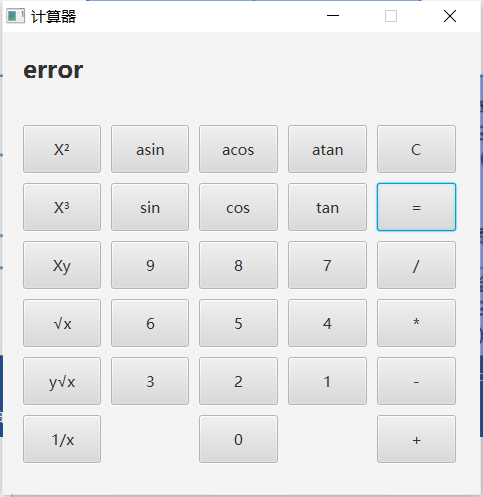
**1、运行界面：**



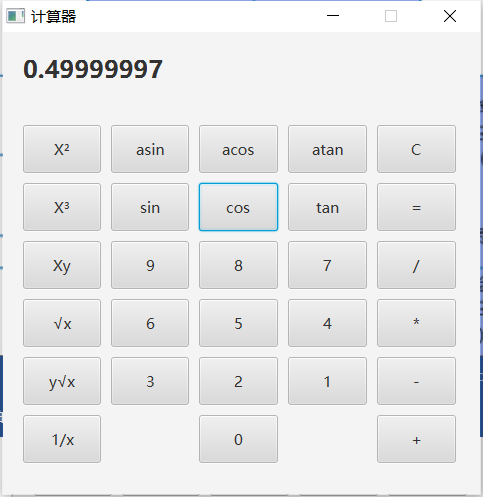
**2、运行结果：**大致能正常运行与显示结果。

**3、运行问题与不足：**

**·无法参与有负整数的运行**

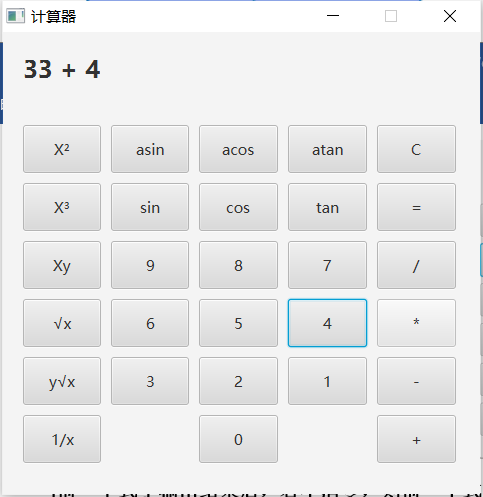
****

**·部分有精度损失的情况，如cos 60：**

****

**·当前一个式子输出结果后，若不清零，则前一个式子的结果存留，如先输入1+2，用等于输出结果后不清零，直接输下一个式子3+4：**

****

****

**五 总结与分析**

通过本次课程设计，我有了解了更多有关Java方面的知识。

从开始思考选题到到完成，前前后后也差不多有了将近十天。从刚开始的无从着手到最后能找到并解决问题，我感觉我和Java之间的距离更近了。

之前写的程序差不多都很基础，遇到瓶颈，更有同学一起讨论、商量，我似乎也习惯了依赖有难题找同学，而本次课程设计，十个选题，同学们各有不同，遇到瓶颈时，我学会了独自思考、查阅资料、网上寻找信息。写程序的道路真的是一步一步，又磕磕绊绊，每当问题出现时，又解决不出来时，心情真的是渐渐焦躁，但是解决出来了，下一个问题就会出现。这段日子，每当点击运行程序的小按钮时，真的是忐忑不安，真的怪不得都说程序员要保护好自己的发际线啊。

实践出真知，经过一次又一次的思考，一遍又一遍的检查，我在渐渐弥补前期我知识欠缺和经验不足，我掌握的知识不再是纸上谈兵，基础也更加扎实了。

虽然本次课程设计仍有不足，精度损失、负数以及上个式子残余的问题依然存在，但是在这段时间，我已经尽了我最大的努力。

本次设计也让我明白了思路即出路，有什么不懂不明白的地方要及时请教或上网查询，而且不要轻易放弃心中一闪而逝的小想法，尝试一下，说不定就解决了一个大难题。所以只要认真钻研，动脑思考，动手实践，就没有弄不懂的知识。

在未来的学习过程中，我会秉承上述观念，积极思考，认知钻研，在实践中得到真理。