**测试文档**

一、白盒测试

本次项目中的白盒测试由程序编写者进行，测试内容主要涉及两方面：程序代码测试和GUI测试。程序代码测试主要包括sin、cos、arcsin、arctan等四个函数，以及GUI代码等运行测试，GUI测试则是通过测试GUI的功能、计算时间、计算结果等。

1.1、单元测试

单元测试是指对软件中的最小可测试单元进行检查和验证，而这里的最小 单元，则要根据实际情况去判定其具体含义。比如在该项目中，所有程序用Python语言撰写，可以将sin、cos、arcsin、arctan等四个函数和GUI视作5个小单元进行测试。

四个函数的测试通过调用Python中的unittest实现，创建测试脚本，引用已经写好的四个三角函数，并建立测试类，该类继承了unittest.TestCase，这里的TestCase是unittest包的核心之一，通过实例化的方法测试函数。在上述类中定义四个测试函数test\_xx，用于分别测试四个函数，如test\_arcsin和test\_cos用于测试反正弦和余弦函数。在测试函数test\_xx中调用self.assertEqual(a, xx(b, c))语句，其中a是预想的测试结果，xx()则是待测试函数，b，c是传入xx()的参数，要注意测试函数必须将test作为前缀，否则无法被调用。

创建测试实例，运行测试代码，若预想结果a与xx(b,c)计算结果一致，则测试通过，程序不会报错，若不一致，程序将出现assert异常，由此判断函数代码的逻辑和功能是否正确。

1.2、子系统测试

运行GUI代码后能出现相应界面，则表示初步测试正确。进一步测试其上正负、删除、归零、小数点、数字和三角函数等功能，通过鼠标点击GUI界面上相应的按钮，观察到正确结果，则说明测试正确。使用GUI上的函数进行计算，若能在极短时间内出现正确结果，表明测试成功。

二、黑盒测试

黑盒测试由用户（非程序编写者）进行，主要从用户需求和体验出发，进行功能测试和稳定性测试，前者包括常见值测试，准确度测试，边界值测试。

2.1 功能测试

2.1.1 常见值测试

该项目的主要实现三角函数和反三角函数的计算，因此常见值为常见的角度值及其对应的函数值。常见的角度有0，±30，±45，±60，±90，±120，±180，±360等（此处注意负号要在输入数字后才能设置），在GUI界面输入上述角度后测试sin和cos函数，观察验证计算结果。对于反三角函数，则需要输入常见的三角函数值，如0，±1，±0.70711，±0.86602，±1.732，±999999(无穷)等，然后计算对应的arcsin和arctan值，观察验证计算结果，完成输入正负号、小数点、数字及相应函数的测试。

输入数字后几点C或del按钮，可以完成清零和删除操作，或者计算出相应函数值后点击C也能清零，完成测试。

2.1.2 准确度测试

准确度测试主要涉及函数计算结果，包括角度和数值计算的准确度。如角度为60的sin和cos计算值分别为0.866和0.5，1.7325输入arctan计算的角度为60.01，0.8的arcsin值为53.1，可见结果保留一到三位小数。

2.1.3 边界值测试

三角函数sin和cos，其输入边界值没有限制，即正负无穷之间任何数都可以，但输出值限制在±1之间，输入0，±90，±180及其附近的角度值进行测试，如±93，±180.2等计算结果都没有超出边界，测试成功。反三角函数中arctan的与sin和cos情况类似，输入无限制，输出限制在±1之间，而arcsin情况最为特殊，输入限制在±1之间，输出范围为±90之内，当输入的值超过1或小于-1时，会出现“无效输入”的提示。

2.2 稳定性测试

稳定性测试主要分为多次测试和连续计算两种方法。

多次测试即多次使用计算器功能，进行输入数字、加负号、加小数点、清零、删除、使用函数等操作，当重复以上操作成百上千次后依然能得到正确结果，则表明通过稳定性测试。

连续计算测试顾名思义，输入角度或数值计算得到对应的数值或角度后，再基于得到的结果进行角度或数值的计算，简言之就是角度或数值的持续计算，如输入30计算sin值为0.5，再计算arctan值为26.57，再计算sin值为0.447照此持续进行下去，能够得到正确结果则表明测试成功。