# 测试文档

## 1白盒测试

白盒测试通过检查软件内部的逻辑结构，对软件中的逻辑路径进行覆盖测试。在程序不同地方设立检查点，检查程序的状态，以确定实际运行状态与预期状态是否一致。本次项目的白盒测试主要涉及两个方面的内容，一是测试程序代码，二是进行GUI测试。测试程序代码是测试sin、cos、arcsin、arctan这四个三角函数函数计算的结果是否正确，以及GUI代码的运行测试。GUI测试则是通过测试检验GUI的功能、计算时间、计算结果是否合理。

### 1.1单元测试

单元是人为规定的最小的被测功能模块，单元测试（模块测试）是开发者编写的一小段代码，用于检验被测代码的一个很小的、很明确的功能是否正确。在此项目中，所有程序均用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。

对于四个函三角数的测试，将通过调用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\_sin和test\_cos用于测试正弦和余弦函数。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。

对于本项目选取部分数据进行测试，选取的数值如表1所示，测试结果如图1所示，由图可见代码通过了测试。

表1 部分测试数据

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| sin | | cos | | arcsin | | arctan | |
| 角度 | 对应数值 | 角度 | 对应数值 | 数值 | 对应角度 | 数值 | 对应角度 |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 30 | 0.5 | 30 | 0.866 | 0.1 | 5.7 | 0.577 | 30 |
| 45 | 0.707 | 45 | 0.707 | 0.2 | 11.5 | 1 | 45 |
| 60 | 0．866 | 60 | 0.5 | 0.3 | 17.5 | 1.732 | 60 |
| 90 | 1 | 90 | 0 | 0.4 | 23.6 | 10 | 84.29 |
| 120 | 0.866 | 120 | -0.5 | 0.5 | 30 | 100 | 89.43 |
| 145 | 0.574 | 145 | -0.819 | 0.6 | 36.9 | 0 | 0 |
| 180 | 0 | 180 | -1 | 0.707 | 45 | -0.577 | -30 |
| 210 | -0.5 | 210 | -0.866 | 0.886 | 60 | -1 | -45 |
| 270 | -1 | 270 | 0 | 0.9 | 64.2 | -1.732 | -60 |
| 360 | 0 | 360 | 1 | 1 | 90 | -10 | -84.29 |
| -30 | -0.5 | -30 | 0.866 | -0.1 | -5.7 | -100 | -89.43 |
| -45 | -0.707 | -45 | 0.707 | -0.2 | -11.5 |  |  |
| -60 | -0.866 | -60 | 0.5 | -0.3 | -17.5 |  |  |
| -90 | -1 | -90 | 0 | -0.4 | -23.6 |  |  |
| -120 | -0.866 | -120 | -0.5 | -0.5 | -30 |  |  |
| -145 | -0.574 | -145 | -0.819 | -0.6 | -36.9 |  |  |
| -180 | 0 | -180 | -1 | -0.707 | -45 |  |  |
| -270 | 1 | -270 | 0 | -0.886 | -60 |  |  |
| -360 | 0 | -360 | 1 | -0.9 | -64.2 |  |  |
|  |  |  |  | -1 | -90 |  |  |



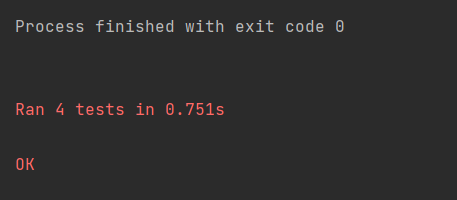


图1 四个三角函数部分数值测试结果

### 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。

由图2可知运行GUI代码后出现的界面正确。由图3可知该代码可以实现正负号的转换、输入小数点及输入数字的功能，并能实现三角函数计算功能。

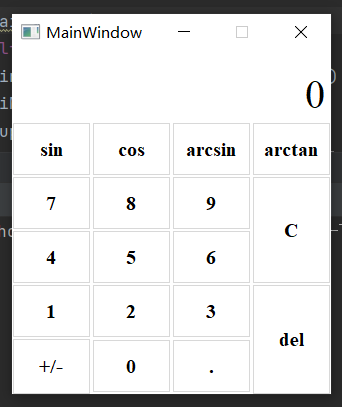


图2 GUI代码界面测试结果

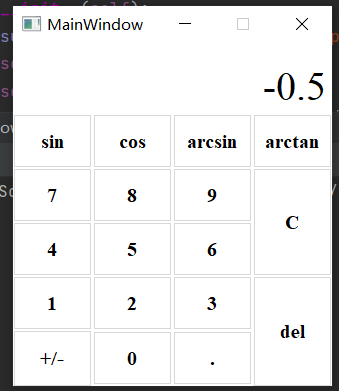
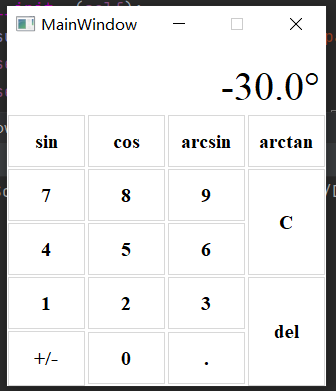
 

图3 程序的正负号的转换、输入小数点、输入数字及计算功能测试结果

## 2黑盒测试

黑盒测试，它是通过测试来检测每个功能是否都能正常使用。在测试中，把程序看作一个不能打开的黑盒子，在完全不考虑程序内部结构和内部特性的情况下，在程序接口进行测试，它只检查程序功能是否按照需求规格说明书的规定正常使用，程序是否能适当地接收输入数据而产生正确的输出信息。黑盒测试着眼于程序外部结构，不考虑内部逻辑结构，主要针对软件界面和软件功能进行测试。

在黑盒测试部分对此项目进行边界值测试，由于三角函数sin、cos、arctan的输入边界值没有限制，即正负无穷之间任何数都可以，所以输入数值较大的数进行测试，测试结果如图4。三角函数sin、cos输出值限制在±1之间，所以随机选取数值进行测试观察其结果如图5所示。三角函数arctan输出值没有限制，所以随机选取数值进行测试观察其结果如图6所示。arcsin函数的取值范围与前面三个函数不同，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，测试结果如图7。从下面得到的测试结果可以知道边界值测试成功。

表格, 日历

描述已自动生成 表格, 日历

描述已自动生成

图4 sin、cos、arctan输入值测试结果

表格, 日历

描述已自动生成 表格, 日历

描述已自动生成

图5 sin、cos输出值测试结果

表格, 日历

描述已自动生成 表格, 日历

描述已自动生成

图6 arctan输出值测试结果

表格, 日历

描述已自动生成 日历

描述已自动生成

图7 arcsin输入值测试结果