

白盒测试实验说明

找一个经典算法题目及其 Java 代码（函数代码 50~100 行且不少于 6 个判断语句，不少于 2 个循环），作为被测函数。

1、安装配置 JUnit，编写测试驱动代码，利用 JUnit 对被测函数进行单元测试。关于 Junit 的教程，可以参考网址

<http://www.cnblogs.com/ysocean/p/6889906.html>，或网址

<https://www.cnblogs.com/cheetah666/p/6530202.html>，或附件中文件“JUnit 入门教程 - v1.0.pdf”，或观看视频“0 2.6 JUNIT 使用示例.mp4”。

2、在被测函数的分支中增加 assert 语句，用于统计覆盖率，验证执行 JUnit 的结果。

3、对于其中的模块、函数单元，分别依据**语句覆盖准则**和**分支覆盖准则**，设计测试用例。利用 JUnit 批量执行这些测试用例，记录测试时间、用例、覆盖率，验证所设计的测试用例是否输出了所要求的覆盖准则的完全覆盖、是否发现缺陷。

要求提交实验报告，内容包括：

- 1) 算法题目和 Java 代码
- 2) 参照教材表 4.1 设计覆盖测试用例；
- 3) 记录测试时间、测试用例、覆盖率、是否发现缺陷。

4、**(选做题)** 对于其中的模块、函数单元，依据**原子谓词覆盖准则**，设计测试用例。利用 JUnit 批量执行这些测试用例，记录测试时间、用例、覆盖率，验证所设计的测试用例是否输出了所要求的覆盖准则的完全覆盖、是否发现缺陷。

要求提交实验报告，内容包括：

- 1) 参照教材表 4.2 设计覆盖测试用例；
- 2) 记录测试时间、测试用例、覆盖率、是否发现缺陷。

JUnit 举例：

被测 Java 代码：

```
class testObject{  
    void func(arg1, arg2...){ //实现某种功能的被测函数
```

```

if(C){
    assertTrue(boolean expected, boolean actual);
    ...
    }//利用在分支中插装语句得到分支覆盖情况
else if (A&&B||C) {//思考一下，如何获得原子谓词的执行情况？
    assertTrue(boolean expected, boolean actual);
    ...
    }
}
}

```

JUnit 代码：

```

class JUnitTestClass{
    @Test
    void test4BlockCoverage(){//用于执行语句覆盖的测试用例集
        a = new testObject ();
        a.func(testcase1);//testcase1,2 是语句覆盖准则的测试用例
        a.func(testcase2);
        ...
    }
    @Test
    void test4BranchCoverage(){//用于执行分支覆盖准则的测试用例集
        test4BlockCoverage();//可利用上一个执行语句覆盖的自动测试的方法
        a = new testObject ();
        a.func(testcase6);//testcase6,7 是分支覆盖的测试用例
        a.func(testcase7);
        ...
    }
}

```