



链滴

JUnit3.8.2设计模式浅谈之装饰模式

作者: [changming](#)

原文链接: <https://ld246.com/article/1471266463500>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)


```

* A Decorator that runs a test repeatedly.
*
*/
public class RepeatedTest extends TestDecorator {
    private int fTimesRepeat;

    //通过构造方法将test目标类传递进去
    public RepeatedTest(Test test, int repeat) {
        super(test);
        if (repeat < 0)
            throw new IllegalArgumentException("Repetition count must be > 0");
        fTimesRepeat= repeat;
    }
    public int countTestCases() {
        return super.countTestCases()*fTimesRepeat;
    }
    //在实现接口方法时，添加附加逻辑，这里是重复执行测试

    public void run(TestResult result) {
        for (int i= 0; i < fTimesRepeat; i++) {
            if (result.shouldStop())
                break;
            super.run(result);
        }
    }
    public String toString() {
        return super.toString()+"(repeated)";
    }
}

```

针对不同的功能扩展，只要继承同一个装饰器即可。

JUnit测试框架的7种设计模式全部聊完，这几种模式结合的非常紧凑，框架内部往往个类担任几种不同角色，显得小而美。