

使用MOCK对象进行单元测试的实例讲解 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/252/2021_2022__E4_BD_BF_E7_94_A8MOCK_c104_252527.htm

1.出了什么问题？单元测试的目标是一次只验证一个方法，小步的前进，细粒度的测试，但是假如某个方法依赖于其他一些难以操控的东东，比如说网络连接，数据库连接，或者是Servlet容器，那么我们该怎么办呢？要是你的测试依赖于系统的其他部分，甚至是系统的多个其他部分呢？在这种情况下，倘若不小心，你最终可能会发现自己几乎初始化了系统的每个组件，而这只是为了给一个测试创造足够的运行环境让它们可以运行起来。忙乎了大半天，看上去我们好像有点违背了测试的初衷了。这样不仅仅消耗时间，还给测试过程引入了大量的耦合因素，比如说，可能有人兴致冲冲地改变了一个接口或者数据库的一张表，突然，你那卑微的单元测试的神秘的挂掉了。在这种情况下发生几次之后，即使是最有耐心的开发者也会泄气，甚至最终放弃所有的测试，那样的话后果就不能想像了。再让我们看一个更加具体的情况：在实际的面向对象软件设计中，我们经常会碰到这样的情况，我们在对现实对象进行构建之后，对象之间是通过一系列的接口来实现。这在面向对象设计里是最自然不过的事情了，但是随着软件测试需求的发展，这会产生一些小问题。举个例子，用户A现在拿到一个用户B提供的接口，他根据这个接口实现了自己的需求，但是用户A编译自己的代码后，想简单模拟测试一下，怎么办呢？这点也是很现实的一个问题。我们是否可以针对这个接口来简单实现一个代理类，来测试模拟，期望代码生成自己

的结果呢？幸运的是，有一种测试模式可以帮助我们：mock对象。Mock对象也就是真实对象在调试期的替代品。

2.现在需要Mock对象吗？关于什么时候需要Mock对象，Tim Mackinnon给我们了一些建议：

- 真实对象具有不可确定的行为(产生不可预测的结果，如股票的行情)
- 真实对象很难被创建(比如具体的web容器)
- 真实对象的某些行为很难触发(比如网络错误)
- 真实情况令程序的运行速度很慢
- 真实对象有用户界面
- 测试需要询问真实对象它是如何被调用的(比如测试可能需要验证某个回调函数是否被调用了)
- 真实对象实际上并不存在(当需要和其他开发小组，或者新的硬件系统打交道的时候，这是一个普遍的问题)

3.如何实现Mock对象？使用mock对象进行测试的时候，我们总共需要3个步骤，分别是：

- 使用一个接口来描述这个对象
- 为产品代码实现这个接口
- 以测试为目的，在mock对象中实现这个接口

在此我们又一次看到了针对接口编程的重要性了，因为被测试的代码只会通过接口来引用对象，所以它完全可以不知道它引用的究竟是真实的对象还是mock对象，下面看一个实际的例子：一个闹钟根据时间来进行提醒服务，如果过了下午5点钟就播放音频文件提醒大家下班了，如果我们要利用真实的对象来测试的话就只能苦苦等到下午五点，然后把耳朵放在音箱旁...我们可不想这么笨，我们应该利用mock对象来进行测试，这样我们就可以模拟控制时间了，而不用苦苦等待时钟转到下午5点钟了。下面是代码：

```
public interface Environmental { private boolean playedWav = false. public long getTime(). public void playWavFile(String fileName). public boolean wavWasPlayed(). public void resetWav().
```

} 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com