

25 | 不破不立：掌握代码级测试的基本理念与方法

2018-08-24 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 12:01 大小 5.52M



你好，我是茹炳晟，今天我和你分享的主题是“不破不立：掌握代码级测试的基本理念与方法”。

我在第三篇文章[《什么是单元测试？如何做好单元测试？》](#)中，为你介绍了单元测试的基本概念和方法，和你聊到了单元测试用例的“输入数据”和“预计输出”，也谈到了驱动代码和桩代码，其实这些概念和方法在代码级测试中也是最基本的。

通常情况下，代码级测试的工作都是由开发人员完成，但是测试框架选型、覆盖率统计工具选型、测试用例设计原则等都需要资深的测试工程师或者测试架构师参与。

所以，代码级测试这个系列，我会和你分享测试人员应该具备的代码级测试基础知识，为你呈现一幅包括代码级测试技术入门、方法论、用例设计，以及覆盖率衡量、典型难点、解决

思路的全景技术视图。

为了更好地协助开发人员做好代码级测试，所以我今天的这次分享是根据实际工程项目中的实践，总结了五种常见的代码错误，以及对应的四大类代码级测试方法。

掌握了这些错误类型、测试方法，相信你就可以搞定代码级测试了，即使自己不用去完成测试工作，也可以让开发人员对你另眼相看，可以更高效地互相配合完成整个项目。

这里需要注意的是，**代码级测试的测试方法一定是一套测试方法的集合，而不是一个测试方法**。因为单靠一种测试方法不可能发现所有潜在的错误，一定是一种方法解决一部分或者一类问题，然后综合运用多种方法解决全部问题。

本着先发现问题，然后解决问题的思路，我在正式介绍代码级测试方法之前，先来概括一下常见的代码错误类型，然后我们再一起讨论代码级测试有哪些方法。这样，我们就可以清晰地看出，每一种代码级测试方法都能覆盖哪些类型的代码错误。

根据过往的经验来看，代码错误，可以分为“有特征”的错误和“无特征”的错误两大类。“有特征”的错误，可进一步分为语法特征错误、边界行为错误和经验特征错误；“无特征”的错误，主要包括算法错误和部分算法错误。


接下来，我将和你详细说说这五类代码错误的具体含义是什么。

常见代码错误类型

第一，语法特征错误

语法特征错误是指，从编程语法上就能发现的错误。比如，不符合编程语言语法的语句等。

如果你使用 IDE 环境进行代码开发，那么 IDE 可以提示你大部分的这类错误，而且只有解决了这类错误，才能编译通过。但是，还会有一些比较隐晦的语法特征错误，IDE 不能及时发现，而且也不会影响编译，只会在运行阶段出错。

 复制代码

```
1 void demoMethod(void)
2 {
3     int a[10];
4     a[10]=88;
```


```
5    ...  
6 }
```

比如，这段 C 语言代码就存在数据越界的问题。

很显然，你从语法上很容易就能发现，这段代码初始化了一个长度为 10 的整型数组 a，但数组下标从 0 开始，所以最大可用的数组空间应该是 a[9]，而这里却使用了 a[10]，造成数组越界，访问了未被初始化的内存空间，代码运行时（Runtime）就会造成意想不到的结果。

第二，边界行为特征错误

边界行为特征错误是指，代码在执行过程中发生异常，崩溃或者超时。之所以称为“边界”，是由于此类错误通常都是发生在一些边界条件上。


 复制代码

```
1 int Division(int a, int b)  
2 {  
3     return a/b;  
4 }
```

这段 C 语言代码就存在具有边界行为特征的错误。当 b 取值为 0 时，Division 函数就会抛出运行时异常。

第三，经验特征错误

经验特征错误是指，根据过往经验发现代码错误。

 复制代码

```
1 void someMethod(void)  
2 {  
3     ...  
4     if(i=2)  
5     {  
6         // if the value of i equals to 2, call method "operationA"  
7         operationA();  
8     }
```

```
8     }
9     else
10    {
11        // if the value of i doesn't equal to 2, call method "operationB"
12        operationB();
13    }
14 }
```

这段 C 语言代码，就是一个典型的具有经验特征错误的代码片段。代码想要表达的意思是：如果变量 `i` 的值等于 2，就调用函数 `operationA`；否则，调用函数 `operationB`。

但是，代码中将 “`if(i==2)`” 错误地写成了 “`if(i=2)`”，就会使原本的逻辑判断操作变成了变量赋值操作，而且这个赋值操作的返回结果永远是 `true`，即这段代码永远只会调用 `operationA` 的分支。

显然，“`if(i=2)`” 在语法上没有错误，但是从过往经验来看，这就很可能是个错误了。也就是说，当你发现一个原本应该出现逻辑判断语句的地方，现在却出现了赋值语句，那就很有可能是代码写错了。


第四，算法错误

算法错误是指，代码完成的计算（或者功能）和之前预先设计的计算结果（或者功能）不一致。

这类错误直接关系到代码需要实现的业务逻辑，在整个代码级测试中所占比重最大，也是最重要的。但是，完全的算法错误并不常见，因为不能准确完成基本功能需求的代码，是一定不会被递交的。所以，在实际工程项目中，最常见的是部分算法错误。

第五，部分算法错误

部分算法错误是指，在一些特定的条件或者输入情况下，算法不能准确完成业务要求实现的功能。这类错误，是整个代码级测试过程中最常见的类型。

 复制代码

```
1 int add(int a, int b)
2 {
3     return a+b;
```

这段 C 语言代码，完成了两个 int 类型整数的加法运算。在大多数情况下，这段代码的功能逻辑都是正确的，能够准确地返回两个整数的加法之和。但是，在某些情况下，可能存在两个很大的整数相加后和越界的情况，也就是说两个很大的 int 数相加的结果超过了 int 的范围。这就是典型的部分算法错误。

代码级测试常用方法

介绍完了语法特征错误、边界行为特征错误、经验特征错误、算法错误、部分算法错误这五类代码错误后，我们再回过头来看看代码级测试的方法有哪些，这些测试方法又是如何揭露这五类代码错误的。

在我看来，代码级测试方法主要分为两大类，分别是静态方法和动态方法。

静态方法，顾名思义就是在不实际执行代码的基础上发现代码缺陷的方法，又可以进一步细分为人工静态方法和自动静态方法；

动态方法是指，通过实际执行代码发现代码中潜在缺陷的方法，同样可以进一步细分为人工动态方法和自动动态方法。

这里需要注意的是，我在这篇文章中只会和你分享这四种方法具体是什么，各有何局限性和优势，分别可以覆盖哪些错误类型。而对于，具体如何用这四种方法完成代码级测试，测试用例如何设计、常用的测试工具如何使用，我会在后面两篇文章（《深入浅出之静态测试方法》和《深入浅出之动态测试方法》）中详细展开。

第一，人工静态方法

人工静态方法是指，通过人工阅读代码查找代码中潜在错误的方法，通常采用的手段包括，开发人员代码走查、结对编程、同行评审等。

理论上，人工静态方法可以发现上述五类代码错误，但实际效果却并不理想。这个方法的局限性，主要体现在以下三个方面：

1. 过度依赖于代码评审者的个人能力，同样的评审流程，发现的问题却相差悬殊；

2. 如果开发人员自行走查自己的代码，往往会存在“思维惯性”，开发过程中没有能考虑的输入和边界值，代码走查时也一样会被遗漏；
3. 由于完全依赖人工，效率普遍较低。

第二，自动静态方法

自动静态方法是指，在不运行代码的方式下，通过词法分析、语法分析、控制流分析等技术，并结合各种预定义和自定义的代码规则，对程序代码进行静态扫描发现语法错误、潜在语义错误，以及部分动态错误的一种代码分析技术。

自动静态方法可以发现语法特征错误、边界行为特征错误和经验特征错误这三类“有特征”的错误，但对于算法错误和部分算法错误这两种“无特征”的错误却无能为力。根本原因在于，自动静态方法并不清楚代码的具体业务逻辑。

目前，自动静态方法无论是在传统软件企业，还是在互联网软件企业都已经被广泛采用，往往会结合企业或项目的编码规范一起使用，并与持续集成过程紧密绑定。

你需要根据不同的开发语言，选择不同的工具。目前有很多工具都可以支持多种语言，比如 Sonar、Coverity 等，你可以根据实际需求来选择。

第三，人工动态方法

人工动态方法是指，设计代码的输入和预期的正确输出的集合，然后执行代码，判断实际输出是否符合预期。我在之前的第三篇文章 [《什么是单元测试？如何做好单元测试？》](#) 中介绍的单元测试，采用的测试方法本质上就是人工动态方法。

在代码级测试中，**人工动态方法是最主要的测试手段，可以真正检测代码的逻辑功能，其关注点是“什么样的输入，执行了什么代码，产生了什么样的输出”，所以最善于发现算法错误和部分算法错误。**

目前，不同的编程语言对应有不同的单元测试框架，比如，对 Java 语言最典型的是 Junit 和 TestNG，对于 C 语言比较常用的是 Google Test 等。

第四，自动动态方法

自动动态方法，又称自动边界测试方法，指的是基于代码自动生成边界测试用例并执行，以捕捉潜在的异常、崩溃和超时的方法。

自动动态方法，可以覆盖边界行为特征错误，通常能够发现“忘记处理某些输入”引起的错误（因为容易忘记处理的输入，往往是“边界”输入）。但是它对于发现算法错误无能为力，毕竟工具不可能了解代码所要实现的功能逻辑。

总结

作为代码级测试系列的第一篇文章，我今天主要和你分享了代码级测试中的常见代码错误类型，以及常用测试方法。

代码错误，可以划分为“有特征”的错误和“无特征”的错误两大类。其中，“有特征”的错误，又可以进一步细分为语法特征错误、边界行为特征错误和经验特征错误；而“无特征”的错误，主要包括算法错误和部分算法错误两类。

针对这五种代码错误，我将代码级测试的方法分成了静态方法和动态方法两大类。顾名思义，静态方法不需要执行实际代码，而动态方法需要通过执行具体的代码去发现代码错误。而每一类方法又可以根据执行方式，进一步细分。也因此，每种测试方法，所能覆盖的错误类型也不同，所以进行代码级测试时，你需要综合运用这些方法，并结合所在公司或者项目的编码规范一起使用。

这四类测试方法的特点，以及可以覆盖的错误类型，可以概括如下：

人工静态方法，本质上通过开发人员代码走查、结对编程、同行评审来完成的，理论上可以发现所有的代码错误，但也因为其对“测试人员”的过度依赖，局限性非常大；

自动静态方法，主要的手段是代码静态扫描，可以发现语法特征错误、边界行为特征错误和经验特征错误这三类“有特征”的错误；

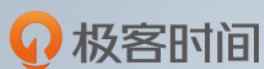
人工动态方法，就是传统意义上的单元测试，是发现算法错误和部分算法错误的最佳方式；

自动动态方法，其实就是自动化的边界测试，主要覆盖边界行为特征错误。

思考题

你所在的公司，还采用过哪些代码级测试的方法，你们又是如何具体开展的呢？

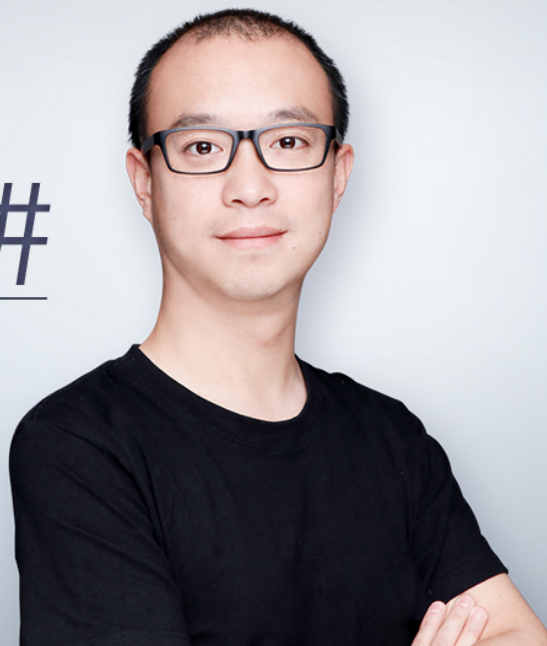
欢迎你给我留言。



软件测试52讲

从小工到专家的实战心法

茹炳晟 eBay中国研发中心
测试基础架构技术主管



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 24 | 紧跟时代步伐：微服务模式下API测试要怎么做？

下一篇 26 | 深入浅出之静态测试方法

精选留言 (13)

写留言



sylan215
2018-08-24

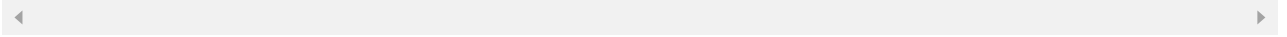
5

1.代码级测试应该算一个比较大的话题了，有专门的公司做这种静态代码检查工具的，一套好贵的说，对于测试来说，不管是人工还是自动，如果能把这五种常见错误发现了，已经很厉害了；

2.目前的五种错误类型确实是最基本最常见的了，这也是我们测试用例设计时需要优先关...
展开

作者回复：很棒的回复。人工动态本质上可以理解成单元测试，基本都是由开发自己来完成的。误报率的问题的确存在，但其实也可以反过来规范代码本身的写法，另外有些误报出现过一次后可

以标注，下次就不会发生了



Jia

2018-08-25

👍 3

对于自动静态测试，还有一个很好用的免费工具，Facebook出的Infer。常见的语言，C，Java, Pathon等都可以分析。

作者回复: 我可以去关注一下，这个我没有用过👍



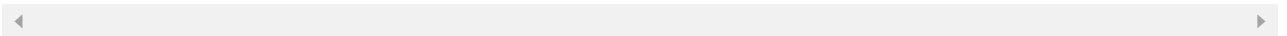
Dream.

2018-08-24

👍 3

所以开发也要学点测试，这样写的程序bug才更少~学完这篇之后眼前一亮，以后自己开发要更注意了。

作者回复: 是的，开发人员一定要建立起这些意识，能够从文中讲到的四个方法的角度去思考，这样一定可以带来更好的代码质量，从根源上解决问题



小老鼠

2018-11-07

👍 1

常见代码错误类型

有特征

第一，语法特征错误

第二，边界行为特征错误

第三，经验特征错误...

展开 ▾



颜瑞

2018-10-24

👍 1

Sonar代码扫描除了有特征点的错误类型外，目前还有安全扫描，也是代码级别的，属于“无特征”类型？





威廉姆

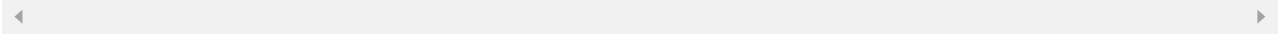
2018-08-25

👍 1

分类清晰，条理清楚，写的不错

展开 ∨

作者回复: 感谢支持👍



口水窝

2019-04-23

👍

开发和测试必须要有共同的利益，且都把这个利益放在第一位的时候，才能共进一致的去努力。比如，开发和测试把保证产品质量放在第一位，那么就会协同起来比较好，要是连这个意识都没有，恐怕只能有无穷无尽的撕逼过程。

展开 ∨



口水窝

2019-04-23

👍

以前的公司，还听说过代码走查，代码评审的。现在的公司，都没做，都是依赖测试人员太多。只能说测试和开发依赖太大，自己的代码不太负责，都是想着反正有人测，我完成我的开发任务就好。

展开 ∨



有态度的笨...

2019-04-08

👍

1、我们公司采用的是，一个开发写好的功能，会有其他两位开发进行代码的review，且review通过后，才能提交到测试人员这里进行测试（也分业务线，因为有的业务线代码review，仅仅是流于形式）。

2、测试人员拿到被测软件的时候，准备好了测试用例或者测试的检查点。然后基于用例...

展开 ∨



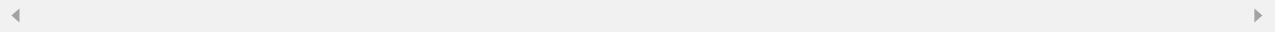
楚耳

2018-12-05

👍

老师，文中提到的基于代码自动生成边界测试用例并执行 这个是怎么做的，用开源工具还是自己开发的工具

作者回复: 一般是自己开发的小工具



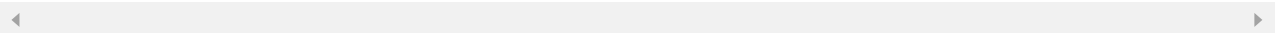
Jalyn

2018-11-22



作为一个有心做好代码级测试的初级测试工程师，需要掌握哪些必备的技术技能呢？

作者回复: 首先需要比较深入地掌握一门语言，然后还要学习至少一个单元测试框架，接下来可能就要学着使用moke



arthur

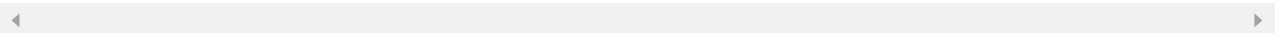
2018-09-02



打卡，写的真好，目前我们产品还木有自动动态方法，学到了 😊

展开 ▾

作者回复: 感谢支持，自动动态方法的投入产出比其实很好的，而且可以发现很多原本只有在后期才能发现的问题



涅槃Ls

2018-08-31



打卡25

展开 ▾

作者回复: 感谢支持

