黑客派

谈谈“烂代码”

作者：someone45057
原文链接：https://hacpai.com/article/1555389142178
来源网站：黑客派
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

<img src="https://static.hacpai.com/images/img-loading.svg" alt="" data-src="http://static.simpledesktops.com/uploads/desktops/2017/10/29/Apple_Park.png">
<h2 id="谈谈-烂代码-">谈谈“烂代码” </h2>
<h3 id="什么样的代码是烂代码-">什么样的代码是烂代码？ </h3>
我个人根据自己的经历总结一下烂代码的几个特征：

 不复用、封装
 相同的逻辑只会复制粘贴
 不模块化
 没有清晰的模块划分，模块与模块的耦合也没有概念
 下面举几个亲身实例：

<h4 id="在前端的发展中-组件化是一个十分重要的概念-各大框架也都给出了自己的实现-">在前端发展中，组件化是一个十分重要的概念，各大框架也都给出了自己的实现。 </h4>
这里以 Vue 为例，我曾经见过“高人”，这么写：
 <img src="https://static.hacpai.com/images/img-loading.svg" alt="15552533701.jpg" data-src="https://img.hacpai.com/file/2019/4/15552533701-67083e01.jpg?imageView2/2/interlace/1/format/jpg">
 这样写的意思就讲所有组件的样式都统一放在 assets 文件夹下然后在组件里逐个去引入。我觉得这样的方式简直糟透了。

 这样的方式完全违背了组件内聚的思想。
 在需要修改的时候还得再进入 assets 文件夹下去找对应的样式文件进行修改
 我也能理解式代码过多需要单独抽离成为一个文件，但是也应该是跟组件的 dom 在同一个目录下，如果脚本部的代码也很多，正好也抽离出来，三个文件放在一个目录下，这样才是更为“内聚”的做法

<h4 id="面对对象是一个很普及的概念">面对对象是一个很普及的概念</h4>
这里以 Java 为例，在 Java 里，如果两个类有相同逻辑的方法时，应该怎么办？
 做法一：制粘贴相同的逻辑
<pre><code class="highlight-chroma">Class A {
    void eat(){
        System.out.println('eating...');
    }
}
Class B {
    void eat(){
        System.out.println('eating...');
    }
}
</code></pre>
这么写，如果有需求改动，比如把 eating 换成 eated，这样就得分别到两个类中进行修改，更合理的做法应该是，让两个类继承自一个父类，就可以复用这个 eat()方法了
<pre><code class="highlight-chroma">Class Parent{
    void(){
        System.out.println('eating...');
    }
}
Class A extends Parent{}
Class B extends Parent {
}
</code></pre>
但是现实中我遇到很多的程序员都是复制粘贴第一种方式
 当然这里得声明，我的代码写的不一定就好，但是我觉得上面的两个例子真的让我很不能理解。