

Java面试陷阱第二波 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/145/2021_2022_Java_E9_9D_A2_E8_AF_95_c104_145292.htm

1. abstract class Name { private String name. public abstract boolean isStupidName(String name) {} } 这有何错误? 答案: 错。abstract method必须以分号结尾，且不带花括号。

2. public class Something { void doSomething () { private String s = "". int l = s.length(). } } 有错吗? 答案: 错。局部变量前不能放置任何访问修饰符 (private, public, 和protected)。final可以用来修饰局部变量(final如同abstract和strictfp, 都是非访问修饰符, strictfp只能修饰class和method而非variable)。

3. abstract class Something { private abstract String doSomething (). } 这好像没什么错吧? 答案: 错。abstract的methods不能以private修饰。abstract的methods就是让子类implement(实现)具体细节的, 怎么可以用private把abstractmethod封锁起来呢? (同理, abstract method前不能加final)。

4. public class Something { public int addOne(final int x) { return x. } } 这个比较明显。答案: 错。int x被修饰成final, 意味着x不能在addOne method中被修改。

5. public class Something { public static void main(String[] args) { Other o = new Other(). new Something().addOne(o). } public void addOne(final Other o) { o.i. } } class Other { public int i. } 和上面的很相似, 都是关于final的问题, 这有错吗? 答案: 正确。在addOne method中, 参数o被修饰成final。如果在addOne method里我们修改了o的reference(比如: o = new Other().), 那么如同上例这题也是错的。但这里修改的是o的member variable(成员变量), 而o

的reference并没有改变。 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