Module 3-面向对象编程

1. Coupling 是指一个类使用另一个类的成员的程度。
2. Loose Coupling 是理想状态，有较好的封装，互相引用最小化，限制 API 使用的范围
3. Cohesion 是指一个类被设计成单一功能或责任的程度。
4. Overriding 和 Overloading

a)第一要注意的是 private 的方法是隐藏的，它不会被继承。题目中如果父类中该方法为 private，不构成 override

b)Override 的时候对封装类不适用

c)要注意，override 的时候，父类声明抛出的异常子类可以不声明，或者声明更比父类更宽泛的异常类，或者多抛出一些 unchecked 异常

下面两个例子：

i.
```
class Super {
    public int m1() throws Exception{}
}
class Sub extends Super{
    public int m1(){}
}
//正确!
```

ii.
```
class Super {
    public int m1()
}
class Sub extends Super{
    public int m1() throws Exception{}
}
//编译错误!!!
```

iii.
```
class Super {
    public int m1() throws FileNotFoundException{}
}
class Sub extends Super{
    public int m1() throws IOException{}
}
//正确!!!
```

iv.
```
class Super {
    public int m1(){}
}
 class Sub extends Super{
    public int m1() throws NullPointerxception{}
}
```

Module 4-异常和断言

1. Exception 分为 checked 和 unchecked， RuntimeException 以下的属于 unchecked，其他的都属于 checked。
2. checked 异常是虚拟机要检查的，所以或者 throw 它，或者 catch 它。对于 unchecked 异常，虚拟机不会管，但是你也可以声明抛出或者 catch
3. 题目中会出现没有 catch 的情况，这个是合法的
4. finally 属于可有可无的。如果有 finally，那么必定会执行。除非在 catch 中调用 System.exit()。
5. 如果有多个 catch 语句，则异常的顺序应该由特殊到一般，否则产生编译错误.
6. 如果在 throw new Exception 后再添加语句，会产生编译错误，因为后面的语句根本执行不到
7. main()方法的声明中也可以加 throws XXXException
8. assert 关键字是在 JDK1.4 引入的。不能将 assert 既作关键字又做变量名。
9. 不要用 assert 来验证 public 方法的参数有效性，可以用其验证 private 方法的参数
10. 不要用 assert 来验证命令行参数的有效性
11. 不要会产生副作用的 assert 语句
12. 用 assert 来标注你认为永远不可能执行到的地方
13. 用-source 1.3 来指定编译版本，用-da 或者-ea 来关闭和打开 assert。可以指定某些类或者包被打开，如-ea:MyClass