

# 实验报告

计 63 国雨晴 2016050024

**第一题 1.1:** 没有正确使用封装概念.

Fred 的成员变量 x 是 public 的,这意味着外部类可以任意地访问修改它.应该改为 private,并且提供方便的更改 x 的 public 方法.

**第一题 1.2:** interface 和 abstract class 都不可以被实例化,只能被其他类或接口实现或继承.interface 中不能包含实例域,只能有静态方法和常量,以及一些 default 方法(在 JAVA SE 8 后),在设计框架中往往是"请求类提供一些特定的功能",如 Comparator 接口使类提供"比较"的功能;abstract class 中可以有实例域,在框架上往往是子类的"抽象"等,与之功能上密切相关.

class 中是不可以包含未定义的函数的,并且也必须对所有 implement 的 interface 和 extend 的父类中未定义的函数给出定义.否则,这个类也必须是 abstract 的.interface 也可以用 extend 关键字合并别的多个 interface,并且不用实现它们.class 只能 extend 一个父类,但可以 implement 多个 interface.

**第二题:** 代码见文件夹 2.

各对象可以调用的方法:

man 可以使用 changeSomething(返回 --count),getName(返回"man"),getDescription(返回"nothing")和 move(输出 I am moving...).

superman 可以使用 changeSomething(返回 ++count),getName(返回"superman"),getDescription(返回"nothing"),move(输出 I am flying...)和 fly(输出"

fly,I am a SuperMan").

pman 可以使用 changeSomething( 返回 --count),getName( 返回 "pman"),getDescription(返回"nothing").

psman 可以使用 changeSomething( 返回 --count),getName( 返回 "psman"),getDescription(返回"nothing").

msMan 可以使用 changeSomething( 返回 ++count),getName( 返回 "msMan"),getDescription(返回"nothing"),move(输出 I am flying...).

代码分析:

(1)

```
Man man=new Man("man","nothing");  
SuperMan sman=(SuperMan)man;
```

不可以.让 SuperMan 类的引用 sman 来指向 Man 类对象,会导致 SuperMan 可以调用子类扩展的方法和实例域,编译器会检查出这种错误.

(2)

```
Man man=new SuperMan("superman","nothing");  
SuperMan sman=(SuperMan)man;  
Man man2=(Man)sman;
```

可以.第一行和第三行用 Man 类型引用 SuperMan 对象,是允许的;第二行强制转换,用 SuperMan 类型引用替换原来 Man 类型的引用,但是由于对象还是 SuperMan 型的,这种做法也是允许的.

(3)

```
Person man=new Man("man","nothing");  
  
SuperMan sman=(SuperMan)man;
```

不可以.第一行用 Person 引用指向 Man 对象,由于 Person 是 Man 实现的接口,虽然不能实例化 Person 类,但是可以用 Person 的引用指向其子类的实例,是可以的;而第二行试图用 SuperMan 类指向 Man 类型的实例,这是非法的(同问题 1).

**第三题:**代码见文件夹 3.

思路:为达到"只创建一个实例"的效果,需要有一个静态变量记录"是否创造过实例";同时,为了以后能方便地引用这个实例,可以把这个实例的引用以静态变量的方式存在这个类中.