第二次作业实验报告

1.1 下方代码是否正确使用封装概念，说明理由。

答: 没有正确使用.成员变量x被定义为public,仍可以被从外部直接读写.

如果要达到封装的效果的话,要把x定义为private,这会使得它对外不可见,无法用object.x的方法访问.同时,最好加入setX方法等可以修改x的方法,方便外部修改x.

1.2 简要说明interface, abstract class, class之间的关系。

答: class是可以实例化的,其中所有函数声明都必须有定义;abstract class和interface中可以有不被定义的函数声明,但不可以被实例化.

abstract class其他地方和class十分相似,往往作为class的父类.

在class或者abstrat class继承一个abstract class时使用extends关键字.而interface被实现时使用implements关键字.二者都是子类(或实现接口的类)需要定义所有抽象类和接口中未定义的方法.

abstract class中可以实现一些方法,甚至可以实现所有的方法,并且可以包含实例域.而interface中,不可以包含任何实例域,只能定义一些静态常量,且在java SE 8之前不可以实现任何方法; 现在,它可以实现简单的方法(使用default修饰的默认方法和静态方法),但是依然不可以包含实例域.

一个class只能继承自一个class或abstrat class, 为了实现多继承,这个class可以implement任意个interface.

抽象类可以实现接口.接口不能实现接口,但是可以继承(extends)一个或多个接口.

抽象类的访问权限没有固定要求,和class没有什么区别.而接口的成员变量默认都是puiblic static final 的, 所有接口的方法默认都是public abstract的.

虽然不能实例化abstract class和interface, 但是可以创建一个abstract class和interface变量(引用). abstract class的变量必须指向其实现了所有抽象方法的子类对象, 类似地,interface变量也必须指向实现了该接口的对象.

使用instanceof可以检测一个对象是否是某个abstract class的子类,也可以检测其是否实现了某个interface.

abstract class主要功能是抽象其子类的共同方法,方便统一调用,或者给后续子类的实现提供一个架构.往往abstract class和其子类逻辑上紧密相关.而interface主要功能是实现另一个工具类要求的一些函数,以方便使用那个类提供的功能,例如实现了Comparable接口后,就可以调用sort方法进行排序;实现ActionListener接口可以方便地被计时器或者按钮等回调.给一个类实现一个接口,并用这个接口的变量指向这个实例,传给需要的工具类即可.

2.3 利用2.1，2.2中定义的类，将上述代码中实例man，superman，pman，psman，msMan所能够调用的方法和调用结果列出，并综合分析产生该结果的方法调用过程。

答:man可以使用的方法:

changeSomething,返回整数--count;

getName,返回字符串man;

getDescription,返回字符串nothing;

move,输出I am moving…;

superMan可以使用的方法:

changeSomething,返回整数++count;

getName,返回字符串superman;

getDescription,返回字符串nothing;

move,输出I am flying…;

fly,输出fly,I am a SuperMan;

pman可以使用的方法:

changeSomething,返回整数--count;

getName,返回字符串pman;

getDescription,返回字符串nothing;

psman可以使用的方法:

changeSomething,返回整数++count;

getName,返回字符串psman;

getDescription,返回字符串nothing;

msMan可以使用的方法:

changeSomething,返回整数++count;

getName,返回字符串msMman;

getDescription,返回字符串nothing;

move,输出I am flying…;

2.4 答: (1)不行.父类实例的引用不能转为子类引用,否则sman就可以调用fly方法,这个方法不属于该对象.

(2)可以.先用Man类型的引用指向SuperMan类型的实例,由于该实例依然是SuperMan的,用SuperMan引用指向它毫无问题.同样,第三行再转回来也没问题.

(3)不行.第一行用Person指针指向Man类型的实例是可以的;而第三行实际上是用SuperMan指针指向Man类型的实例,这样这个Man类型的实例就可以使用fly方法,显然有问题.

3 设计思路:首先需要把这个类的构造函数改为private,防止外部调用,并在使用时用静态方法getInstance方法替换这个方法.然后,在第一次调用getInstance时,新建一个实例;此后的调用需要返回同一个实例,因此需要用静态方法保存这个实例在类中.