

## 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 abstract 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 abstract 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 public 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.