**实验8 抽象类、内部类与接口**

**班级\_\_\_\_\_\_\_\_\_\_\_学号\_\_\_\_\_\_\_\_\_\_\_\_\_姓名\_\_\_\_\_\_\_\_\_\_\_\_\_**

**（一）实验目的**

1、掌握多态的设计思想及实现方法；

2、掌握接口的特点、构成及实现方法；

3、掌握内部类的特点及实现方法。

**（二）实验内容和步骤**

1、阅读程序，先判断有没有问题，如果没有，根据自己的判断写出程序的执行结果，然后运行程序，依据程序运行结果，对比分析给出程序执行流程的解释。

**(1)第1题**

class Fu

{

public void show()

{

System.out.println("fu show");

}

}

class Zi extends Fu

{

public void show()

{

System.out.println("zi show");

}

public void method()

{

System.out.println("zi method");

}

}

class Test

{

public static void main(String[] args)

{

Fu f = new Zi();

f.method();//输出zi method 编译看左边，执行看右边

}

}

**（2）第2题**

class A

{

public void show()

{

show2();

}

public void show2()

{

System.out.println("我");

}

}

class B extends A

{

public void show2()

{

System.out.println("爱");

}

}

class C extends B

{

public void show()

{

super.show();

}

public void show2()

{

System.out.println("你");

}

}

public class Test

{

public static void main(String[] args)

{

A a = new B();

a.show();//System.out.println("爱");

B b = new C();

b.show();//c.show->a.show->

}

}

}

**（3）第3题，**判断下面的赋值哪些是正确的

class Animal{}

class Dog extends Animal{}

class Cat extends Animal{}

以及

Animal p0 = new Animal();

Dog p1 = new Dog ();

Cat p2 = new Cat ();

Animal p3 = new Dog ();

Animal p4 = new Cat ();

p0 = p1;//

p1 = p2;//false

p1 = (Dog)p3;

p2 = (Cat)p4;

**（4）第4题**

abstract class AbsArgRef {

public abstract void show();

}

class AbsArgRefDemo {

public void method(B p) {

p.show();

}

}

class B extends AbsArgRef {

public abstract void show() {

System.out.println("这里是抽象类引用测试");

}

}

interface IntArgRef {

public abstract void show();

}

class IntArgRefDemo {

public C getC() {

return new C();

}

}

class C implements IntArgRef {

public void show() {

System.out.println("这里是接口引用测试");

}

}

class ReferenceTest {

public static void main(String[] args) {

IntArgRefDemo iard = new IntArgRefDemo();

IntArgRef iar = iard.getC(); // new C()

iar.show();//C.show();->System.out.println("这里是接口引用测试");

new AbsArgRefDemo().method(new B());

//System.out.println("这里是抽象类引用测试");

new IntArgRefDemo().getC().show();//C().show()System.out.println("这里是接口引用测试");

AbsArgRefDemo aard = new AbsArgRefDemo();

B b = new B();

aard.method(b); //System.out.println("这里是抽象类引用测试");

}

}

**2、编程题**

现需要开发一个战斗类游戏，游戏中涉及三个角色，每个角色均有一个ID号、血量两个属性，每个角色都有同一个移动(move)方法，此外每个角色都有一项必杀技，分别是降龙掌(surDragonPalm)、六脉剑(sixPulseSowrd)和一阳指(oneSunFinger)。请设计一个类，提供对角色ID号、血量的设置和查询，提供角色移动方法和必杀技方法，方法的输出可以使用一个打印相应方法被调用的语句来模拟，并对设计的类进行测试。要求设计的类中必须使用到面向对象设计中的抽象类和接口。

import java.util.Arrays;

import java.util.Scanner;

interface Skill{

public abstract void PersonSkill();

}

abstract class People {

private int Id;

private int Blood;

public People() {

}

public People(int Id,int Blood) {

this.Id = Id;

this.Blood = Blood;

}

public void move() {

System.out.println("i can move!");

}

public void SetId (int Id){

this.Id = Id;

}

public void SetBlood (int Blood){

this.Blood = Blood;

}

public int GetId (){

return this.Id;

}

public int GetBlood (){

return this.Blood;

}

}

class P1 extends People implements Skill{

public P1() {

}

public P1(int id,int Blood)

{

super(id,Blood);

}

public void PersonSkill() {

System.out.println("技能:降龙掌");

}

}

class P2 extends People implements Skill{

public P2() {

}

public P2(int id,int Blood)

{

super(id,Blood);

}

public void PersonSkill() {

System.out.println("技能:六脉剑");

}

}

class P3 extends People implements Skill{

public P3() {

}

public P3(int id,int Blood)

{

super(id,Blood);

}

public void PersonSkill() {

System.out.println( "技能:一阳指");

}

}

public class Game {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("初始化第一位英雄的编号");

int id =sc.nextInt();

System.out.println("初始化第一位英雄的血量");

int blood =sc.nextInt();

P1 A = new P1(id,blood);

System.out.println("血量 ："+A.GetBlood()+" id："+A.GetId());

A.move();A.PersonSkill();

System.out.println("初始化第二位英雄的编号");

id =sc.nextInt();

System.out.println("初始化第二位英雄的血量");

blood =sc.nextInt();

P2 B = new P2(id,blood);

System.out.println("血量 ："+B.GetBlood()+" id："+B.GetId());

B.move();B.PersonSkill();

System.out.println("初始化第三位英雄的编号");

id =sc.nextInt();

System.out.println("初始化第三位英雄的血量");

blood =sc.nextInt();

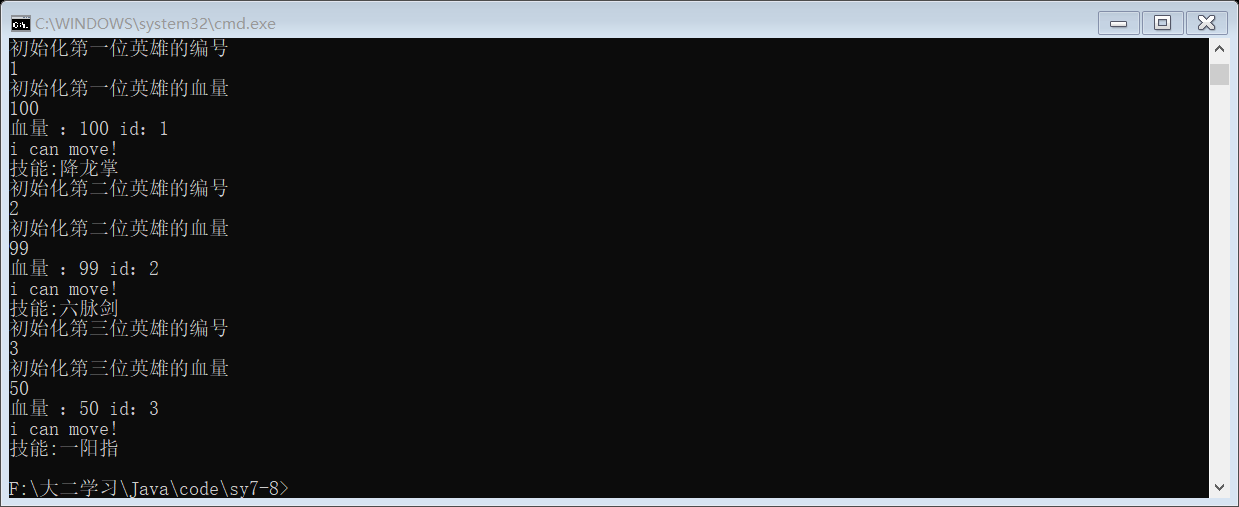
P3 C = new P3(id,blood);

System.out.println("血量 ："+C.GetBlood()+" id："+C.GetId());

C.move();C.PersonSkill();

}

}

****

**（三）存在的问题、解决办法及小结**

问题：最后一题的多处使用了代码的复制，代码的利用率低。主要原因是无法实现那个未被父类实现的子类方法。最后一题的代码有待改进！！

小结：本次实验加强了对于接口和内部类的理解。