**继承作业答案**

1. **选择题**
   1. A
   2. BD
   3. D
   4. A
   5. AD
2. **判断题**
3. √
4. ×
5. √
6. √
7. √
8. ×
9. √
10. **简答题**

答案略

1. **编码题**
   1. 编写应用程序，创建类的对象，分别设置圆的半径、圆柱体的高，计算并分别显示圆半径、圆面积、圆周长，圆柱体的体积。（7分）

**提示:**（1）编写一个圆类Circle，该类拥有：

* + - 1. 一个成员变量，radius（私有，浮点型）;//存放圆的半径；
      2. 两个构造方法

Circle（） //将半径设为0

Circle（double r ） //创建Circle对象时将半径初始化为r

* + - 1. 三个成员方法

double getArea（） //获取圆的面积

double getPerimeter（） //获取圆的周长

void show（） //将圆的关径、周长、面积输出到屏幕

（2） 编写一个圆柱体类Cylinder，它继承于上面的Circle类。还拥有：

* + 1. 一个成员变量，double hight （私有，浮点型）; //圆柱体的高；
    2. 构造方法

//创建Cylinder对象时将半径初始化为r,高度初始化为h

Cylinder（double r,double h）

* + 1. 成员方法

double getVolume（） //获取圆柱体的体积

void showVolume（） //将圆柱体的体积输出到屏幕

**public** **class** Circle {

**private** **double** radius;

**public** Circle(){ //将半径设为0

**this**.radius = 0;

}

**public** Circle(**double** r ){ //创建Circle对象时将半径初始化为r

**this**.radius = r;

}

**public** **double** getArea(){ //获取圆的面积

**return** Math.***PI*** \* **this**.radius \* **this**.radius ;

}

**public** **double** getPerimeter(){//获取圆的周长

**return** Math.***PI*** \* **this**.radius \* 2 ;

}

/\*\* 将圆的面积 ,周长 \*/

**public** **void** show(){

System.***out***.println("圆的半径:"+**this**.radius);

System.***out***.println("圆的面积:"+**this**.getArea());

System.***out***.println("圆的周长:"+**this**.getPerimeter());

}

}

**public** **class** Cylinder **extends** Circle {

**private** **double** hight;

/\*\* 构造时设置圆的高，并调用父类的构造设置半径 \*/

**public** Cylinder(**double** r,**double** h){

**super**(r);

**this**.hight = h;

}

/\*\* 获得体积 \*/

**public** **double** getVolume(){

**double** area = **super**.getArea();

**return** area \* hight;

}

/\*\*打印半径，周长，面积，体积\*/

**public** **void** showVolume(){

**super**.show();

System.***out***.println("圆的体积是："+**this**.getVolume());

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Cylinder c = **new** Cylinder(10,10);

c.showVolume();

}

}

* 1. 请使用面向对象的思想，设计自定义类，描述出租车和家用轿车的信息。

设定

1. 出租车类:

属性包括：车型，车牌，所属出租公司；方法包括：启动，停止

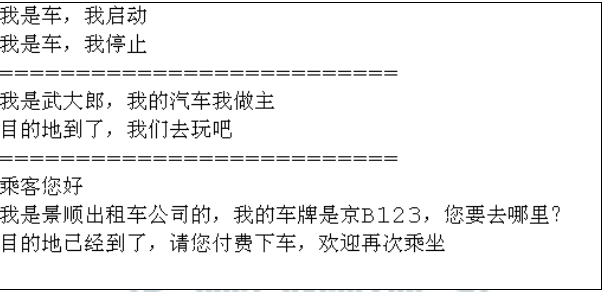
1. 家用轿车类：

属性包括：车型，车牌，车主姓名；方法包括：启动，停止

要求

1. 分析出租车和家用轿车的公共成员，提取出父类—汽车类
2. 利用继承机制，实现出租车类和家用轿车类
3. 编写测试类，分别测试汽车类，出租车类和家用轿车类对象的相关方法
4. 定义名为car的包存放汽车类，出租车类，家用轿车类和测试类

运行效果



**public** **class** Car {

**private** String type;

**private** String carNo;

**public** String getType() {

**return** type;

}

**public** **void** setType(String type) {

**this**.type = type;

}

**public** String getCarNo() {

**return** carNo;

}

**public** **void** setCarNo(String carNo) {

**this**.carNo = carNo;

}

**public** **void** start(){

System.***out***.println("我是车，我启动");

}

**public** **void** stop() {

System.***out***.println("我是车，我停止");

}

}

**public** **class** FamilyCar **extends** Car {

**private** String owner;

**public** FamilyCar(String owner) {

**this**.owner = owner;

}

**public** **void** start() {

System.***out***.println("我是"+owner+"，我的汽车我做主");

}

**public** **void** stop() {

System.***out***.println("目的地到了，我们去玩吧");

}

}

**public** **class** TaxiCar **extends** Car {

**private** String company;

**public** TaxiCar(String company) {

**this**.company = company;

}

**public** **void** start() {

System.***out***.println("乘客您好\n我是"+**this**.company+

"的，我的车牌号是"+**this**.getCarNo()+"，你要去那里？");

}

**public** **void** stop() {

System.***out***.println("目的地已经到了，请您下车付款，欢迎再次乘坐");

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Car c = **new** Car();

c.start();

c.stop();

System.***out***.println("===========================");

FamilyCar fc = **new** FamilyCar("武大郞");

fc.start();

fc.stop();

System.***out***.println("============================");

TaxiCar tc = **new** TaxiCar("景顺出租公司");

tc.setCarNo("京B123");

tc.start();

tc.stop();

}

}

* 1. 某公司要开发新游戏，请用面向对象的思想，设计游戏中的蛇怪和蜈蚣精

设定

1. 蛇怪类:

属性包括：怪物名字，生命值，攻击力

方法包括：攻击，移动（曲线移动），补血（当生命值<10时，可以补加20生命值）

1. 蜈蚣精类：

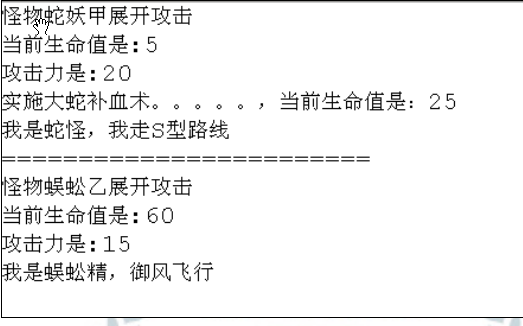
属性包括：怪物名字，生命值，攻击力

方法包括：攻击，移动（飞行移动）

要求

1. 分析蛇怪和蜈蚣精的公共成员，提取出父类—怪物类
2. 利用继承机制，实现蛇怪类和蜈蚣精类
3. 攻击方法，描述攻击状态。内容包括怪物名字，生命值，攻击力
4. 编写测试类，分别测试蛇怪和蜈蚣精的对象及相关方法
5. 定义名为mon的包存怪物类，蛇怪类，蜈蚣精类和测试类

运行效果



//妖怪类

**public** **class** Monster {

**private** String name;//怪物名

**private** **int** hp;//血量

**private** **int** aggressivity;//攻击力

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** **int** getHp() {

**return** hp;

}

**public** **void** setHp(**int** hp) {

**this**.hp = hp;

}

**public** **int** getAggressivity() {

**return** aggressivity;

}

**public** **void** setAggressivity(**int** aggressivity) {

**this**.aggressivity = aggressivity;

}

/\*\* 攻击 \*/

**public** **void** attack() {

System.***out***.println("怪物"+**this**.getName()+"展开攻击");

System.***out***.println("当前生命值："+**this**.getHp());

System.***out***.println("攻击力是："+**this**.getAggressivity());

}

/\*\* 移动 \*/

**public** **void** move() {

}

}

/\*\* 怪蛇类 \*/

**public** **class** Snake **extends** Monster {

/\*\* 加血 \*/

**public** **void** addHp(){

**this**.setHp(**this**.getHp()+20);

System.***out***.println("实施大蛇补血术......，当前的生命值是"+**this**.getHp());

}

/\*\* 重写父类的移动方法\*/

**public** **void** move() {

System.***out***.println("我是蛇，我走S线路");

}

}

/\*\* 蜈蚣类 \*/

**public** **class** Centipede **extends** Monster {

/\*\* 重写父类的移动方法\*/

**public** **void** move() {

System.***out***.println("我是蜈蚣精，我御风飞行");

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Snake sn = **new** Snake();//生产一个蛇精

sn.setName("怪蛇甲");

sn.setHp(5);

sn.setAggressivity(20);

sn.attack();//调用攻击方法

**if**(sn.getHp()<10){//当生命少于10时，加20血

sn.addHp();

}

sn.move();//蛇移动

System.***out***.println("======================");

Centipede ct = **new** Centipede();//new一个蜈蚣

ct.setName("蜈蚣精乙");

ct.setHp(60);

ct.setAggressivity(15);

ct.attack();//蜈蚣攻击

ct.move();

}

}

1. **可选题**
   1. 请用面向对象的思想，设计自定义类描述演员和运动员的信息

设定

1. 演员类:

属性包括：姓名，年龄，性别，毕业院校，代表作

方法包括：自我介绍

1. 运动员类：

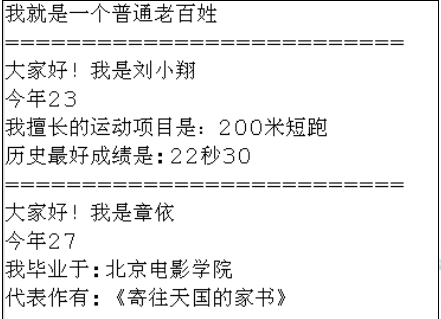
属性包括：姓名，年龄，性别，运动项目，历史最好成绩

方法包括：自我介始

要求

1. 分析演员和运动员的公共成员，提取出父类—人类
2. 利用继承机制，实现演员类和运动员类
3. 编写测试类，分别测试人类，演员类和运动员类对象及相关方法
4. 定义名为act的包存人类，演员类，运动员类和测试类

运行效果



**public** **class** Person {

//姓名，年龄，性别

String name;

**int** age;

String sex;

**public** Person(){}

**public** Person(String name, **int** age, String sex) {

**this**.name = name;

**this**.age = age;

**this**.sex = sex;

}

//介绍

**public** **void** introduction(){

System.***out***.println("大家好，我是一个老百姓");

}

}

**public** **class** Sportsman **extends** Person{//运动员类

**private** String item;

**private** String bestResult;

**public** Sportsman(){}

**public** Sportsman(String name,**int** age,String sex,String item,String best) {

**super**(name,age,sex);

**this**.item = item;

**this**.bestResult = best;

}

**public** **void** introduction() {

System.***out***.println("大家好，我是"+**this**.name);

System.***out***.println("今年"+**this**.age);

System.***out***.println("我擅长的项目是："+**this**.item);

System.***out***.println("历史最好成绩是："+**this**.bestResult);

}

}

**public** **class** Performer **extends** Person {//演员类

**private** String school;//学校

**private** String production;//作品

**public** Performer(){}

**public** Performer(String name,**int** age,String sex,String school,String pro) {

**super**(name,age,sex);

**this**.school = school;

**this**.production = pro;

}

**public** **void** introduction() {

System.***out***.println("大家好，我是"+**this**.name);

System.***out***.println("今年"+**this**.age);

System.***out***.println("我毕业于："+**this**.school);

System.***out***.println("代表作有："+**this**.production);

}

}

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Person p = **new** Person();

p.introduction();

System.***out***.println("==============");

Sportsman sm = **new** Sportsman("刘小翔",23,"男",

"200米短跑","22秒30");

sm.introduction();

System.***out***.println("==============");

Performer pf = **new** Performer("章依",27,"女","北京电影学院",

"《寄往天国的家书》");

pf.introduction();

}

}