**《Java程序设计》实验报告**

**学生姓名： 耿立博 班级： 19计算机3班 学号：2019207320123**

**实验日期： 2021.4.25 指导教师： 胡继礼、谷宗运、李芳芳**

**实验环境：win7+Jdk1.8**

1. **实验目的**

**掌握** 1. 掌握Java类的结构、类的定义、方法和属性的定义以及对象的实现；

2. 掌握构造函数的使用；方法的参数传递和返回值的用法；

3. 掌握Java类的结构、类的定义、方法和属性的定义以及对象的实现；

4. 掌握构造函数的使用；方法的参数传递和返回值的用法；

5. 掌握Java的继承及多态

**熟悉** 1. 类及其成员修饰符的使用；

2. 类变量与实例变量，以及类方法与实例方法的区别。

**3**. 类及其成员修饰符的使用；

4. 类变量与实例变量，以及类方法与实例方法的区别。

1. **实验内容**
2. 定义一个Animal类（包名cn.edu.ahtcm.model）要求如下

（1）定义成员变量：名称name，体重weight，颜色color，成员变量定义为私有的，同时生成相应的get和set方法；

（2）包括void动态方法run（）（输出“Animal run fastly”）；void eat（）（输出“Animal eat something”），以及受保护的叫方法shout()(输出“Animal shout”)

（3）定义重载方法run(String name),输出谁跑得快

（4）包括有参构造方法

（5）重写toString()方法，实现打印输出成员变量的值；

（6）在main方法里新建一个对象a（new调用构造方法），然后打印输出a的各个成员变量，再通过set方法改变name值，再调用两个run方法和eat方法

2.定义一个Dog类（包名cn.edu.ahtcm.model）要求如下

（1）继承父类Animal，增加新的成员变量：种类category

（2）有参构造方法

（3）重写父类的run（）方法（输出“dog run fastly”）；eat（）（输出“dog love bone”）

（4）在main方法里新建一个对象d1（new调用构造方法，Dog d1 = new ….），然后打印输出的d1的run和eat方法

（5）在main方法里新建一个对象d2（new调用构造方法, Animal d2 = new ….），然后打印输出的d2的run和eat方法,观察java的多态

（6）将d2 强制转换为Dog类型，并使用关键字instanceof判断d2是否是Dog类型，如果是打印输出d2的种类值。

**三、实验过程（按上述实验内容要求写出源代码及运行结果）**

**1.实验-1**

**【源码】**

**package** cn.edu.ahtcm.model;

/\*

\* ID:2019207320123

\* name:耿立博

\* date:2021.4.25

\*/

**public** **class** Animal {

//1.定义成员变量：名称name，体重weight，颜色color，成员变量定义为私有的，同时生成相应的get和set方法；

**private** String name;

**private** **double** weight;

**private** String color;

**public** String getName() {

**return** name;

}

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

**this**.name = name;

}

**public** **double** getWeight() {

**return** weight;

}

**public** **void** setWeight(**double** weight) {

**this**.weight = weight;

}

**public** String getColor() {

**return** color;

}

**public** **void** setColor(String color) {

**this**.color = color;

}

//2.包括void动态方法run（）（输出“Animal run fastly”）；void eat（）（输出“Animal eat something”），以及受保护的叫方法shout()(输出“Animal shout”)

**void** run(){

System.***out***.println("Animal run fastly");

}

**void** eat(){

System.***out***.println("Animal eat something");

}

**private** **void** shout(){

System.***out***.println("Animal shout");

}

//3.定义重载方法run(String name),输出谁跑得快

**void** run(String name){

System.***out***.println(name+" run fastly");

}

//4.包括有参构造方法

**public** Animal(String name, **double** weight, String color) {

**super**();

**this**.name = name;

**this**.weight = weight;

**this**.color = color;

}

//5.重写toString()方法，实现打印输出成员变量的值；

@Override

**public** String toString() {

**return** "Animal [name=" + name + ", weight=" + weight + ", color=" + color + "]";

}

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

// **TODO** Auto-generated method stub

//6.在main方法里新建一个对象a（new调用构造方法），然后打印输出a的各个成员变量，再通过set方法改变name值，再调用两个run方法和eat方法

Animal a = **new** Animal("富贵",3,"black");

System.***out***.println(a);

System.***out***.println(a.name);

System.***out***.println(a.weight);

System.***out***.println(a.color);

a.setName("小富贵");

a.run();

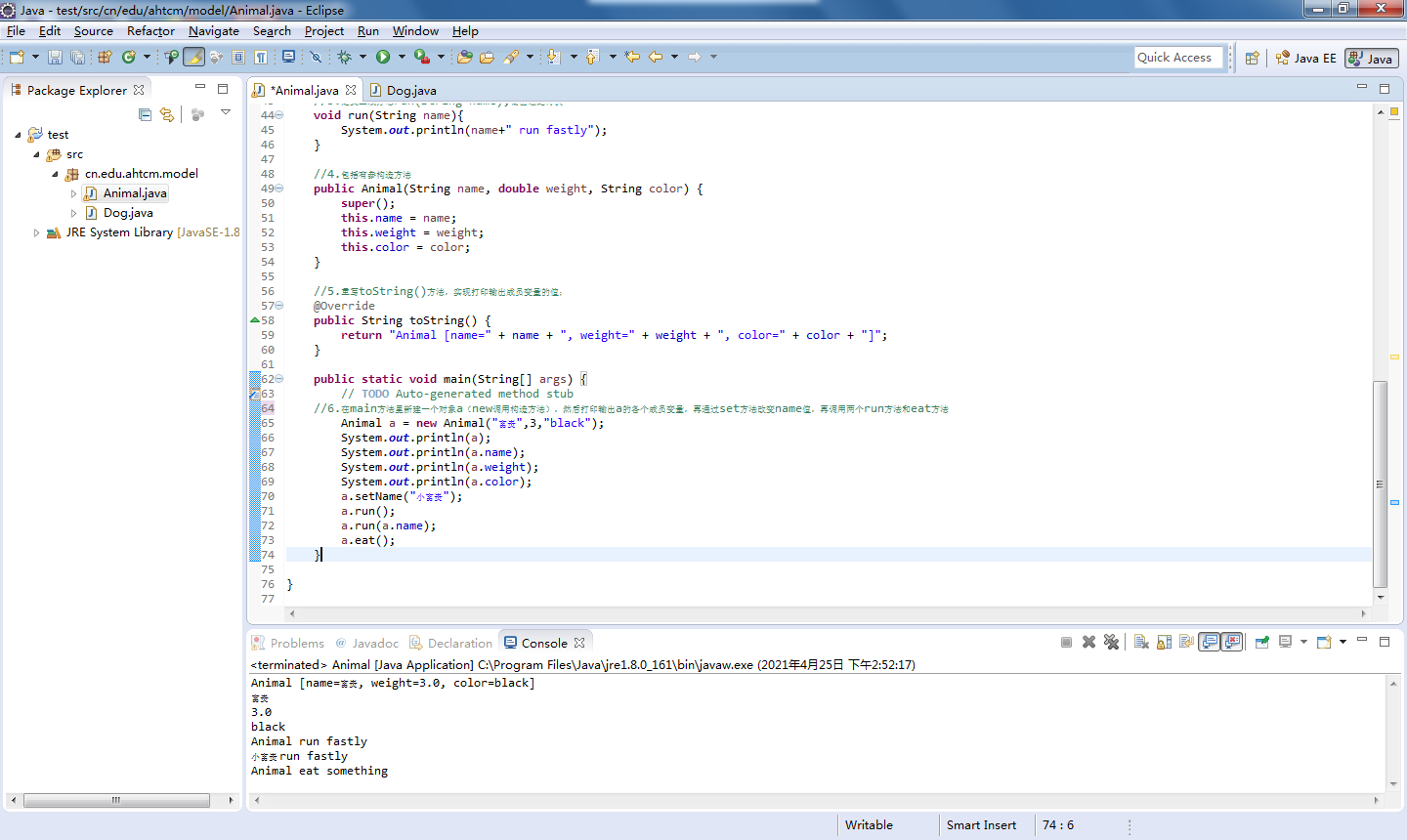
a.run(a.name);

a.eat();

}

}

**【运行结果】**



**2.实验-2**

**【源码】**

**package** cn.edu.ahtcm.model;

/\*

\* ID:2019207320123

\* name:耿立博

\* date:2021.4.25

\*/

//1.继承父类Animal，增加新的成员变量：种类category

**public** **class** Dog **extends** Animal {

String catagory;

//2.有参构造方法

**public** Dog(String name, **double** weight, String color, String catagory) {

**super**(name, weight, color);

**this**.catagory = catagory;

}

//3.重写父类的run（）方法（输出“dog run fastly”）；eat（）（输出“dog love bone”）

@Override

**void** run() {

System.***out***.println("dog run fastly");

}

@Override

**void** eat() {

System.***out***.println("dog love bone");

}

**public** Dog(String name, **double** weight, String color) {

**super**(name, weight, color);

// **TODO** Auto-generated constructor stub

}

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

// **TODO** Auto-generated method stub

//4.在main方法里新建一个对象d1（new调用构造方法，Dog d1 = new ….），然后打印输出的d1的run和eat方法

Dog d1 = **new** Dog("小玉",4,"black","金毛");

d1.run();

d1.eat();

//5.在main方法里新建一个对象d2（new调用构造方法, Animal d2 = new ….），然后打印输出的d2的run和eat方法,观察java的多态

Animal d2 = **new** Dog("小向",4,"yellow","柴犬");

d2.run();

d2.eat();

//6.将d2 强制转换为Dog类型，并使用关键字instanceof判断d2是否是Dog类型，如果是打印输出d2的种类值。

**if**(d2 **instanceof** Dog){

System.***out***.println(((Dog)d2).catagory);

}

}

}

**【运行结果】**

