**浙 江 农 林 大 学 数 计 学 院**

**JAVA程序设计实验报告**

**实 验 四: 封装和继承**

**学生姓名: 刘糠杰**

**专　 业: 智能科学与技术**

**指导教师: 戴丹**

**完成日期** （ 2024年 11月 15日）

1. **实验任务**

**## 第一题：**

某企业有很多员工(Employee)。

按照工作内容不同分教研部员工(Teacher)和行政部员工(AdminStaff)

1. 教研部根据教学的方式不同又分为讲师(Lecturer)和助教(Tutor)

2. 行政部根据负责事项不同,又分为维护专员(Maintainer),采购专员(Buyer)

3. 公司的每一个员工都编号,姓名和其负责的工作

4. 每个员工都有工作的功能，但是具体的工作内容又不一样。

package hello;

public class Test {

    public static void main(String[] args) {

    }

}

class Employee {

    String id;

    String name;

    String work;

    public Employee(String id,String name,String work) {

        this.id = id;

        this.name = name;

        this.work = work;

    }

}

class Teacher extends Employee {

    public Teacher(String id,String name,String work) {

        super(id,name,work);

    }

}

class AdminStaff extends Employee {

    public AdminStaff(String id,String name,String work) {

        super(id,name,work);

    }

}

class Maintainer extends Employee {

    public Maintainer(String id,String name,String work) {

        super(id,name,work);

    }

}

class Buyer extends Employee {

    public Buyer(String id,String name,String work) {

        super(id,name,work);

    }

}

**## 第二题：**

在某教学资源管理系统中，存在学生、老师角色会进入系统。

​学生信息和行为（名称，年龄，所在班级，查看课表，填写听课反馈fillForm）

​老师信息和行为（名称，年龄，部门名称，查看课表，发布问题publishForm）

​定义角色类作为父类包含属性（名称，年龄），行为（查看课表）

​定义子类：学生类包含属性（所在班级），行为（填写听课反馈）

​定义子类：老师类包含属性（部门名称），行为（发布问题）

package hello;

public class Test {

    public static void main(String[] args) {

    }

}

class Role{

    String name;

    int age;

    public Role(String name,int age) {

        this.name = name;

        this.age = age;

    }

    public void ViewTimetable() {

        System.out.println(this.name + "正在查看课表");

    }

}

class Student extends Role {

    String classid;

    public Student(String name,int age,String classid) {

        super(name, age);

        this.classid = classid;

    }

    public void fillForm() {

        System.out.println(this.name + "正在填写听课反馈");

    }

}

class Teacher extends Role {

    String department;

    public Teacher(String name,int age,String department) {

        super(name, age);

        this.department = department;

    }

    public void publishForm() {

        System.out.println(this.name + "正在发布问题");

    }

}

**## 第三题：**

​完成代码(按照标准格式写)，然后在测试类中测试。

​1.手机类Phone

​ 属性:品牌brand,价格price

​ 无参构造,有参构造

​ 行为:打电话call,发短信sendMessage,玩游戏playGame

​2.测试类

​ 创建Phone类对象,调用Phone类中的方法

```java

package hello;

public class Test {

    public static void main(String[] args) {

        Phone phone = new Phone("小米",999.9);

        System.out.println("手机品牌: " + phone.getBrand() + ", 价格: " + phone.getPrice());

        phone.call("张三");

        phone.sendMessage("李四", "你好！");

        phone.playGame("王者荣耀");

    }

}

class Phone {

    private String brand;

    private double price;

    public Phone() { }

    public Phone(String brand, double price) {

        this.brand = brand;

        this.price = price;

    }

    public void call(String contact) {

        System.out.println("拨打电话给: " + contact);

    }

    public void sendMessage(String contact, String message) {

        System.out.println("发送短信给: " + contact + "，内容为: " + message);

    }

    public void playGame(String game) {

        System.out.println("正在玩游戏: " + game);

    }

    public String getBrand() {

        return brand;

    }

    public double getPrice() {

        return price;

    }

}

```

图示, 示意图

描述已自动生成

**## 第四题：**

​分析以下需求，并用代码实现

​1.定义Person类

​属性：姓名name、性别gender、年龄age、国籍nationality；

​方法：吃饭eat、睡觉sleep，工作work。

​2.根据人类，创建一个学生类Student

​增加属性：学校school、学号stuNumber；

​重写工作方法（学生的工作是学习）。

​3.根据人类，创建一个工人类Worker

​增加属性：单位unit、工龄workAge；

​重写工作方法（工人的工作是盖房子）。

​4.根据学生类，创建一个学生干部类 StudentLeader

​增加属性：职务job；

​增加方法：开会meeting。

​5.编写测试类分别对上述3类具体人物进行测试。

​6.要求运行结果:

学生需要学习!

工人的工作是盖房子!

学生干部正在开会!

```java

package hello;

public class Test {

    public static void main(String[] args) {

        Student student = new Student();

        student.name = "小明";

        student.work();

        Worker worker = new Worker();

        worker.name = "老张";

        worker.work();

        StudentLeader studentLeader = new StudentLeader();

        studentLeader.name = "小红";

        studentLeader.meeting();

    }

}

class Person {

    String name;

    String gender;

    int age;

    String nationality;

    public void eat() {

        System.out.println(name + " 在吃饭");

    }

    public void sleep() {

        System.out.println(name + " 在睡觉");

    }

    public void work() {

        System.out.println(name + " 在工作");

    }

}

class Student extends Person {

    String school;

    String stuNumber;

    @Override

    public void work() {

        System.out.println("学生需要学习!");

    }

}

class Worker extends Person {

    String unit;

    int workAge;

    @Override

    public void work() {

        System.out.println("工人的工作是盖房子!");

    }

}

class StudentLeader extends Student {

    String job;

    public void meeting() {

        System.out.println("学生干部正在开会!");

    }

}

```

图片包含 图示

描述已自动生成

**## 第五题：**

​1.定义项目经理类

属性：姓名 工号 工资 奖金

行为：工作work

2.定义程序员类

属性：姓名 工号 工资

行为：工作work

​3.向上抽取一个父类,让这两个类都继承这个父类,共有的属性写在父类中，子类重写父类中的方法

​4.编写测试类:完成这两个类的测试

package hello;

public class Test {

    public static void main(String[] args) {

        ProjectManager pm = new ProjectManager("张三", "P001", 12000, 3000);

        Programmer programmer = new Programmer("李四", "Q0001", 8000);

        pm.work();

        programmer.work();

    }

}

class Employee {

    protected String name;

    protected String employeeId;

    protected double salary;

    public Employee(String name, String employeeId, double salary) {

        this.name = name;

        this.employeeId = employeeId;

        this.salary = salary;

    }

    public void work() {

    };

}

class ProjectManager extends Employee {

    private double bonus;

    public ProjectManager(String name, String employeeId, double salary, double bonus) {

        super(name, employeeId, salary);

        this.bonus = bonus;

    }

    @Override

    public void work() {

        System.out.println(name + " (工号: " + employeeId + ") 是项目经理，正在工作，工资: " + salary + "，奖金: " + bonus);

    }

}

class Programmer extends Employee {

    public Programmer(String name, String employeeId, double salary) {

        super(name, employeeId, salary);

    }

    @Override

    public void work() {

        System.out.println(name + " (工号: " + employeeId + ") 是程序员，正在工作，工资: " + salary);

    }

}



**## 第六题：**

​根据需求完成代码:

​1.定义动物类

​属性：年龄，颜色

行为:eat(String something)方法(无具体行为,不同动物吃的方式和东西不一样,something表示吃的东西)

​生成空参有参构造，set和get方法

​2.定义狗类继承动物类

​行为:eat(String something)方法,看家lookHome方法(无参数)

​3.定义猫类继承动物类

​行为:eat(String something)方法,逮老鼠catchMouse方法(无参数)

​4.定义Person类

​属性：姓名，年龄

​行为：keepPet(Dog dog,String something)方法

​功能：喂养宠物狗，something表示喂养的东西

​该方法调用后打印结果为：颜色为黑色的2岁的狗，在吃骨头

​行为：

keepPet(Cat cat,String something)方法

功能：喂养宠物猫，something表示喂养的东西

​该方法调用后打印结果为：颜色为白色的2岁的猫，在吃小鱼干

​5.生成空参有参构造，set和get方法

​6.测试以上方法

```java

package hello;

public class Test {

    public static void main(String[] args) {

        Dog dog = new Dog(2, "黑色");

        Cat cat = new Cat(2, "白色");

        Person person = new Person("小张", 30);

        person.keepPet(dog, "骨头");

        person.keepPet(cat, "小鱼干");

    }

}

class Animal {

    private int age;

    private String color;

    public Animal() {

    }

    public Animal(int age, String color) {

        this.age = age;

        this.color = color;

    }

    public void eat(String something) {

    }

    public int getAge() {

        return age;

    }

    public void setAge(int age) {

        this.age = age;

    }

    public String getColor() {

        return color;

    }

    public void setColor(String color) {

        this.color = color;

    }

}

class Cat extends Animal {

    public Cat(int age, String color) {

        super(age, color);

    }

    @Override

    public void eat(String something) {

        System.out.println("颜色为" + getColor() + "的" + getAge() + "岁的猫，在吃" + something);

    }

    public void catchMouse() {

        System.out.println("猫在逮老鼠");

    }

}

class Dog extends Animal {

    public Dog(int age, String color) {

        super(age, color);

    }

    @Override

    public void eat(String something) {

        System.out.println("颜色为" + getColor() + "的" + getAge() + "岁的狗，在吃" + something);

    }

    public void lookHome() {

        System.out.println("狗在看家");

    }

}

class Person {

    private String name;

    private int age;

    public Person() {

    }

    public Person(String name, int age) {

        this.name = name;

        this.age = age;

    }

    public void keepPet(Dog dog, String something) {

        dog.eat(something);

    }

    public void keepPet(Cat cat, String something) {

        cat.eat(something);

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public int getAge() {

        return age;

    }

    public void setAge(int age) {

        this.age = age;

    }

}

```

文本

描述已自动生成