

天津商业大学

**宝德学院**

JAVA程序设计

课程实验报告

专业：\_\_\_\_计算机科学与技术\_\_\_\_\_

班级：\_\_\_\_\_\_\_\_\_1701班\_\_\_\_\_\_\_\_\_

学号：\_\_\_\_\_\_\_\_17502107\_\_\_\_\_\_\_\_\_

姓名：\_\_\_\_\_\_\_\_\_刘文正\_\_\_\_\_\_\_\_\_\_

1. 实验名称：

继承

1. 实验内容：

实验教材P58，7.3.1、7.3.3

1. 实验目的：
2. 掌握面继承的实现和继承的作用
3. 掌握方法的重写
4. 掌握继承关系中的构造方法和子类对象的构造过程
5. 掌握this、super关键字的使用
6. 程序设计与实现：

1、实验7.3.1 公司雇员类的封装

1. 程序代码：

package com.java.company;

public class Employee {

private String name;

private String no;

private double salary;

public Employee(String name,String no){

this.name=name;

this.no=no;

}

public void printInfo(){

System.out.println("姓名："+name+"\n员工号："+no);

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

}

package com.java.company;

public class HourlyEmployee extends Employee{

private double salaryPerHour;

private int hourPerMonth;

public HourlyEmployee(String name, String no,double salarePerHour) {

super(name, no);

this.salaryPerHour=salarePerHour;

}

public void setHourPerMonth(int hourPerMonth) {

this.hourPerMonth = hourPerMonth;

}

public void computerSalary(){

if(this.hourPerMonth<160){

setSalary(salaryPerHour \* this.hourPerMonth);

}else{

setSalary((this.salaryPerHour\* (hourPerMonth-160)\*1.5)+(160\*this.salaryPerHour));

}

System.out.println("工作时长为："+hourPerMonth+"\n工资为："+getSalary());

}

public void printInfo(){

super.printInfo();

System.out.println("工种：计时工");

}

}

package com.java.company;

public class Manager extends Employee{

private String level;

public Manager(String name, String no,String level) {

super(name, no);

this.level=level;

}

public void computeSalary(){

if(level.equals("经理")){

setSalary(10000);

}else if(level.equals("副经理")){

setSalary(6000);

}else if(level.equals("车间主任")){

setSalary(4000);

}else{

setSalary(3000);

}

System.out.println("工资"+getSalary());

}

public void printInfo(){

super.printInfo();System.out.println("职位："+level);

}

}

package com.java.company;

public class Test {

public static void main(String[] args) {

Manager m= new Manager("张三","001","经理");

m.printInfo();

m.computeSalary();

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

HourlyEmployee he = new HourlyEmployee("李四", "055", 30);

he.printInfo();

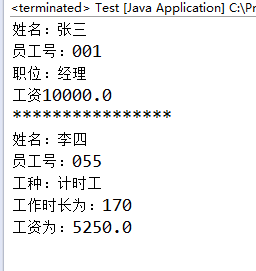
he.setHourPerMonth(170);

he.computerSalary();

}

}

1. 运行结果：



2、实验7.3.3 饲养员喂养动物

1. 程序代码：

package com.java.feedAnimal;

public class Animal {

public void eat(){

System.out.println("吃饭时间到了,");

}

}

package com.java.feedAnimal;

public class Bone extends Food {

public Bone(int weight){

super(weight);

}

public String toString(){

return "Bone";

}

}

package com.java.feedAnimal;

public class Cat extends Animal {

public void eat(Fish fish){

eat();

System.out.println(this+"喜欢吃"+fish);

}

public String toString(){

return "Cat";

}

}

package com.java.feedAnimal;

public class Dog extends Animal {

public void eat(Bone bone){

eat();

System.out.println(this+"喜欢吃"+bone);

}

public String toString(){

return "Dog";

}

}

package com.java.feedAnimal;

public class Feeder {

private String name;

public Feeder(String name){

this.name=name;

}

public void feed(Cat cat,Fish fish){

cat.eat(fish);

System.out.println("饲养员"+name+"拿着"+fish.getWeight()+"克"+fish+"喂养"+cat+"。");

}

public void feed(Dog dog,Bone bone){

dog.eat(bone);

System.out.println("饲养员"+name+"拿着"+bone.getWeight()+"克"+bone+"喂养"+dog+"。");

}

}

package com.java.feedAnimal;

public class Fish extends Food {

public Fish(int weight){

super(weight);

}

public String toString(){

return "Fish";

}

}

package com.java.feedAnimal;

public class Food {

private int weight;

public Food(int weight){

this.weight=(weight);

}

public int getWeight() {

return weight;

}

}

package com.java.feedAnimal;

public class TestDemo {

public static void main(String[] args){

Feeder feeder = new Feeder("张三");

Dog dog = new Dog();

Bone bone = new Bone(500);

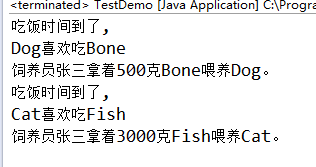
feeder.feed(dog,bone);

feeder.feed(new Cat(),new Fish(3000));

}

}

1. 运行结果：



1. 实验体会：

更加详细的了解了继承的用法，使java程序代码更加简洁，方便阅读。