**实验报告**

**学院（系）名称：**计算机科学与工程学院

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 王帆 | | **学号** | | 20152180 | | | **专业** | | 计算机科学与技术 | |
| **班级** | | 2015级1班 | | **实验项目** | | 实验二：工资管理软件设计 | | | | | | |
| **课程名称** | | | | Java程序设计 | | | | | **课程代码** | | 0667056 | |
| **实验时间** | | | | 2018年 10 月25日第1、2节 | | | | | **实验地点** | | 7-219 | |
| 考核标准 | 实验过程  25分 | | 程序运行  20分 | | 回答问题  15分 | | 实验报告  30分 | 特色  功能  5分 | | 考勤违纪情况  5分 | **成绩** |  |
| 成绩栏 |  | |  | |  | |  |  | |  | 其它批改意见:  教师签字： | |
| 考核内容 | 评价在实验课堂中的表现，包括实验态度、编写程序过程等内容等。 | | □功能完善,  □功能不全  □有小错  □无法运行 | | ○正确  ○基本正确  ○有提示  ○无法回答 | | ○完整  ○较完整  ○一般  ○内容极少  ○无报告 | ○有  ○无 | | ○有  ○无 |
| **一、实验目的**  按照图中的继承关系定义每一个类，并最终显示所有人的详细信息。  **二、实验题目与要求**  某公司由6人组成，他们分别是：   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 序号  No. | 姓名  name | 地 址  address | 电 话  phone | 职 务  title | 社会安全号  socialSN | 基本工资  payRate | | 1 | Sam | 123 Main Line | 555-0469 | manager | 123-45-6789 | $2423.07 | | 2 | Peter | 456 Off Line | 555-0101 | employee | 987-65-4321 | $1246.15 | | 3 | Mary | 789 Off Rocker | 555-0690 | employee | 010-20-3040 | $1169.23 | | 4 | Cliff | 678 Fifth Ave. | 555-0000 | hourly | 958-47-3625 | $10.55 | | 5 | Al | 987 Suds Ave. | 555-8374 | volunteer | 无 | $0.00 | | 6 | Gus | 321 Off Line | 555-7282 | volunteer | 无 | $0.00 |   其中，Sam是经理，每月除得到基本工资外，还获得$500.00的红利；Peter和Mary是合同工，每月拿基本工资；Cliff是小时工，他的每月的工资额=基本工资×工作小时数，他的工作小时数为每月40小时；而Al和Gus是义工，不拿工资。图1反应了上述情况。  StaffMember是抽象类，有一个抽象方法pay。  Manager类中的bonus代表红利；awardBonus方法用来增加红利。  Hourly类中的hoursWorked代表一个月工作的小时数；addHours方法用来增加小时数。  Staff类中的main方法用来驱动整个程序的运行；payDetail方法用来显示所有人的详细信息；构造方法用来初始化上述6个人，并将他们存储在staffMember类型的数组中。  每个类都有toString方法，用来返回该类的详细信息。要求你按照此图的继承关系定义每一个类，并最终显示所有人的详细信息。显示结果如下：  姓名：Sam  地址：123 Main Line  电话：555-0469  社会安全号：123-45-6789  工资：$2923.07  -----------------------------------------  姓名：Peter  地址：456 Off Line  电话：555-0101  社会安全号：987-65-4321  工资：$1246.15  -----------------------------------------  姓名：Mary  地址：789 Off Rocker  电话：555-0690  社会安全号：010-20-3040  工资：$1169.23  -----------------------------------------  姓名：Cliff  地址：678 Fifth Ave.  电话：555-0000  社会安全号：958-47-3625  工作小时数：40  工资：$422.00  -----------------------------------------  姓名：Al  地址：987 Suds Ave.  电话：555-8374  多谢！  -----------------------------------------  姓名：Gus  地址：321 Off Line  电话：555-7282  多谢！  -----------------------------------------    图1 工资管理软件——类图   1. **实验过程与实验结果**   **设计思路：**  根据类图，对软件结构进行组织与设计，构建对应类并实现相应方法。  **实现过程：**   1. ***StaffMember*是抽象类，包括抽象方法pay()。构造函数中初始化*StaffMember*成员，重载toString（）函数。**   **public** **abstract** **class** StaffMember {  **protected** String name;  **protected** String address;  **protected** String phone;    **public** StaffMember(String name, String address, String phone) {  **this**.name = name;  **this**.address = address;  **this**.phone = phone;  }    **public** **abstract** **double** pay();    @Override  **public** String toString() {  **return** "StaffMember [name=" + name + ", address=" + address + ", phone=" + phone + "]";  }  }   1. **Employee类继承抽象类*StaffMember*。构造函数中利用super关键字调用父类的构造函数的toString()方法，用来返回该类的详细信息。**   **public** **class** Employee **extends** StaffMember {  **public** String sociaISN;  **public** **double** payRate;  **public** Employee(String name, String address, String phone, **double** payRate, String sociaISN) {  **super**(name, address, phone);  **this**.payRate = payRate;  **this**.sociaISN = sociaISN;  }  @Override  **public** **double** pay() {  **return** payRate;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工资：$" + pay()  + "\n------------------------";  }  }   1. **Manager类中的bonus代表红利；awardBonus方法用来增加红利。Manager继承StaffMember。构造函数中利用super关键字调用父类的构造函数。toString方法，用来返回该类的详细信息。**   **public** **class** Manager **extends** Employee {  **double** bonus = 0;  **public** Manager(String name, String address, String phone, **double** payRate, String sociaISN, **double** bonus) {  **super**(name, address, phone, payRate, sociaISN);  **this**.bonus = bonus;  }  **public** **void** awardBonus(**double** bonus) {  **this**.bonus += bonus;  }  **public** **double** pay() {  **return** payRate + bonus;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工资：$" + pay()  + "\n------------------------";  }  }   1. **Volunter类继承抽象类StaffMember。构造函数中利用super关键字调用父类的构造函数。toString方法，用来返回该类的详细信息。**   **public** **class** Volunter **extends** StaffMember {  **public** Volunter(String name, String address, String phone) {  **super**(name, address, phone);  }  @Override  **public** **double** pay() {  **return** 0;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n多谢！\n------------------------";  }  }   1. **Hourly类中的hoursWorked代表一个月工作的小时数；addHours方法用来增加小时数。Hourly继承Employee。构造函数中利用super关键字调用父类的构造函数。toString方法，用来返回该类的详细信息。**   **public** **class** Hourly **extends** Employee {  **int** hoursWorked = 0;  **public** Hourly(String name, String address, String phone, **double** payRate, String sociaISN, **int** hoursWorked) {  **super**(name, address, phone, payRate, sociaISN);  **this**.hoursWorked = hoursWorked;  }  **public** **void** addHours(**int** hours) {  **this**.hoursWorked += hours;  }  **public** **double** pay() {  **return** payRate \* hoursWorked;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工作小时数：" + hoursWorked  + "\n工资：$" + pay() + "\n------------------------";  }  }   1. **Staff类中的main方法用来驱动整个程序的运行；payDetail方法用来显示所有人的详细信息；构造方法用来初始化上述6个人，并将他们存储在staffMember类型的数组中。**   **public** **class** Staff {  **public** **static** **void** payDetail(StaffMember obj) {  System.***out***.println(obj.toString());  }  **public** **static** **void** main(String[] args) {  StaffMember[] staffList = **new** StaffMember[6];  staffList[0] = **new** Manager("Sam", "123 Main Line", "555-0469", 2423.07, "123-45-6789", 500);  staffList[1] = **new** Employee("Peter", "456 Off Line", "555-0101", 1246.15, "987-65-4321");  staffList[2] = **new** Employee("Mary", "789 Off Rocker", "555-0690", 1169.23, "010-20-3040");  staffList[3] = **new** Hourly("Cliff", "678 Fifth Ave", "555-0000", 10.55, "958-47-3625", 40);  staffList[4] = **new** Volunter("Al", "987 Suds Ave", "555-8374");  staffList[5] = **new** Volunter("Gus", "321 Off Line", "555-7282");  **for** (**int** i = 0; i < staffList.length; i++) {  *payDetail*(staffList[i]);  }  }  **public** Staff() {  }  }  **示例与演示：**    图 演示结果  **四、收获与体会**   1. 掌握了Java中面向对象设计的基本思路； 2. 掌握了继承、封装与多态的基本思路。 3. 能够使用toString()方法进行对象的格式化操作。   **五、源代码清单**  **// Employee**  **package** edu.tjut.salary;  **public** **class** Employee **extends** StaffMember {  **public** String sociaISN;  **public** **double** payRate;  **public** Employee(String name, String address, String phone, **double** payRate, String sociaISN) {  **super**(name, address, phone);  **this**.payRate = payRate;  **this**.sociaISN = sociaISN;  }  @Override  **public** **double** pay() {  **return** payRate;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工资：$" + pay()  + "\n------------------------";  }  }  **// Hourly**  **package** edu.tjut.salary;  **public** **class** Hourly **extends** Employee {  **int** hoursWorked = 0;  **public** Hourly(String name, String address, String phone, **double** payRate, String sociaISN, **int** hoursWorked) {  **super**(name, address, phone, payRate, sociaISN);  **this**.hoursWorked = hoursWorked;  }  **public** **void** addHours(**int** hours) {  **this**.hoursWorked += hours;  }  **public** **double** pay() {  **return** payRate \* hoursWorked;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工作小时数：" + hoursWorked  + "\n工资：$" + pay() + "\n------------------------";  }  }  **// Manager**  **package** edu.tjut.salary;  **public** **class** Manager **extends** Employee {  **double** bonus = 0;  **public** Manager(String name, String address, String phone, **double** payRate, String sociaISN, **double** bonus) {  **super**(name, address, phone, payRate, sociaISN);  **this**.bonus = bonus;  }  **public** **void** awardBonus(**double** bonus) {  **this**.bonus += bonus;  }  **public** **double** pay() {  **return** payRate + bonus;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n社区安全号：" + sociaISN + "\n工资：$" + pay()  + "\n------------------------";  }  }  **// StaffMember**  **package** edu.tjut.salary;  **public** **abstract** **class** StaffMember {  **protected** String name;  **protected** String address;  **protected** String phone;    **public** StaffMember(String name, String address, String phone) {  **this**.name = name;  **this**.address = address;  **this**.phone = phone;  }    **public** **abstract** **double** pay();    @Override  **public** String toString() {  **return** "StaffMember [name=" + name + ", address=" + address + ", phone=" + phone + "]";  }  }  **// Volunter**  **package** edu.tjut.salary;  **public** **class** Volunter **extends** StaffMember {  **public** Volunter(String name, String address, String phone) {  **super**(name, address, phone);  }  @Override  **public** **double** pay() {  **return** 0;  }  **public** String toString() {  **return** "姓名：" + name + "\n地址：" + address + "\n电话：" + phone + "\n多谢！\n------------------------";  }  }  **// Staff**  **package** edu.tjut.test;  **import** edu.tjut.salary.Employee;  **import** edu.tjut.salary.Hourly;  **import** edu.tjut.salary.Manager;  **import** edu.tjut.salary.StaffMember;  **import** edu.tjut.salary.Volunter;  **public** **class** Staff {  **public** **static** **void** payDetail(StaffMember obj) {  System.out.println(obj.toString());  }  **public** **static** **void** main(String[] args) {  StaffMember[] staffList = **new** StaffMember[6];  staffList[0] = **new** Manager("Sam", "123 Main Line", "555-0469", 2423.07, "123-45-6789", 500);  staffList[1] = **new** Employee("Peter", "456 Off Line", "555-0101", 1246.15, "987-65-4321");  staffList[2] = **new** Employee("Mary", "789 Off Rocker", "555-0690", 1169.23, "010-20-3040");  staffList[3] = **new** Hourly("Cliff", "678 Fifth Ave", "555-0000", 10.55, "958-47-3625", 40);  staffList[4] = **new** Volunter("Al", "987 Suds Ave", "555-8374");  staffList[5] = **new** Volunter("Gus", "321 Off Line", "555-7282");  **for** (**int** i = 0; i < staffList.length; i++) {  payDetail(staffList[i]);  }  }  **public** Staff() {  }  } | | | | | | | | | | | | |