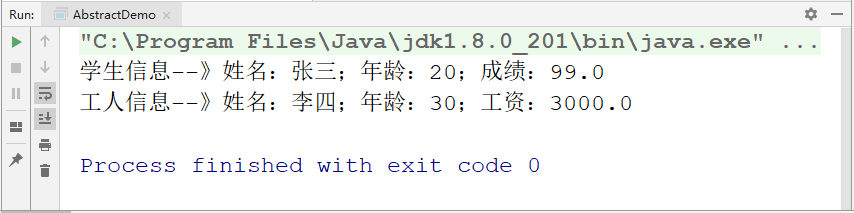
【案例4-4】学生和老师

**【案例介绍】**

**1.案例描述**

在班级中上课时，老师在讲台上讲课，偶有提问，会点名学生回答问题。虽然老师和学生都在讲话，但讲话的具体内容却不相同。本案例要求使用抽象类的知识编写一个程序实现老师上课的情景。

**2.运行结果**

运行结果

**【案例目标】**

* 学会“学生和老师”的实现思路。
* 独立完成“学生和老师”的源代码编写、编译及运行。
* 掌握方法的重写。
* 掌握抽象类的应用。

**【案例分析】**

（1）定义一个抽象类Person，在Person类中声明两个属性：name和age，并设置其对应的getter方法，用于获取人的姓名和年龄；在Person类中声明一个有参构造方法，用于对Person类中的属性进行初始化；在Person类中声明一个say()方法和一个getContent()方法，在say()方法中调用getContent()方法。

（2）定义类 Student，并继承Person类。在Student类中声明一个属性score；声明Student类的构造方法并重写Person类的getContent()方法。

（3）定义类 Worker，并继承Person类。在Worker类中声明一个属性salary；声明Worker类的构造方法并重写Person类的getContent()方法。

（6）在main() 方法中使用父类引用指向子类对象的方式（向上转型），分别使用Person类和Worker类实例化两个Person对象per1和per2，并分别使用per1和per2调用say()方法。

**【案例实现】**

AbstractDemo.java

1. abstract class Person {
2. private String name;
3. private int age;
4. public Person(String name, int age) {
5. this.name = name;
6. this.age = age;
7. }
8. public String getName() {
9. return name;
10. }
11. public int getAge() {
12. return age;
13. }
14. public void say(){
15. System.out.println(this.getContent());
16. }
17. public abstract String getContent();
18. }
19. class Student extends Person{
20. private float score;
21. public Student(String name, int age,float score) {
22. super(name, age);
23. this.score = score;
24. }
25. @Override
26. public String getContent() {
27. return "学生信息--》姓名："+super.getName()+"；年龄：
28. "+super.getAge()+"；成绩："+this.score;
29. }
30. }
31. class Worker extends Person{
32. private float salary;
33. public Worker(String name, int age,float salary) {
34. super(name, age);
35. this.salary = salary;
36. }
37. @Override
38. public String getContent() {
39. return "工人信息--》姓名："+super.getName()+"；年龄：
40. "+super.getAge()+"；工资："+this.salary;
41. }
42. }
43. public class AbstractDemo{
44. public static void main(String[] args){
45. Person per1 = null;
46. Person per2 = null;
47. per1 = new Student("张三",20,99.0f);
48. per2 = new Worker("李四",30,3000.0f);
49. per1.say();
50. per2.say();
51. }
52. }

上述代码中，第1~18行代码定义了一个抽象类Person，在Person类中声明了name和age属性和其对应的getter方法，并声明了say()方法和getContent()方法，在say()方法中调用了getContent()方法；第19~30行代码定义了一个Student类并继承Person类，在Student类中声明了score属性，声明了Student类的构造方法并重写Person类的getContent()方法；第31~42行代码定义了一个Worker类并继承Person类，在Worker类中声明了salary属性，声明了Worker类的构造方法并重写Person类的getContent()方法；第45~48行代码分别使用向上转型的方式实例化了两个Person对象per1和per2，并分别使用per1和per2调用say()方法。