实验一 java类与对象

1. 粘贴程序代码（可截图）。

1.1Employee.java

1. public class Employee {
2. private String name,gender;
3. private int id,age;
4. *// 尽量保持函数参数与成员变量顺序一致*
5. Employee(String name,String gender,int id,int age){
6. this.name = name;
7. this.gender = gender;
8. this.id = id;
9. this.age = age;
10. }
11. *// 这里的this可以省略*
12. String getName(){
13. return this.name;
14. }
15. String getGender(){
16. return this.gender;
17. }
18. int getId(){
19. return this.id;
20. }
21. int getAge(){
22. return this.age;
23. }
24. *// 如果返回值是Employee类型,则可以进行链式调用*
25. Employee setName(String name){
26. this.name = name;
27. *//调用远程服务,可能是对数据库的操作,可以使用boolean类型判断操作成功还是失败*
28. *// employee.setname(name);*
29. return this;
30. }
31. *//   Employee setName(String name){*
32. *//     // 链式调用*
33. *//     this.name = name;*
34. *//     //调用远程服务,可能是对数据库的操作,可以使用boolean类型判断操作成功还是失败*
35. *//     employee.setname("张三").setId();*
36. *//     return this;*
37. *// }*
38. Employee setGender(String gender){
39. this.gender = gender;
40. return this;
41. }
42. Employee setId(int id){
43. this.id = id;
44. return this;
45. }
46. Employee setAge(int age){
47. this.age = age;
48. return this;
49. }
50. @Override
51. public String toString(){
52. String str = "Employee [name=" + name + ", gender=" + gender + ", id=" + id + ", age=" + age + "]";
53. return str;
54. }
55. }

2.2Test.java

public class Test {

    public static void main(String[] args) {

        String name = "张三";

        String gender = "男";

        int id = 202300001;

        int age = 22;

        Employee employee = new Employee(name,gender,id,age);

        System.out.println(employee);

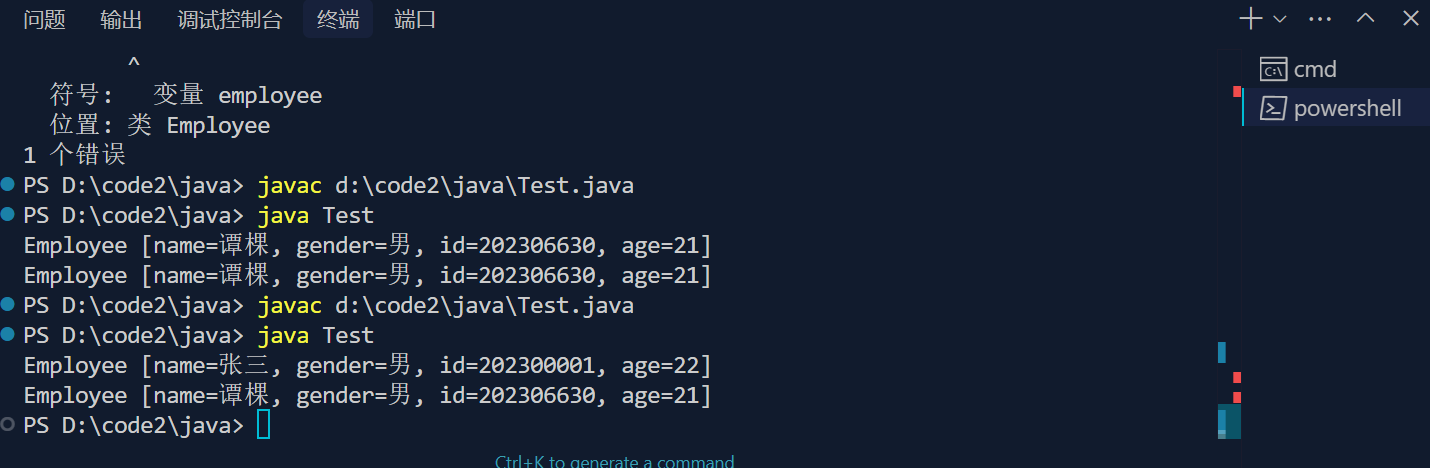
        employee.setName("谭棵").setGender("男").setId(202306630).setAge(21);

        System.out.println(employee);

    }

}

2.3运行结果



1. 粘贴程序的输出信息。

## 实验二 继承、接口与多态

1. 粘贴程序代码（可截图）。
2. 粘贴程序的输出信息。

实验三 异常处理

1. 粘贴程序代码（可截图）。
2. 粘贴程序的输出信息。

实验四 输入输出

1. 粘贴程序代码（可截图）。
2. 粘贴程序的输出信息。