实验一 java类与对象

1. 粘贴程序代码（可截图）。

1.1Employee.java

public class Employee {

    private String name,gender;

    private int id,age;

*// 尽量保持函数参数与成员变量顺序一致*

    Employee(String name,String gender,int id,int age){

        this.name = name;

        this.gender = gender;

        this.id = id;

        this.age = age;

    }

*// 这里的this可以省略*

    String getName(){

        return this.name;

    }

    String getGender(){

        return this.gender;

    }

    int getId(){

        return this.id;

    }

    int getAge(){

        return this.age;

    }

*// 如果返回值是Employee类型,则可以进行链式调用*

    Employee setName(String name){

        this.name = name;

*//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*

*// employee.setname(name);*

        return this;

    }

*//   Employee setName(String name){*

*//     // 链式调用*

*//     this.name = name;*

*//     //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*

*//     employee.setname("张三").setId();*

*//     return this;*

*// }*

    Employee setGender(String gender){

        this.gender = gender;

        return this;

    }

    Employee setId(int id){

        this.id = id;

        return this;

    }

    Employee setAge(int age){

        this.age = age;

        return this;

    }

    @Override

    public String toString(){

        String str = "Employee [name=" + name + ", gender=" + gender + ", id=" + id + ", age=" + age + "]";

        return str;

    }

}

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);

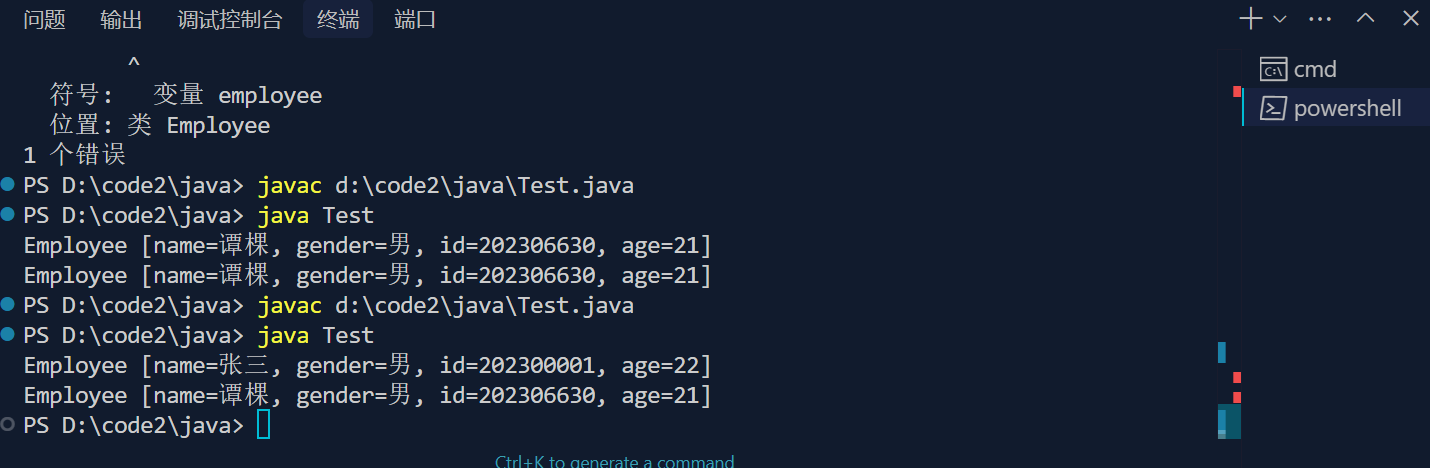
        System.out.println(employee);

    }

}

1. 粘贴程序的输出信息。

2.3运行结果



## 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。