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);
    }
}
```

2. 粘贴程序的输出信息。

2.3 运行结果

```
问题    输出    调试控制台    终端    端口
          ^
  符号:   变量 employee
  位置: 类 Employee
1 个错误
PS D:\code2\java> javac d:\code2\java\Test.java
PS D:\code2\java> java Test
Employee [name=谭棵, gender=男, id=202306630, age=21]
Employee [name=谭棵, gender=男, id=202306630, age=21]
PS D:\code2\java> javac d:\code2\java\Test.java
PS D:\code2\java> java Test
Employee [name=张三, gender=男, id=202300001, age=22]
Employee [name=谭棵, gender=男, id=202306630, age=21]
PS D:\code2\java>
```

实验二　继承、接口与多态

1. 粘贴程序代码（可截图）。

◆ 实验一

■ Employee.java

```java
package people;
import people.People;

public class Employee extends People{
    int id;
    protected String test = "子类";
    // @Override
    //  public String getTest(){
    //      return this.test;
    //  }
    public Employee(String name,String gender,int age,int id){
        super(name,gender,age);
        this.id = id;
    }

    public void speak(){
        System.out.println("speak");
    }

    public void eat(){
        System.out.println("eat");
    }

    public void work(){
        System.out.println("work");
    }

    public int getId(){
```

```
        return this.id;
    }
    public Employee setId(int id){
        this.id = id;
        return this;
    }
}
```

- People.java

```
package people;
public abstract class People {
    protected String name,gender;
    protected int age;

    protected String test = "父类";

    public String getTest(){
        return this.test;
    }

    public People(String name,String gender,int age){
        this.name = name;
        this.gender = gender;
        this.age = age;
    }

    // 定义抽象方法
    public abstract void speak();
    public abstract void eat();

    public String getName(){
        return this.name;
    }

    public String getGender(){
        return this.gender;
    }

    public int getAge(){
        return this.age;
    }

    // 链式调用,返回this 当前对象指针
    public People setName(String name){
        this.name = name;
        return this;
```

```java
    }

    public People setGender(String gender){
        this.gender = gender;
        return this;
    }

    public People setAge(int age){
        this.age = age;
        return this;
    }
}
```

- Main.java

```java
import java.util.Scanner;

import people.Employee;


public class Main {
    public static void main(String[] args){
        Employee employee = new Employee("谭棵","男",20,202306630);
        System.out.println("这是一名员工：");
        System.out.println("姓名："+employee.getName());
        System.out.println("性别："+employee.getGender());
        System.out.println("年龄："+employee.getAge());
        System.out.println("工号："+employee.getId());
        // 父类定影的返回的是父类的指针,setName 返回的是 People 的指
针,setName 返回的是 Employee 的指针,父类没有 work 方法
        // ((Employee)employee).setName("李四").setAge(20);
        // ((Employee)employee.setName("李四").setAge(20)).work();
        //
        employee.setName("李四").setAge(20);
        employee.eat();
        employee.speak();
        employee.work();
        System.out.println();
        // People people = new People("丽丝","女",16);
        // System.out.println("这是一个普通的人：");
        // System.out.println("姓名："+people.getName());
        // System.out.println("性别："+people.getGender());
        // System.out.println("年龄："+people.getAge());
        // people.eat();
        // people.speak();
```

```
        // System.out.println(employee.getTest()); 返回的是父类的属性,因为是引用关系
    }
}
```

- 实验二采用继承的方式
  - Animal.java

```java
package animal;

public class Animal {
    // 动物的吃法都是不一样的
    public void eat(){
        System.out.println("eat");
    }

    // 动物的睡眠方法
    public void sleep() {
        System.out.println("sleep");
    }
}
```

  - Rabbit.java

```java
package animal;

public class Rabbit extends Animal {
    private String name;
    private int age;
    private String gender;

    @Override
    public void eat() {
        System.out.println("我是兔子，我吃草！");
    }
}
```

  - Tiger.java

```java
// Source code is decompiled from a .class file using FernFlower decompiler.
package animal;

public class Tiger implements Animal {
   private String name;
   private int age;
   private String gender;

   public Tiger() {
   }
```

```
    public void eat() {
        System.out.println("我是老虎，我吃肉！");
    }
}
```

- Main.java

```
import animal.Animal;
import animal.Rabbit;
import animal.Tiger;
public class Main{
    public static void main(String[] args) {
            Rabbit r = new Rabbit();
            Tiger t = new Tiger();
            r.eat();
            t.eat();
            r.sleep();
            t.sleep();
        }
}
```

◆ 实验二采用接口的方式

- 与继承类似,只是将 Animal 改为了 interface,Rabbit 和 Tigger 使用 implements 调用接口
- Animal.java

```
package animal;

public interface Animal {
    // 动物的吃法都是不一样的
    void eat();

    // 动物的睡眠方法
    default void sleep() {
        System.out.println("sleep");
    }
}
```

2. 粘贴程序的输出信息。

➢ 实验一运行结果

```
(pt2) PS D:\code\Experimental_Report\JAVA\E2_继承、接口与多态\实验1_员工> java Main
这是一名员工:
姓名: 谭棵
性别: 男
年龄: 20
工号: 202306630
eat
speak
work
```

➢ 实验二采用继承的方式

```
(pt2) PS D:\code\Experimental Report\JAVA\E2 继承、接口与多态\实验2 动物世界 继承> java Main
我是兔子，我吃草！
我是老虎，我吃肉！
sleep
sleep
```

➢ 实验二采用接口的方式

```
(pt2) PS D:\code\Experimental_Report\JAVA\E2_继承、接口与多态\实验2_动物世界> java Main
我是兔子，我吃草！
sleep
我是老虎，我吃肉！
sleep
```

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。