封装、继承

回顾

今天任务

```
1. 封装
2. private关键字
3. static
4. 继承
5. super
6. 访问权限修饰符
```

教学目标

```
1. 掌握封装性
2. 掌握static关键字
3. 掌握继承
4. 掌握super
5. 掌握访问权限修饰符
```

第一节：封装性

> 例：插线板，使用者只需要知道插上插座电器可以工作即可，至于内部线路如何连接，不需要了解

1.1 封装的概念

> 什么是封装：在类中，对于不想被类外直接访问的成员，进行私有化，同时对外提供一个共有的方法为了访问私有的成员

1.2private

> 使用private访问权限实现成员的私有化，private修饰的成员就是私有成员，只能在类内部直接访问，类外不能直接访问

1.3 get和set方法

> get方法表示访问私有属性的方法：
>
> 语法：

```
public 属性类型 getXx(){
    return 属性;
}
```

> set方法表示修改私有属性的值的方法：

```
public void setXx(参数类型 参数){
    this.xx = 参数;
}
```

第二节：static关键字

> static关键字只能修饰类成员。

2.1 静态代码块

> 代码块：单独存在的一对大括号中的代码称为代码块。代码块分为：静态代码块，动态代码块，局部代码块

> 局部代码块：声明在方法中的代码块，执行时机与声明位置相关。

> 动态代码块：又称构造代码块，声明在类体中的代码块，创建对象时自动执行一次，没创建一个对象就执行一次动态代码块。

> 静态代码块：使用static关键字修饰的动态代码块，在类加载时自动执行，并只执行一次。

2.2 静态属性

> 所有本类对象所共有且相同的一个属性，不会随着对象的改变而改变的属性。例如：圆周率

> 静态属性先于对象，不依赖于对象，可以直接通过类名直接访问。

```
public class Person{
    static int a = 5;
}
public class DemoPerson{
    public static void main(String[]args){
        System.out.println(Person.a);//Person类中的a属性是一个静态属性，可以直接通过类名访问
    }
}
```

2.3 静态方法

> 先于对象的方法，不依赖于对象，可以直接通过类名直接调用。

```
public class Person{
    public static void fun(){
        System.out.println("fun");
    }
}
public class DemoPerson{
    public static void main(String[]args){
        Person.fun();
    }
}
```

第三节：继承

3.1 继承的概念

> 在原有类的基础上，产生一个新的类，在新的类中可以访问原有类中的非私有成员，并且可以添加一些自己独有的成员，这个过程叫做继承

3.2 类的继承的使用

> 使用extends关键实现两个类的继承关系
>
> 原有类：父类，根类，超类，基类
>
> 新的类：子类

3.3 语法：

```
public class FatherClass{
        //属性
        //方法
}
public class ChildClass extends FatherClass{
        //属性
        //方法
}
```

3.4 不能被子类继承的成员：

> 1）私有成员：私有成员不能被子类继承 2）构造方法：父类中的构造方法不能被子类继承，但是会在子类的构造方法中调用（子类的构造方法中默认第一条语句是调用父类的构造方法）

3.5 继承的特点：

> 1）类单继承，一个子类只能有一个父类 2）一个父类可以有多个子类 3）子类只能继承父类中非私有的成员 4）父类的引用可以指向子类的实例,但是，父类的引用无法访问子类中独有的成员

第四节： super关键字

4.1 super关键字：用法和this类似

4.2 super的概念

> super表示父类的引用，或父类

4.3 super的使用规则

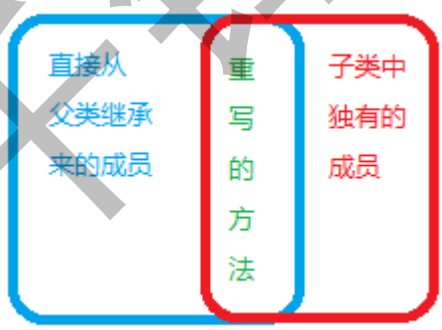
> 1）super.属性：表示访问父类中的属性，当子类中定义了与父类同名的属性时，若想在子类中访问父类的同名属性，需要使用super.属性访问
>
> 2）super.方法：表示调用父类中的方法，在子类中需要调用父类中没有被重写的方法时，需要使用super.方法调用
>
> 3）super()：表示调用父类的构造方法，注意：super()必须是子类构造方法中第一条语句 子类中构造方法默认第一条语句会调用父类的无参数构造方法super()，也可以手动调用父类中带参数的构造方法

4.4 练习

```java
public class Animal{
    int a = 10;
    public String name;
    public int age;
    public Animal(){}
    public Animal(String name, int age){
        this.name = name;
        this.age = age;
    }
    public void fun(){
        System.out.println("Animal类的fun方法");
    }
}
public class Dog extends Animal{
    int a = 20;
    public Dog(){
        super();//调用父类的构造方法
    }
    @Override
    public void fun(){
        System.out.println(a);//输出结果为20
        System.out.println(super.a);//输出结果为10
        super.fun();//调用父类中没有被重写过的方法
    }
}
```

4.5 继承中的构造方法

> 父类中的构造方法不会被子类继承，但是创建子类对象时，会调用父类的构造方法。也就是说，创建子类对象时，需要先创建一个父类的对象，在父类对象实例的内存的基础上附加一块子类独有成员的内存整体是一个子类对象。



第五节： 方法重写

5.1 什么是方法的重写：

> 在继承过程中，子类中从父类继承来的方法无法满足自己的需求时，可以在子类中对方法进行完善，这 个过程叫做方法重写，方法的重写相当于在子类中覆盖父类中的方法

5.2 方法重写的要求

> 从方法的签名部分入手
>
> 访问权限 其他修饰符 返回值 方法名(参数列表)
>
> 重写方法的过程中，子类中的方法访问权限，大于等于父类中被重写方法
>
> 其他修饰，相对随意
>
> 返回值，相同（类型相兼容）
>
> 方法名必须完全一致
>
> 参数列表相同

5.3 方法的重写和方法的重载的区别

> 方法的重载：Overload，在同一个类中一组方法名形同，参数个数或类型不同的方法，互为重载方法
>
> 方法的重写：Override，在继承过程中，在类中完善父类中继承来的方法
>
> 方法重写是：@Override注解表示验证重写方法的格式是否正确

5.4 有关方法重写之后的调用：

> 只要在子类中重写过的方法，创建出的对象只要有关子类都会执行重写后的方法
>
> 也就是说对象在调用方法时，实例决定执行的方法，与引用无关

第六节：访问权限

	本类	同包中类或同包子类	不同包子类	不同包类
public	v	v	v	v
protected	v	v	v	x
默认	v	v	x	x
private	v	x	x	x

```java
package a;
public class Person{
    public String name;
    protected int age;
    char sex;
    private double sal;
    public Person(){}
    public Person(String name, int age, char sex, double sal){
        this.name = name;
        this.age = age;
        this.sex = sex;
        this.sal = sal;
    }
    public static void main(String[]args){
        Person p = new Person("张三", 12, 'm', 5000);
        System.out.println(p.name);
        System.out.println(p.age);
        System.out.println(p.sex);
        System.out.println(p.sal);
    }
}
```

```java
package a;
public class Student extends Person{
    public static void main(String[]args){
        Person p = new Person("张三", 12, 'm', 5000);
        System.out.println(p.name);
        System.out.println(p.age);
        System.out.println(p.sex);
        //System.out.println(p.sal);//同包子类中无法访问父类中私有成员
    }
}

package a;
public class Demo{
    public static void main(String[]args){
        Person p = new Person("张三", 12, 'm', 5000);
        System.out.println(p.name);
        System.out.println(p.age);
        System.out.println(p.sex);
        //System.out.println(p.sal);//同包类中无法访问父类中私有成员
    }
}
```

```java
package b;
public class Student extends Person{
    public static void main(String[]args){
        Person p = new Person("张三", 12, 'm', 5000);
        System.out.println(p.name);
        System.out.println(p.age);
        //System.out.println(p.sex);//不同包中子类中无法访问默认权限成员
        //System.out.println(p.sal);
    }
}

package b;
public class Demo{
    public static void main(String[]args){
        Person p = new Person("张三", 12, 'm', 5000);
        System.out.println(p.name);
        //System.out.println(p.age);//不同包中不能访问受保护属性
        //System.out.println(p.sex);
        //System.out.println(p.sal);//不同包类中无法访问父类中私有成员
    }
}
```

课前默写

> 设计一个矩形类（Rectangle），有长，宽属性，计算周长功能，计算面积的功能
>
> 设计一个四棱住类（Column），有底面矩形、高属性，计算表面积，计算体积

作业

> 1.写一个工具类，类中定义静态方法，实现数组的降序排序（选择和冒泡分别写一个类）
>
> 2.写一个工具类，类中定义静态方法，实现二分法查找
>
> 3.使用单例设计模式，设计一个阿里巴巴董事局主席类，然后，测试。
>
> 要求类的对象有一些属性（名字，性别、身高、薪水） 模拟一个场景：阿里巴巴主管外部事务的高管给来访的外宾介绍本公司首脑的情形。
>
> 4.做一把椅子的过程 两块铁皮Iron 磨成一把刀， 刀 Knife可以砍树 Tree， 砍成木材 Meterial， 木材做成椅子Chair
>
> 要求：设计一个工具类，铁皮磨成到、木材做成椅子的方法在工具类中实现

面试题

> 方法重载与方法重写的区别