1.面向对象的四大特征

封装、继承、多态、抽象。

2.封装

2.1什么是封装

封装就是将设计者和使用者分开。其目的是保护数据。

2.2如何实现封装

通过 private 关键字对属性和方法进行封装，封装后的属性和方法其他类中无法访问。只能在当前类中访问。

```java
package com.iweb.lesson01;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 08:54
 * 描述: Person
 */
public class Person {

    // private 进行封装，封装后只能当前中可以访问
    private String name; // 姓名
    private int age;  // 年龄

    private void show() {
        System.out.println(name + "\t" + age);
    }

}
```

```java
package com.iweb.lesson01;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 08:54
 * 描述: Test
 */
public class Test {

    public static void main(String[] args) {

        Person person = new Person();
        // person.name = "jack"; // 安全性
        // System.out.println(person.name);


```

```
        // person.show();

    }

}
```

被封装的数据外部不能直接访问，如果需要访问，则必须提供访问的接口。

```
package com.iweb.lesson02;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 08:59
 * 描述: Phone
 */
public class Phone {

    private String model; // 型号
    private int memory; // 内存


    public Phone(String model, int memory) {
        this.model = model;
        this.memory = memory;
    }

    // 查看手机的型号
    public String getModel() {
        return model;
    }

    // 查看手机的内存
    public int getMemory() {
        return memory;
    }

    // 拍照
    public void camera() {
        System.out.println("拍照");
    }

    // 打开应用
    public void openApp() {
        memory -= 1;
    }

}
```

```
package com.iweb.lesson02;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 08:59
 * 描述: Test
 */
```

```java
public class Test {

    public static void main(String[] args) {

        Phone phone = new Phone("华为", 8);

        // 查看手机型号
        System.out.println(phone.getModel());
        System.out.println(phone.getMemory());
        // 通过手机进行拍照
        phone.camera();

        // 打开APP
        phone.openApp();
        // 查看剩余内存
        System.out.println(phone.getMemory());

    }

}
```

对数据进行操作无非2中操作： 设置和查看。

设置数据： 构造函数重载和 set方法

查看数据： get方法

```java
package com.iweb.lesson03;

import java.util.concurrent.Phaser;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:10
 * 描述: Phone
 */
public class Phone {

    private String model = "华为M30"; // 型号      华为M30 华为P30
    private int memory = 4; // 内存    4 8

    public Phone() {

    }

    /*构造函数*/
    public Phone(String model) {
        this.model = model;
    }

    public Phone(int memory) {
        this.memory = memory;
    }

```

```
    public Phone(String model, int memory) {
        this.model = model;
        this.memory = memory;
    }

    /*set方式*/
    public void setModel(String model) {
        this.model = model;
    }

    public void setMemory(int memory) {
        this.memory = memory;
    }

    /*GET方法*/
    public String getModel() {
        return model;
    }

    public int getMemory() {
        return memory;
    }
}
```

```
package com.iweb.lesson03;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:10
 * 描述: Test
 */
public class Test {
    public static void main(String[] args) {

        Phone p1 = new Phone();
        System.out.println(p1.getModel() + "\t" + p1.getMemory());

        Phone p2 = new Phone(8);
        System.out.println(p2.getModel() + "\t" + p2.getMemory());

        Phone p3 = new Phone("P30", 8);
        System.out.println(p3.getModel() + "\t" + p3.getMemory());


        Phone p4 = new Phone();
        p4.setModel("P40");
        p4.setMemory(8);
        System.out.println(p4.getModel() + "\t" + p4.getMemory());

    }
}
```

私有的使用场景： 充电是暴露给用户的接口，而由充电的时间来计算电池的使用时间是手机内部的方法，用户不能直接调用。

```java
/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:10
 * 描述: Phone
 */
public class Phone {

    private int time; // 电池使用时间

    private int cTime;

    // 计算使用时间
    private void setTime() {
        time = cTime * 24;
    }

    // 用户充电
    public void chongDian(int cTime) {
        this.cTime = cTime;
    }

    // 查看剩余使用时间
    public int getTime() {
        return time;
    }
}
```

PS: 在编写方法的时候，一个方法的有效代码不能超出 30行。否则就需要进行方法的封装。一个方法确保只做一件事情。

3.继承

3.1什么继承

继承是类于类之间的关系，一个类继承另外一个类，被继承的类叫做父类、超类、基类，继承的类叫做子类、派生类。

子类继承了父类的所有属性和方法。包含私有的人、私有的不能直接访问。

```java
package com.iweb.lesson05;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:35
 * 描述: F : 父类
 */
public class F {

    private String word = "F"; // 私有的子类不能直接访问
    int number = 100; // 子类可以访问

    // 子类可以访问
    public void show() {
        System.out.println("F is show");
    }
```

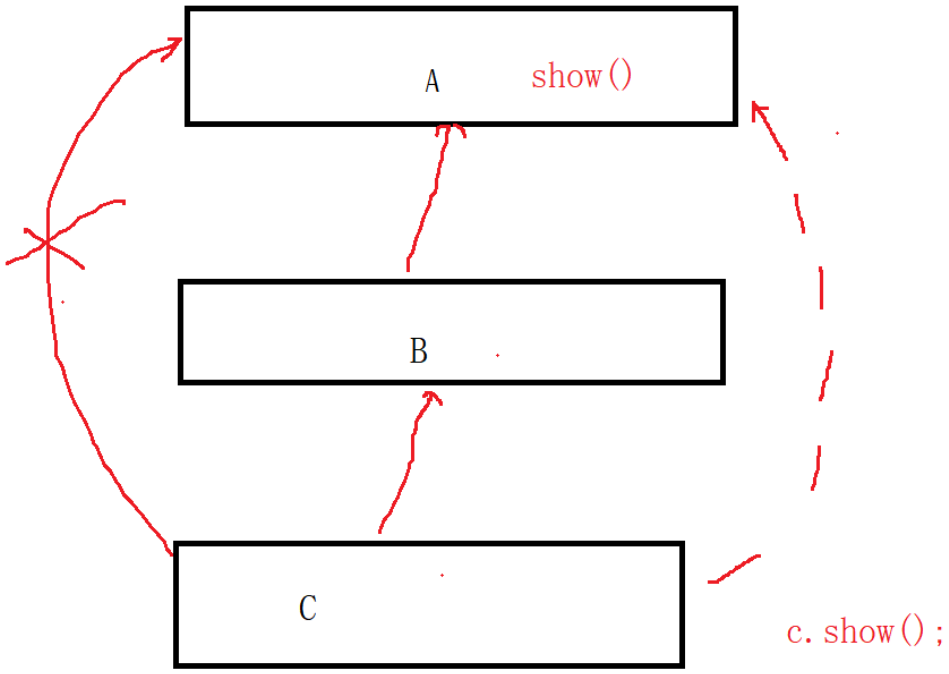
```
17
18 }
```

```
1  package com.iweb.lesson05;
2
3  /**
4   * 作者: jack
5   * 时间: 2021-04-21 0021 09:35
6   * 描述: S : 子类
7   */
8  public class S extends F {
9
10
11 }
```

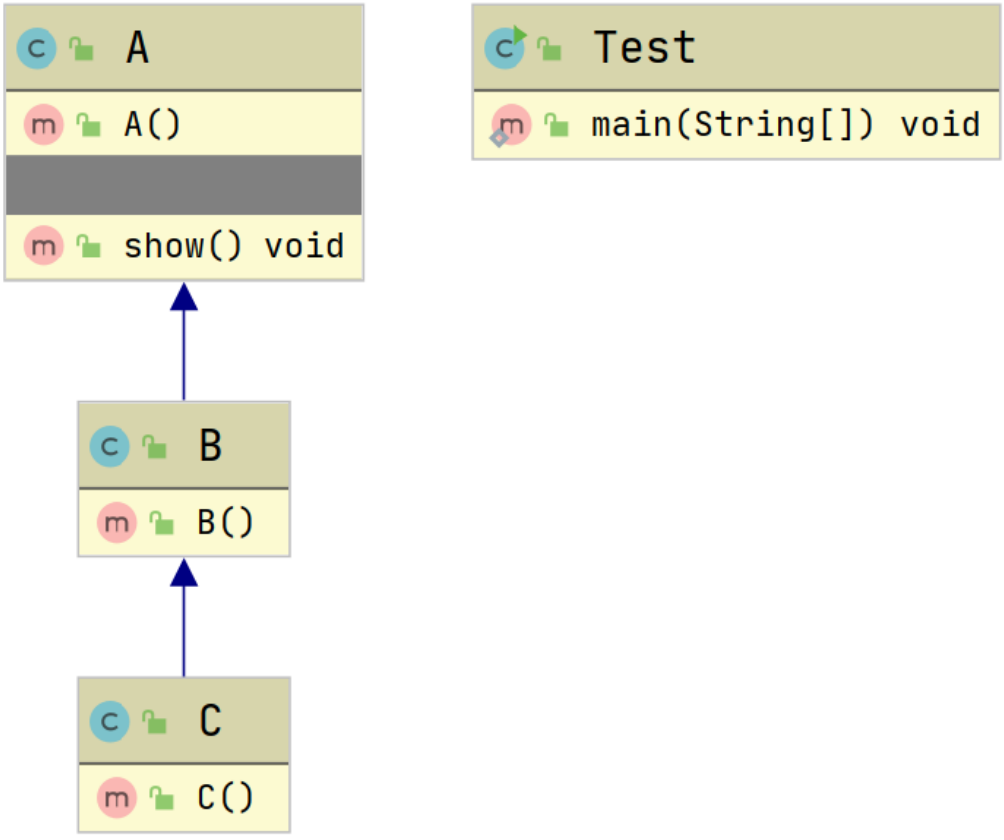
```
1  package com.iweb.lesson05;
2
3  /**
4   * 作者: jack
5   * 时间: 2021-04-21 0021 09:35
6   * 描述: Test
7   */
8  public class Test {
9
10     public static void main(String[] args) {
11
12         S s = new S();
13         System.out.println(s.number);
14
15         s.show();
16
17     }
18
19 }
```

3.2继承的特征

1. 子类继承了父类的所有属性和方法。包含私有的、私有的不能直接访问。
2. 继承只能是单继承，不能是多继承。可以间接继承。
3. 要想创建子类对象必先创建父类对象。



说明：C继承了B，B继承了A，C就间接继承了A, 可以直接使用A中的show().



3.3super关键字

super：

1. 在构造函数中可以调用父类的构造函数，必须是第一行。
2. 在方法中调用父类方法

在构造函数中的使用

```java
package com.iweb.lesson06;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:43
 * 描述: A
 */
public class A {

    public A() {
        System.out.println("A is init");
    }

    public A(String name) {
        System.out.println("A(String name)");
    }

    public void show() {
        System.out.println("A is show");
    }

}
```

```java
package com.iweb.lesson06;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:43
 * 描述: B
 */
public class B extends A {
    public B() {
        super("jack"); // A 的构造函数: 可以通过参数列表决定调用哪个构造函数
        System.out.println("B is init");
    }
}
```

在方法中使用：没有什么大用只是一个语法而已

使用场景： 当子类中重写了父类中的方法，而子类中又需要调用这个父类的方法的时候

```java
package com.iweb.lesson06;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:43
 * 描述: A
 */
public class A {

    public A() {
        System.out.println("A is init");
    }

```

```
    public A(String name) {
        System.out.println("A(String name)");
    }

    public void show() {
        System.out.println("A is show");
    }

}
```

```
package com.iweb.lesson06;

/**
 * 作者: jack
 * 时间: 2021-04-21 0021 09:43
 * 描述: B
 */
public class B extends A {
    public B() {
        super("jack"); // A 的构造函数: 可以通过参数列表决定调用哪个构造函数
        System.out.println("B is init");
    }


    public void eat() {
        super.show();
    }

    // 方法重写
    @Override
    public void show() {
        // 一些业务
        super.show();
        // 有一些业务
    }
}
```

4.抽象

5.多态