

38从零开始学Java38之面向对象三大特征之封装

前言

配套开源项目资料

一. 三大特征

二. 封装

1. 概念

2. 必要性

3. 实现要求

3.1 private关键词

3.2 getter和setter方法

4. 封装案例

三. 结语

四. 配套视频

五. 今日作业

1. 第一题

作者：孙玉昌，昵称【**一一哥**】，另外【**壹壹哥**】也是我哦

千锋教育高级教研员、CSDN博客专家、万粉博主、阿里云专家博主、掘金优质作者

前言

我们知道，Java是面向对象的编程语言。关于面向对象的概念，壹哥在之前的文章中就已经给大家详细地讲过了。但是关于面向对象，我们还有另一块不得不学的内容，那就是面向对象的特征！

可以说，面向对象的三大特征，是我们每个java程序员都必须掌握的基本功。在日常的开发中，每段java代码的编写实现，都应该体现出这些特征的基本要求。另外在我们面试的时候，面向对象的特征更是初级程序员的必考要点，面试官动不动就来一句：你给我说说面向对象的特征有哪些吧。

全文大约【2500】字，不说废话，只讲可以让你学到技术、明白原理的纯干货！本文带有丰富的案例及配图视频，让你更好地理解 and 运用文中的技术概念，并可以给你带来具有足够启迪的思考.....

配套开源项目资料

Github:

d/LearnJava

GitHub – SunLtd/LearnJava

Contribute to SunLtd/LearnJava development by creating an account on GitHub.

GitHub

Gitee:



——哥/从零开始学Java

从零开始学Java系列 稀土掘金专栏地址: <https://juejin.cn/column/7175082165548351546> CSDN专...

Gitee

一. 三大特征

我们知道，面向对象是一种以对象为核心的软件开发思想。在这种编程思想中，有着明显的几个特征，通常我们归纳为如下三个：

- **封装(Encapsulation)**：将类或接口的实现细节部分包装、隐藏起来。
- **继承(extends)**：子类继承父类的特征和行为，使得子类对象(实例)拥有父类的实例域和方法；或子类从父类继承方法，使得子类拥有父类相同的行为。
- **多态(polymorphism)**：同一个行为具有多个不同表现形式或形态的能力，即同一个接口可以使用不同的实例执行不同的操作。

但也有的面试官会问我们面向对象有哪四大特征？这就让你很懵逼，不都是三大特征吗？如果非要有这种傻缺的问法，那我们就可以再多回答一个**“抽象”**的概念即可。

关于继承、多态和抽象等的概念，壹哥会在后面的文章中单独讲解，本篇文章先只讲封装是怎么回事。

二. 封装

1. 概念

所谓的封装，其实就是**尽可能地隐藏对象的内部实现细节，控制对象的修改及访问权限，将某些实现细节部分包装、隐藏起来，并对外提供公开的方法访问私有成员。一般我们是利用private访问修饰符来实现。**

通俗地说，封装就是把一个Java类的部分实现进行“隐藏”、“包裹”，不让你完全看见它的“内部”是咋回事。就好比一个银行，有的地方(大厅)可以对外暴露，但有的地方(金库)就需要隐藏。我们在修建银行的时候，金库在设计之初就是要被隐藏的，但金库会预留一个渠道对外联通，让钱可以进出。

Java里的封装其实也是这样的。把一个类中不想让别人直接看到的内容封装隐藏起来，不让别人直接操作这部分内容，但又可以通过某种方法对外展示。这样，我们对java类的操作就可进可退，保证了类的安全性。

所以封装可以被认为是一种类的保护屏障，防止该类的代码和数据被外部类定义的代码随机访问。如果想要访问该类的代码和数据，必须通过严格的接口控制。通过适当的封装，让程式码更容易理解与维护，也加强了代码的安全性。

2. 必要性

那么Java里的类一定都要封装吗？我们来看一个场景，如下图所示：

```
public class TestEncapsulation {
    public static void main(String[] args) {
        Student s1 = new Student();
        s1.name = "tom";
        s1.age = 20000;
        s1.sex = "male";
        s1.score = 1000;
    }
}

class Student{
    String name;
    int age;
    String sex;
    double score;
}
```

在对象的外部，为对象的属性赋值，可能存在非法数据的录入。

就目前的技术而言，并没有办法对属性的赋值加以控制。

千锋教育-孙玉昌

比如上面的Student类，内部有4个非私有的属性。我们可以在对象的外部，为这几个属性直接赋值。在上面的代码中，我们给age赋值20000，学生的年龄是20000，这合理吗？得分是1000合理吗？这就出现了非法的数据录入，存在数据的安全问题。这种问题该怎么解决？如何对属性的赋值加以控制？

针对这种情况，我们就可以使用封装！将类内部的属性进行私有化，然后再提供对外的方法调用这些属性，并在方法内部对属性的赋值进行合理控制。**别的地方只能通过规定的方法来访问数据，隐藏了类的实现细节，方便修改和实现，这样我们就不担心非法数据的录入问题了！**这也就是封装的意义所在！

既然封装有其存在的必要性，那么我们该如何实现封装呢？请继续往下看！

3. 实现要求

如果我们想实现类的封装，主要有以下两个步骤：

1. 给成员变量添加private修饰符；
2. 添加公开的get和set方法；
3. 在赋值和取值方法中，加入属性控制语句，对属性值的合法性进行判断。

3.1 private关键词

private是一种访问权限修饰符，是一个表示“私有的”关键词，可以修饰类、方法和变量等。如果用来修饰成员变量，可以实现成员变量的私有化，限制该成员变量只能在类的内部直接访问，类外不能直接访问。

3.2 getter和setter方法

通常情况下，**这些get和set方法被称为getter和setter方法**。任何访问类中私有成员变量的操作都要通过这些getter和setter方法。

get方法是用来获取私有属性的方法，其语法结构为：

```
1 public 返回值类型 getXxx(){
2
3     return 返回属性值;
4 }
```

set方法是用来修改私有属性值的方法，其语法结构为：

```
1 public void setXxx(属性类型 属性){
2     this.xxx = 属性;
3     //其他对属性进行操作的代码
4 }
```

这里我们要注意，**get方法一般没有参数，但有返回值；set方法一般没有返回值，但有参数！！**

4. 封装案例

为了让大家更好地理解如何实现封装，壹哥设计一个Person类，我们来看看代码：

```
1  /**
2   * @author 一一哥Sun
3   * QQ: 2312119590
4   * CSDN、掘金、知乎找我哦
5   * 属性:姓名、年龄、性别
6   * 封装步骤:(1).把成员变量变成私有的;(2).添加公开的get和set方法
7   */
8  public class Person {
9
10     // 私有化属性
11     private String name;
12     private int age;
13     private String sex;
14
15     //设置公开方法
16     // 获取name
17     public String getName() {
18         return name;
19     }
20
21     // 对name赋值
22     public void setName(String name) {
23         //this关键字解决了实例变量(private String name)和局部变量(String name)
        之间的同名冲突。
24         this.name = name;
25     }
26
27     public int getAge() {
28         return this.age;
29     }
30
31     public void setAge(int age) {
32         // 限制年龄的范围
33         if (age > 100 || age < 1) {
34             // 设置一个默认值
35             this.age = 18;
36         } else {
37             this.age = age;
38         }
39     }
40
41     public String getSex() {
42         return this.sex;
43     }
44 }
```

```

45     public void setSex(String sex) {
46         // 设置性别
47         if ("男".equals(sex) || "女".equals(sex)) {
48             this.sex = sex;
49         } else {
50             this.sex = "男";
51         }
52     }
53
54     //提供无参和有参的构造方法
55     public Person() {
56         super();
57     }
58
59     public Person(String name, int age, String sex) {
60         super();
61         this.name = name;
62         this.age = age;
63         this.sex = sex;
64     }
65
66 }

```

从上述代码中，我们可以看出实现类的封装其实也很简单：

1. 成员变量(属性)私有化(用private修饰)；
2. 添加公开的get和set方法，用public修饰；
3. 添加有参和无参的若干构造方法。

一般我们把这样封装好的一个类成为Java Bean，Java Bean一般有两种：

1. 负责数据承载的Bean；
2. 负责业务逻辑的Bean。

我们上面的案例，就是一个数据承载的Bean。

-----正片已结束，来根事后烟-----

三. 结语

至此，我们就把面向对象的第一个基本特征——封装 学习完了。现在你知道该如何正确地封装一个Java Bean了吗？在面试时，我们要把封装的概念及其优点讲清楚：

- 良好的封装能够减少耦合；
- 封装可以使得类内部的结构自由修改；
- 封装可以对成员变量进行更精确的控制；
- 封装隐藏了类的信息和实现细节。

如果你独自学习觉得有很多困难，可以加入壹哥的学习互助群，大家一起交流学习。

四. 配套视频

如果你不习惯阅读技术文章，或是对文中的技术概念不能很好地理解，可以来看看壹哥帮你筛选出的视频教程。[与本文配套的Java学习视频，链接如下：](#)

<https://player.bilibili.com/player.html?bvid=BV1Ja411x7XB&p=78&page=78>

五. 今日作业

1. 第一题

封装一个部门类，设计合理的属性及访问方法。