链滴

关于封装、继承、多态的学习心得

作者：aopstudio

原文链接：https://ld246.com/article/1584190841657

来源网站：链滴

<h2 id="前言">前言</h2>
封装、继承、多态是面向对象语言的三大特性。笔者主修的是 Java 语言，在刚学 Java 的时候老就反复告诉我们这三个词。不过之前一直也只是知道个大概。最近又仔细看了一下网上的一些解析，定记录一下自己的学习心得。
<h2 id="开门见山--三者的关系">开门见山——三者的关系</h2>
我认为这三者并不是三个平行的概念，而是一个层层递进的关系，即封装最基本的，有了封装以后才能实现继承，在封装和继承的基础上才能实现多态。
<h2 id="一切的基石--封装">一切的基石——封装</h2>
封装就是把一类数据和与这类数据的相关操作封装起来，数据不能随意被访问和改动，只能通过开的接口来进行增删改查。Java 里面为了实现封装，设置了四个访问修饰符，分别是：

public：public 修饰的成员变量或方法，项目中的任何类都可以访问
protected：只有同个包内的类和自己的子类能访问
friendly（不写）：只有同个包内的类能访问，也就是说就算有个子类，但它在其他的包中，也能进行访问
private：只有自己这个类能访问

一般在实践中，对于所有成员变量都会有用 private 进行修饰，同时设置公开的 get 和 set 方法访问和修改成员变量的值。除此之外，再加上提供默认构造方法和实现序列化接口，一个类就可以称为 Java Bean。这正是封装思想的体现。
<h2 id="实现代码复用和扩展--继承">实现代码复用和扩展——继承</h2>
封装了一个类后，发现它和很多其他的类都有共性。而我们如果单独地去定义每一个类，那么一相同的属性和方法每次都要重写一遍，太麻烦了。我们可以把它们的共性重新封装成一个基类，对于个特定的类用一个子类去继承这个基类，在子类中可以扩展出它们独有的属性和方法，也可以对基类的方法进行重写以更加适应子类的实际应用。
关于继承，我认为可以分为广义的继承和狭义的继承。狭义的继承指的是对于类的继承，广义的承包含了对于类的继承和对于接口的继承。对于接口的继承正式的说法应该叫做实现。在 Java 语言这两种概念的关键字是不同的。继承类的关键字是 extends，实现接口的关键字是 implements。类调的是所属关系，就像人只能有一个亲爸爸，在 Java 中类也只能继承一个父类。接口强调的是行为范，比如说你会哪些本领，而你的本领可以是好多个老师教的，Java 中的接口指的也是在逻辑上它能行的操作，因此一个 Java 类可以实现好多个接口。
<h2 id="多态">多态</h2>
多态的三个前提条件是：

继承（广义上的继承）
子类对父类（或接口）方法进行了重写
向上转型，即父类引用指向子类对象

多态的前提条件就有继承，所以如果没有继承多态也就无从说起。多态的意义在于：把不同的子对象都当作父类来看，可以屏蔽不同子类对象之间的差异，写出通用的代码，做出通用的编程，以适需求的不断变化。 赋值之后，父类型的引用就可以根据当前赋值给它的子对象的特性以不同的方式运。它提高了代码的可维护性和扩展性。
Java Collection 接口是一个极好的展示多态应用的例子。Collection 接口的关系示意图如下：<mg src="https://ld246.com/images/img-loading.svg" alt="Collection.png" data-src="https://3logfile.com/file/2020/03/Collection-3afd4bb0.png?imageView2/2/interlace/1/format/jpg">
这里标了 <<interface>> 的都是接口，没有标的都是类。这张图里还有一个特别的方，就是 LinkedList 同时实现了两个接口，也就是说它既可以当 List 使用，又可以当 Queue 使用。我们了解了多态以后，如果想要使用 LinkedList，我们就要根据具体用途来定义。
如果我们要把 LinkedList 当作适合频繁插入删除节点的线性表来使用，我们就应该写成
<pre><code class="language-java highlight-chroma">List l</spn>=new LinkedList();

</code></pre>
这时候开发者就只能够调用 LinkedList 实现的 List 接口中的方法，而不能使用它实现的 Queue 接口中的方法。
如果我们的需求变更了，由原本的频繁插入删除节点变成了频繁随机访问节点，那么我们只需要 LinkedList 改成 ArrayList 就行，其他地方都不用变。
而如果我们想要把 LinkedList 当作先入先出的队列来使用，我们就应该写成
<pre><code class="language-java highlight-chroma">Queue q=new LinkedList();
</code></pre>
如果我们的需求变动为需要对队列中的节点根据优先级进行排序，我们只需要将 LinkedList 修为 PriorityQueue 即可，其他地方都不用变。