

04 | 理论一：当谈论面向对象的时候，我们到底在谈论什么？

王争 · 设计模式之美



考虑到各个水平层次的同学，并且保证专栏内容的系统性、全面性，我会循序渐进地讲解跟设计模式相关的所有内容。所以，专栏正文的第一个模块，我会讲一些设计原则、设计思想，比如，面向对象设计思想、经典设计原则以及重构相关的知识，为之后学习设计模式做铺垫。

在第一个模块中，我们又首先会讲到面向对象相关的理论知识。提到面向对象，我相信很多人都不陌生，随口都可以说出面向对象的四大特性：封装、抽象、继承、多态。实际上，面向对象这个概念包含的内容还不止这些。所以，今天我打算花一节课的时间，先大概跟你聊一下，当我们谈论面向对象的时候，经常会谈到的一些概念和知识点，为学习后面的几节更加细化的内容做一个铺垫。

特别说明一下，对于今天讲到的概念和知识点，大部分我都是点到为止，并没有展开详细讲解。如果你看了之后，对某个概念和知识点还不是很清楚，那也没有关系。在后面的几节课中，我会花更多的篇幅，对今天讲到的每个概念和知识点，结合具体的例子，一一做详细的讲解。

什么是面向对象编程和面向对象编程语言？

面向对象编程的英文缩写是 OOP，全称是 Object Oriented Programming。对应地，面向对象编程语言的英文缩写是 OOPL，全称是 Object Oriented Programming Language。

面向对象编程中有两个非常重要、非常基础的概念，那就是类（class）和对象（object）。这两个概念最早出现在 1960 年，在 Simula 这种编程语言中第一次使用。而面向对象编程这个概念第一次被使用是在 Smalltalk 这种编程语言中。Smalltalk 被认为是第一个真正意义上的面向对象编程语言。

1980 年左右，C++ 的出现，带动了面向对象编程的流行，也使得面向对象编程被越来越多的人认可。直到今天，如果不按照严格的定义来说，大部分编程语言都是面向对象编程语言，比如 Java、C++、Go、Python、C#、Ruby、JavaScript、Objective-C、Scala、PHP、Perl 等等。除此之外，大部分程序员在开发项目的时候，都是基于面向对象编程语言进行的面向对象编程。

以上是面向对象编程的大概发展历史。在刚刚的描述中，我着重提到了两个概念，面向对象编程和面向对象编程语言。那究竟什么是面向对象编程？什么语言才算是面向对象编程语言呢？如果非得给出一个定义的话，我觉得可以用下面两句话来概括。

面向对象编程是一种编程范式或编程风格。它以类或对象作为组织代码的基本单元，并将封装、抽象、继承、多态四个特性，作为代码设计和实现的基石。

面向对象编程语言是支持类或对象的语法机制，并有现成的语法机制，能方便地实现面向对象编程四大特性（封装、抽象、继承、多态）的编程语言。

一般来讲，面向对象编程都是通过使用面向对象编程语言来进行的，但是，不用面向对象编程语言，我们照样可以进行面向对象编程。反过来讲，即便我们使用面向对象编程语言，写出来的代码也不一定是面向对象编程风格的，也有可能是面向过程编程风格的。这里听起来是不是有点绕？不过没关系，我们在后面的第 7 节课中，会详细讲解这个问题。

除此之外，从定义中，我们还可以发现，理解面向对象编程及面向对象编程语言两个概念，其中最关键的一点就是理解面向对象编程的四大特性。这四大特性分别是：封装、抽象、继承、

多态。不过，关于面向对象编程的特性，也有另外一种说法，那就是只包含三大特性：封装、继承、多态，不包含抽象。为什么会有这种分歧呢？抽象为什么可以排除在面向对象编程特性之外呢？关于这个问题，在下一节课详细讲解这四大特性的时候，我还会再拿出来说一下。不过，话说回来，实际上，我们没必要纠结到底是四大特性还是三大特性，关键还是理解每种特性讲的是什么内容、存在的意义以及能解决什么问题。

而且，在技术圈里，封装、抽象、继承、多态也并不是固定地被叫作“四大特性”（features），也有人称它们为面向对象编程的四大概念（concepts）、四大基石（cornerstones）、四大基础（fundamentals）、四大支柱（pillars）等等。你也发现了吧，叫法挺混乱的。不过，叫什么并不重要。我们只需要知道，这是前人进行面向对象编程过程中总结出来的、能让我们更容易地实现各种设计思路的几个编程套路，这就够了。在之后的课程讲解中，我统一把它们叫作“四大特性”。

如何判定某编程语言是否是面向对象编程语言？

如果你足够细心，你可能已经留意到，在我刚刚的讲解中，我提到，“如果不按照严格的定义来说，大部分编程语言都是面向对象编程语言”。为什么要加上“如果不按照严格的定义”这个前提呢？那是因为，如果按照刚刚我们给出的严格的面向对象编程语言的定义，前面提到的有些编程语言，并不是严格意义上的面向对象编程语言，比如 JavaScript，它不支持封装和继承特性，按照严格的定义，它不算是面向对象编程语言，但在某种意义上，它又可以算得上是一种面向对象编程语言。我为什么这么说呢？到底该如何判断一个编程语言是否是面向对象编程语言呢？

还记得我们前面给出的面向对象编程及面向对象编程语言的定义吗？如果忘记了，你可以先翻到上面回顾一下。不过，我必须坦诚告诉你，那个定义是我自己给出的。实际上，对于什么是面向对象编程、什么是面向对象编程语言，并没有一个官方的、统一的定义。而且，从 1960 年，也就是 60 年前面向对象编程诞生开始，这两个概念就在不停地演化，所以，也无法给出一个明确的定义，也没有必要给出一个明确定义。

实际上，面向对象编程从字面上，按照最简单、最原始的方式来理解，就是将对象或类作为代码组织的基本单元，来进行编程的一种编程范式或者编程风格，并不一定需要封装、抽象、继承、多态这四大特性的支持。但是，在进行面向对象编程的过程中，人们不停地总结发现，有了这四大特性，我们就能更容易地实现各种面向对象的代码设计思路。

比如，我们在面向对象编程的过程中，经常会遇到 is-a 这种类关系（比如狗是一种动物），而继承这个特性就能很好地支持这种 is-a 的代码设计思路，并且解决代码复用的问题，所以，继承就成了面向对象编程的四大特性之一。但是随着编程语言的不迭代、演化，人们发现继承这种特性容易造成层次不清、代码混乱，所以，很多编程语言在设计的时候就开始摒弃继承特性，比如 Go 语言。但是，我们并不能因为它摒弃了继承特性，就一刀切地认为它不是面向对象编程语言了。

实际上，我个人觉得，只要某种编程语言支持类或对象的语法概念，并且以此作为组织代码的基本单元，那就可以被粗略地认为它就是面向对象编程语言了。至于是否有现成的语法机制，完全地支持了面向对象编程的四大特性、是否对四大特性有所取舍和优化，可以不作为判定的标准。基于此，我们才有了前面的说法，**按照严格的定义，很多语言都不能算得上面向对象编程语言，但按照不严格的定义来讲，现在流行的大部分编程语言都是面向对象编程语言。**

所以，多说一句，关于这个问题，我们一定不要过于学院派，非要给面向对象编程、面向对象编程语言下个死定义，非得对某种语言是否是面向对象编程语言争个一清二白，这样做意义不大。

什么是面向对象分析和面向对象设计？

前面我们讲了面向对象编程（OOP），实际上，跟面向对象编程经常放到一块儿来讲的还有另外两个概念，那就是面向对象分析（OOA）和面向对象设计（OOD）。面向对象分析英文缩写是 OOA，全称是 Object Oriented Analysis；面向对象设计的英文缩写是 OOD，全称是 Object Oriented Design。OOA、OOD、OOP 三个连在一起就是面向对象分析、设计、编程（实现），正好是面向对象软件开发要经历的三个阶段。

关于什么是面向对象编程，我们前面已经讲过了。我们现在再来讲一下，什么是面向对象分析和设计。这两个概念相对来说要简单一些。面向对象分析与设计中的“分析”和“设计”这两个词，我们完全可以从字面上去理解，不需要过度解读，简单类比软件开发中的需求分析、系统设计即可。不过，你可能会说，那为啥前面还加了个修饰词“面向对象”呢？有什么特殊的意义吗？

之所以在前面加“面向对象”这几个字，是因为我们是围绕着对象或类来做需求分析和设计的。分析和设计两个阶段最终的产出是类的设计，包括程序被拆解为哪些类，每个类有哪些属性方

法，类与类之间如何交互等等。它们比其他的分析和设计更加具体、更加落地、更加贴近编码，更能够顺利地过渡到面向对象编程环节。这也是面向对象分析和设计，与其他分析和设计最大的不同点。

看到这里，你可能会问，那面向对象分析、设计、编程到底都负责做哪些工作呢？简单点讲，面向对象分析就是要搞清楚做什么，面向对象设计就是要搞清楚怎么做，面向对象编程就是将分析和设计的结果翻译成代码的过程。今天，我们只是简单介绍一下概念，不展开详细讲解。在后面的面向对象实战环节中，我会用两节课的时间，通过一个实际例子，详细讲解如何进行面向对象分析、设计和编程。

什么是 UML？我们是否需要 UML？

讲到面向对象分析、设计、编程，我们就不得不提到另外一个概念，那就是 UML（Unified Model Language），统一建模语言。很多讲解面向对象或设计模式的书籍，常用它来画图表达面向对象或设计模式的设计思路。

实际上，UML 是一种非常复杂的东西。它不仅仅包含我们常提到类图，还有用例图、顺序图、活动图、状态图、组件图等。在我看来，即便仅仅使用类图，学习成本也是很高的。就单说类之间的关系，UML 就定义了很多种，比如泛化、实现、关联、聚合、组合、依赖等。

要想完全掌握，并且熟练运用这些类之间的关系，来画 UML 类图，肯定要花很多的学习精力。而且，UML 作为一种沟通工具，即便你能完全按照 UML 规范来画类图，可对于不熟悉的人来说，看懂的成本也还是很高的。

所以，从我的开发经验来说，UML 在互联网公司的项目开发中，用处可能并不大。为了文档化软件设计或者方便讨论软件设计，大部分情况下，我们随手画个不那么规范的草图，能够达意，方便沟通就够了，而完全按照 UML 规范来将草图标准化，所付出的代价是不值得的。

所以，我这里特别说明一下，专栏中的很多类图我并没有完全遵守 UML 的规范标准。为了兼顾图的表达能力和你的学习成本，我对 UML 类图规范做了简化，并配上了详细的文字解释，力图让你一眼就能看懂，而非适得其反，让图加重你的学习成本。毕竟，我们的专栏并不是一个讲方法论的教程，专栏中的所有类图，本质是让你更清晰地理解设计。

重点回顾

今天的内容讲完了，我们来一起总结回顾一下，你需要重点掌握的几个概念和知识点。

1. 什么是面向对象编程？

面向对象编程是一种编程范式或编程风格。它以类或对象作为组织代码的基本单元，并将封装、抽象、继承、多态四个特性，作为代码设计和实现的基石。

2. 什么是面向对象编程语言？

面向对象编程语言是支持类或对象的语法机制，并有现成的语法机制，能方便地实现面向对象编程四大特性（封装、抽象、继承、多态）的编程语言。

3. 如何判定一个编程语言是否是面向对象编程语言？

如果按照严格的定义，需要有现成的语法支持类、对象、四大特性才能叫作面向对象编程语言。如果放宽要求的话，只要某种编程语言支持类、对象语法机制，那基本上就可以说这种编程语言是面向对象编程语言了，不一定非得要求具有所有的四大特性。

4. 面向对象编程和面向对象编程语言之间有何关系？

面向对象编程一般使用面向对象编程语言来进行，但是，不用面向对象编程语言，我们照样可以进行面向对象编程。反过来讲，即便我们使用面向对象编程语言，写出来的代码也不一定是面向对象编程风格的，也有可能是面向过程编程风格的。

5. 什么是面向对象分析和面向对象设计？

简单点讲，面向对象分析就是要搞清楚做什么，面向对象设计就是要搞清楚怎么做。两个阶段最终的产出是类的设计，包括程序被拆解为哪些类，每个类有哪些属性方法、类与类之间如何交互等等。

课堂讨论

今天我们要讨论的话题有两个：

1. 在文章中，我讲到 UML 的学习成本很高，沟通成本也不低，不推荐在面向对象分析、设计的过程中使用，对此你有何看法？
2. 有关面向对象的概念和知识点，除了我们今天讲到的，你还能想到其他哪些吗？

欢迎在留言区发表你的观点，积极参与讨论。你也可以把这篇文章分享给你的朋友，邀请他一起学习。

AI智能总结

本文深入探讨了面向对象编程的概念及其相关内容，包括面向对象编程的历史发展、定义、特性以及面向对象分析和设计。作者强调了面向对象编程的基本特性：封装、抽象、继承、多态，并讨论了面向对象编程语言的判定标准。此外，文章还提到了UML（统一建模语言）的复杂性及其在实际项目中的使用情况。总体而言，本文全面深入地探讨了面向对象编程的相关概念，适合对该主题感兴趣的读者阅读。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

全部留言 (357)

最新 精选



王争 置顶

2019-11-11

UML中定义了类之间的关系：泛化、实现、关联、聚合、组合、依赖，试问下小伙伴们，你们都能搞清楚这几个的区别吗？能否准确的用不同的箭头、图线来画出来吗？即便你能画出来，团队里的小伙伴都能看懂吗？不过，关于类之间的关系，我后面会在实战篇中讲到的，但是，我会简化成四种关系，更好理解。

共 35 条评论 >

145



王争 置顶

2019-11-11

在这篇文章中，“面向对象编程”一词多义，不同的场景、语境下，解释不同。文章中没有点到这一点，我这里稍微补充说明一下：

1. 文章前半部分，面向对象编程指的是一种编程风格或者范式。
2. 文章后半部分，在讲到面向对象分析、设计、编程的时候，面向对象编程是一种行为。

**daniel李**

2019-11-11

当看到老师说uml意义不大的时候我就懵了，还好原来是指不需要按严格标准死磕uml。

我平时在功能开发初期和后期都是用uml把我的想法可视化然后让师兄审核，减少pr被reject机率。而且也容易让别的工程师接手做功能拓展。

不过确实互联网公司如果不是大厂，确实很少人能看懂uml。

作者回复: 实际上，大厂也未必都在用。比如类图中几种类关系，同学们有几个能准确的用不同的图线画出来呢？

**BK**

2019-11-12

确实很多公司使用 UML 的频率比较低，大厂也是这样。另外我自己对 Google 的了解是对代码要求很高，那么代码交付还有文档的要求想必也很高，不知道 Google 公司在文档上面会不会大量使用 UML，如果不用的话，有没有更好的方式或语言来维护高质量的文档

作者回复: 好像没有多少人画uml，文档主要还是聚焦于业务和更上层的架构设计、技术难点说明，实现直接看代码。

**编程界的小学生**

2019-11-11

1.我觉得首先uml这东西很牛逼，很有必要去画，但是也需要分场景，比如crud还强行画一个出来那就是浪费时间，比如超级复杂的东西要画，那我觉得就可以简化，多配上文字注释。比如需求一般，不是很复杂也不是很简单的那种也可以好好画一下，必要的地方配上文案描述。uml能帮助我们瞬间理解这个东西到底要做什么，流程是怎样的，画出来不光是现在看还是以后复习看，他都很香！

2.我觉得缺少了一个“组合”，首先要以类和对象作为代码的基石，还要能灵活的支持组合特性才算不严谨的面向对象语言。组合算封装特性的一部分吗？还是说只要以类和对象为基石的开发语言都支持组合？

作者回复: 组合跟封装应该没啥关系呢。



👍 5



潇潇雨歇

2019-11-11

1, 关于UML, 大学的时候学过, 但是没有好好学, 大概能看得懂, 非常复杂的话不必弄的太规范; 可以在自己画的时候做个讲解, 关于规范UML的讲解, 以免学生误解。简化不等于乱画, 大体思想核心是在的。我觉得这样挺好的。可以使用类UML的形式, 《Head First 设计模式》也是用的这种, 大家更能看懂。因为学习UML是有成本的, 老师可以在课程中讲解一些UML的基本用法, 不需要详细介绍。

2, 面向对象的话还有接口, 组合等概念。我学的是JavaScript, 其实js严格来说并不是面向对象编程语言, 像类本身都是没有的, 可以说是基于对象编程语言。Java这种就是严格上的面向对象编程语言。其实我也想学学Java, 老师有什么好的学习资料推荐吗? 比如书籍, 文档, 博客什么。感谢🙏

作者回复: 周末会上一篇加餐, 你可以简单看下专栏中用到的所有的java语法。至于书籍推荐, 你可以随便找本比较薄的java书籍看看就可以, 先入个门。

共 2 条评论 >

👍 4



Kang

2019-11-12

请教下 mysql源码里面是否实现面向对象的思想呢, 怎么理顺一部分, 比如源码里的优化器

作者回复: 我没读过mysql源码呢, 不大清楚你说的那个问题。



👍 1



return

2019-11-12

老师, 面向对象主要用来解决扩展复杂性, 写出来的代码确实要比面向过程更舒服 更易扩展。

但是 我思考其 为什么会有这种效果时, 又说不上具体原因, 有人说是 面向对象 把 实体和行为节藕, 实体是稳定的, 而面向过程只有行为, 行为是不稳定的。好想有点道理 但是又有点抽象。

老师能否指点指点😊

作者回复: 看下一篇文章, 看能否解决你的疑惑呢



1



Running

2019-11-11

古老的C语言是不是可以认为面向对象的编程语言呢? 结构体可以认为是类吗?

作者回复: c是面向过程的编程语言 后面会讲到 别急

共 2 条评论 >



1



初心

2019-11-11

有一点很有体会:使用面向对象编程语言开发出来的功能未必符合面向对象的思想, 很可能是面向过程的。如何避免这种问题呢? 后续有类似案例吗? 争老师

作者回复: 有的 看专栏目录就知道了



1