

Java语言面向对象编程

面向对象思想概述

1.什么是面向对象

在之前我们在方法那一章节中介绍了关于面向过程的开发思想，它是一种软件开发的方式。对应的面向对象也是一种软件开发方式，为什么会出现面向对象的开发方式，那它一定是为了弥补面向过程所带来的一些缺陷。

1.1，面向对象提出背景—软件危机

在早期的软件开发过程中，大多数的软件都是采用面向过程的开发方式进行开发的，但是随着软件的覆盖面越来越广，软件的复杂性和需求也就越来越高，但是面向过程的开发方式是以方法或者说是函数为中心的，也就是固定的去逐步解决问题，但是在面向一些复杂的业务逻辑时，面向过程的逐步求精就显得不够灵活同时当时并没有软件工程化的思想，从而导致许多的公司投入了大量的资金，人力和时间之后任然无法得到预期的软件产品而破产，从而在那个时期阶段中软件开发陷入了一种停滞的状态，这就是软件危机。

后来随着一些新的软件开发思想的提出，其中影响最大的就是面向对象的开发思想和软件工程化，也就是IEEE制定的软件工程提出之后，软件的开发逐渐的变得灵活和能够以团队化，工程化的方式开发，管理和维护软件，使得大型的，复杂的软件得以构建。
















1.2，面向对象核心内容

面向对象是一种解决软件开发的方法和范式，它提供了一种软件开发的思想。它的主体内容包括了：

- 面向对象分析（Object Oriented Analysis, OOA）
- 面向对象设计（Object Oriented Design, OOD）
- 面向对象编程（Object Oriented Programming, OOP）

我们作为程序员在大多数的地方所说的面向对象代表的是OOP的内容，当然它只是面向对象的一个部分，也就是面向对象编程。

想要进行面向对象的开发，那么所使用的编程语言一定要支持面向对象开发的性质，Java是一种纯面向对象开发的语言，当然现在的很多主流编程语言都是支持面向对象开发的比如C++，Python，JavaScript等，同样也有C这样的主流编程语言并不支持面向对象的开发而是一种纯面向过程的开发的语言。这里所说的主流语言是根据[TIOBE Index - TIOBE](#)具有权威性的编程语言排行榜查阅的，当然并不是说排名最高的语言就最好，需要看的是它所涉及的领域，比如C++在桌面软件等应用领域是远远超过Java的，Java在服务器领域是远远超过C++的，而目前随着人工智能和大数据的发展，Python一举成为第一。这里我想说的是Java并不是第一个面向对象的编程语言并且面向对象也不是Java特有的一种内容，Java只是在面向对象的思想中真正的将面向对象进行实践的工具，Java之所以能够在很多面向对象语言中作为面向对象语言的代表是因为它支持的面向对象的思想运用得很适用于当前软件开发行业的发展。

Mar 2023	Mar 2022	Change	Programming Language		Ratings	Change
1	1			Python	14.83%	+0.57%
2	2			C	14.73%	+1.67%
3	3			Java	13.56%	+2.37%
4	4			C++	13.29%	+4.64%
5	5			C#	7.17%	+1.25%
6	6			Visual Basic	4.75%	-1.01%
7	7			JavaScript	2.17%	+0.09%
8	10	▲		SQL	1.95%	+0.11%
9	8	▼		PHP	1.61%	-0.30%
10	13	▲		Go	1.24%	+0.26%
11	9	▼		Assembly language	1.11%	-0.79%
12	15	▲		MATLAB	1.08%	+0.28%
13	12	▼		Delphi/Object Pascal	1.06%	-0.06%
14	23	▲		Scratch	1.00%	+0.47%
15	17	▲		Classic Visual Basic	0.98%	+0.38%

我们知道计算机一直致力于将现实世界映射到计算机世界中，现实世界中有着各种各样的事物或者是东西（Object）比如一台计算机，一个人，一只猫，一辆车，一栋建筑……。面向对象思想就是将现实世界中的这一个个的东西或事物看作是一个个计算机中的对象，这些对象是具有属性和行为，比如一个人具有姓名，年龄，籍贯，身高，家庭住址，身份等的属性和行为比如吃饭，睡觉，做饭，走路，跑步等行为。通过这些属性和行为来刻画现实生活中的任何一种事物或东西（Object）从而实现一种新的软件开发的思路。并且这种思想所需的属性和行为是在特定的业务环境中构建的。

比如我想要开发一款学生管理系统，那么在这个系统中的参与者是学生，老师和教务人员，对于学生在学生管理系统中应该具有怎样的属性和行为呢，这是根据具体的学校业务要求而决定的，普遍来说学生应该具有姓名，学号，身份证号，联系电话，专业班级，专业这样的属性，对于学生的操作来说他们可以查询成绩，查看课程，办理各种学生业务等。

这就是面向对象思想，当然这听起来并不深奥，但是在应对一些复杂的业务的时候我们会抽象出许多的内容，它是一种具有设计思想的内容，也就是说它是不确定的，相对于面向过程思想来说，面向过程的步骤，比如实现一个冒泡排序，那么基本就是这样的一种算法步骤，是一种固定的过程，而面向对象来说由于它在对一个系统进行建模的时候可以有多样的建模方式抽象出多种不同的内容，比如上述的学生管理系统我们可以把教师和职工抽象成一个教职工然后根据教师和职工的职称来区分，当然也可以分开写，但那一个更好呢？这需要在特定业务下才能判断，所以学习面向对象开发并不是一蹴而就的而是一种需要经验和不断开发的过程才能慢慢体会到我在实现这个系统的时候应该采用哪一种设计才更好。

可以说在最初时，当需求单一，或者简单时，我们一步步去操作没问题，并且效率也挺高（面向过程）。可随着需求的更改，功能的增多，发现需要面对每一个步骤很麻烦了，这时就开始思索，能不能把这些步骤和功能进行封装，封装时根据不同的功能，进行不同的封装，功能类似的封装在一起。这样结构就清晰了很多。用的时候，找到对应的类就可以了（面向对象）。

2,面向对象与面向过程的联系和区别

除了面向对象，还有面向过程、指令式编程和函数式编程。在所有的编程范式中，我们接触最多的还是面向过程和面向对象两种。当然目前一种面向接口编程的思想也在逐渐深入。这里提到的是Java也是支持面向接口编程的，它也是有接口这样的一种内容的。

2.1, 面向对象和面向过程的联系

我们可以考虑这样的一个问题，就是开车和造车。

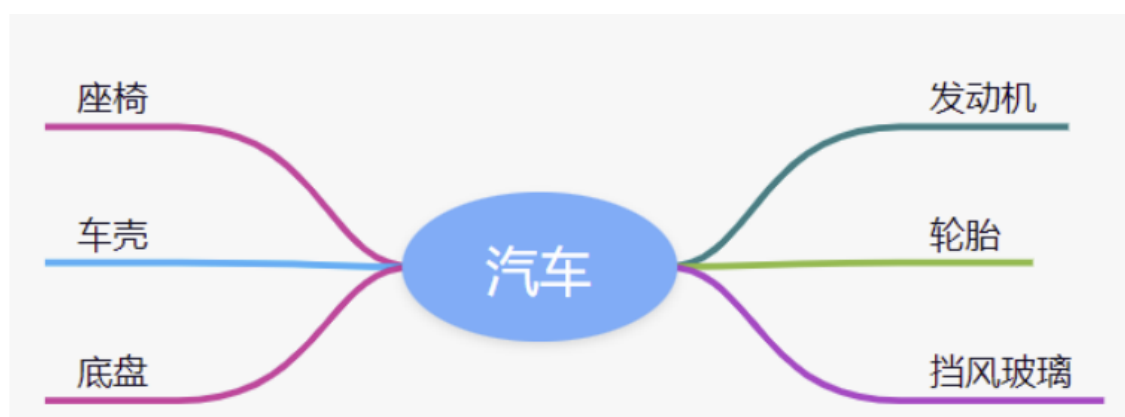
1. 如何开车？

面向过程思想思考问题时，我们首先思考“怎么按步骤实现？”并将步骤对应成方法，一步一步，最终完成。这个适合简单任务，不需要过多协作的情况。针对如何开车，可以列出步骤：



2. 如何造车？

造车太复杂，需要很多协作才能完成。此时我们思考的是“车怎么设计？”，而不是“怎么按特定步骤造车的问题”。这就是思维方式的转变，前者就是面向对象思想。所以，面向对象(Object-Oriented)思想更契合人的思维模式。用面向对象思想思考“如何设计车”，自然地，我们就会从“车由什么组成”开始思考。发现，车由如下结构组成：



我们找轮胎厂完成制造轮胎的步骤，发动机厂完成制造发动机的步骤，...；这样，大家可以同时进行车的制造，最终进行组装，大大提高了效率。但是，具体到轮胎厂的一个流水线操作，仍然是有步骤的，还是离不开面向过程思维！因此，面向对象可以帮助我们宏观上把握、从整体上分析整个系统。但是，具体到实现部分的微观操作（就是一个个方法），仍然需要面向过程的思路去处理。所以我们千万不要把面向过程和面向对象对立起来。他们是相辅相成的。面向对象离不开面向过程！

2.2, 面向对象和面向过程的区别

1. 面向过程的程序设计思想（Process-Oriented Programming），简称 POP 关注的焦点是过程：过程就是操作数据的步骤。如果某个过程的实现代码重复出现，那么就可以把这个过程抽取为一个函数。这样就可以大大简化冗余代码，便于维护。典型的语言：C语言代码结构：以函数为组织单位。是一种“执行者思维”，适合解决简单问题。扩展能力差、后期维护难度较大。
2. 面向对象的程序设计思想（Object Oriented Programming），简称 OOP 关注的焦点是类：在计算机程序设计过程中，参照现实中事物，将事物的属性特征、行为特征抽象出来，用类来表示。典型的语言：Java、C#、C++、Python、Ruby 和 PHP 等代码结构：以类为组织单位。每种事物都具备自己的属性和行为/功能。是一种“设计者思维”，适合解决复杂问题。代码扩展性强、可维护性高。

一个经典的例子是如何把大象塞入冰箱。

1. 面向过程：

打开冰箱门。
把大象装进冰箱。
把冰箱门关上。

2. 面向对象：

```
人{
    打开(冰箱){
        冰箱.开门();
    }
    操作(大象){
        大象.进入(冰箱);
    }
    关闭(冰箱){
        冰箱.关门();
    }
}
冰箱{
    开门(){ }
    关门(){ }
}
大象{
    进入(冰箱){ }
}
```

3,面向对象的三大特性

面向对象的三大特性是面向对象程序设计中的重要内容，我们学习OOP实际上就是学习这个语言是如何支持实现这三大特性的，所以它是我们OOP的重点和主线内容。这里我们先来认识一下面向对象的三大特性是什么，后面的我们整个的内容将围绕这三大特性展开。

- 封装
- 继承
- 多态