

java面向对象编程 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022_java_E9_9D_A2_E5_90_91_c104_253533.htm Java 的核心是面向对象编程。

事实上，所有的Java 程序都是面向对象的，你别无选择。这一点与C 不同，因为在那里你可以选择是否面向对象编程。

面向对象编程与Java 密不可分，因此，在你编写哪怕是最简单的Java 程序以前，也必须理解它的基本原则。因此，本章

先从面向对象编程的概念讲起。1 两种范型 我们知道，所有的计算机程序都由两类元素组成：代码和数据。此外，从概念上讲，程序还可以以它的代码或是数据为核心进行组织编写。也就是说，一些程序围绕“正在发生什么”编写，而另一些程序则围绕“谁将被影响”编写。这两种范型决定程序的构建方法。第一种方法被称为面向过程的模型

（process-oriented model），用它编写的程序都具有线性执行的特点。面向过程的模型可认为是代码作用于数据，像C这样的过程式语言采用这个模型是相当成功的。然而，正如在第1章提到的，当程序变得更大并且更复杂时，就会出现问题。

为了管理不断增加的复杂性，第二种方式，也就是面向对象的编程（object-oriented programming）被构思出来了。面向对象的编程围绕它的数据（即对象）和为这个数据严格定义的接口来组织程序。面向对象的程序实际上是用数据控制

对代码的访问。下面你将看到，将控制的实体变换为数据，可使程序在组织结构上从若干方面受益。2 抽象 面向对象编程的一个实质性的要素是抽象。人们通过抽象（abstraction）

处理复杂性。例如，人们不会把一辆汽车想象成由几万个互

互

相独立的部分所组成的一套装置，而是把汽车想成一个具有自己独特行为的对象。这种抽象使人们可以很容易地将一辆汽车开到杂货店，而不会因组成汽车各部分零件过于复杂而不知所措。他们可以忽略引擎、传动及刹车系统的工作细节，将汽车作为一个整体来加以利用。使用层级分类是管理抽象的一个有效方法。它允许你根据物理意义将复杂的系统分解为更多更易处理的小块。从外表看，汽车是一个独立的对象。一旦到了内部，你会看到汽车由若干子系统组成：驾驶系统，制动系统，音响系统，安全带，供暖，便携电话，等等。再进一步细分，这些子系统由更多的专用元件组成。例如，音响系统由一台收音机、一个CD播放器、或许还有一台磁带放音机组成。从这里得到的重要启发是，你通过层级抽象对复杂的汽车（或任何另外复杂的系统）进行管理。复杂系统的分层抽象也能被用于计算机程序设计。传统的面向过程程序的数据经过抽象可用若干个组成对象表示，程序中的过程步骤可看成是在这些对象之间进行消息收集。这样，每一个对象都有它自己的独特行为特征。你可以把这些对象当作具体的实体，让它们对告诉它们做什么事的消息作出反应。这是面向对象编程的本质。面向对象的概念是Java的核心，对程序员来讲，重要的是要理解这些概念怎么转化为程序。你将会发现，在任何主要的软件工程项目中，软件都不可避免地要经历概念提出、成长、衰老这样一个生命周期，而面向对象的程序设计，可以使软件在生命周期的每一个阶段都处变不惊，有足够的应变能力。例如，一旦你定义好了对象和指向这些对象的简明的、可靠的接口，你就能很从容很自信地解除或更替旧系统的某些组成部分。

3 面向对象编程

的3个原则 所有面向对象的编程语言都提供帮助实现面向对象模型的机制，这些机制是封装，继承及多态性。现在让我们来看一下它们的概念。封装 封装（Encapsulation）是将代码及其处理的数据绑定在一起的一种编程机制，该机制保证了程序和数据都不受外部干扰且不被误用。理解封装性的一个方法就是把它想成一个黑匣子，它可以阻止在外部定义的代码随意访问内部代码和数据。对黑匣子内代码和数据的访问通过一个适当定义的接口严格控制。如果想与现实生活中的某个事物作对比，可考虑汽车上的自动传送。自动传送中包含了有关引擎的数百比特的信息，例如你正在以什么样的加速度前进，你行驶路面的坡度如何，以及目前的档位。作为用户，你影响这个复杂封装的方法仅有一个：移动档位传动杆。例如，你不能通过使用拐弯信号或挡风玻璃擦拭器影响传动。所以档位传动杆是把你和传动连接起来的惟一接口。此外，传动对象内的任何操作都不会影响到外部对象，例如，档位传动装置不会打开车前灯！因为自动传动被封装起来了，所以任何一家汽车制造商都可以选择一种适合自己的方式来实现它。然而，从司机的观点来看，它们的用途都是一样的。与此相同的观点能被用于编程。封装代码的好处是每个人都知道怎么访问它，但却不必考虑它的内部实现细节，也不必害怕使用不当会带来负面影响。Java 封装的基本单元是类。尽管类将在以后章节详细介绍。现在仍有必要对它作一下简单的讨论。一个类（class）定义了将被一个对象集共享的结构和行为（数据和代码）。一个给定类的每个对象都包含这个类定义的行为和结构，好像它们是从同一个类的模子中铸造出来似的。因为这个原因，对象有时被看作是

类的实例（instances of a class）。所以，类是一种逻辑结构，而对象是真正存在的物理实体。当创建一个类时，你要指定组成那个类的代码和数据。从总体上讲，这些元素都被称为该类的成员（members）。具体地说，类定义的数据称为成员变量（member variables）或实例变量（instance variables）。操作数据的代码称为成员方法（member methods）或简称方法（methods）。如果你对C/C++熟悉，可以这样理解：Java程序员所称的方法，就是C/C++程序员所称的函数（function）。在完全用Java编写的程序中，方法定义如何使用成员变量。这意味着一个类的行为和接口是通过方法来定义的，类这些方法对它的实例数据进行操作。既然类的目的是封装复杂性，在类的内部就应该有隐藏实现复杂性机制。类中的每个方法或变量都可以被标记为私有（private）或公共（public）。类的公共接口代表类的外部用户需要知道或可以知道的每件事情；私有方法和数据仅能被一个类的成员代码所访问，其他任何不是类的成员的代码都不能访问私有的方法或变量。既然类的私有成员仅能被程序中的其他部分通过该类的公共方法访问，那么你就能保证不希望发生的事情就一定不会发生。当然，公共接口应该小心仔细设计，不要过多暴露类的内部内容。100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com