

ITO之路

——奔跑在 Java 的路上

田玉娟 赵国栋 张林福 编著

钱伟荣 主审



中国商务出版社
CHINA COMMERCE AND TRADE PRESS

ITO 之路——奔跑在 Java 的路上

田玉娟 赵国栋 张林福 编 著
钱伟荣 主 审

中国商务出版社

图书在版编目 (CIP) 数据

ITO 之路:奔跑在 Java 的路上/田玉娟, 赵国栋,
张林福编著. —北京: 中国商务出版社, 2010. 9
ISBN 978-7-5103-0299-2

I. ①I… II. ①田…②赵…③张… III. ①
JAVA 语言—程序设计 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 180482 号

ITO 之路——奔跑在 Java 的路上
ITO ZHILU——BENPAO ZAI JAVA DE LUSHANG

田玉娟 赵国栋 张林福 编 著
钱伟荣 主 审

出 版: 中国商务出版社
发 行: 北京中商图出版物发行有限责任公司
社 址: 北京市东城区安定门外大街东后巷 28 号
邮 编: 100710
电 话: 010—64269744 (编辑室)
010—64266119 (发行部)
010—64263201 (零售、邮购)
网 址: www.cctpress.com
邮 箱: cctp@cctpress.com
照 排: 卓越无限
印 刷: 北京市松源印刷有限公司
开 本: 889 毫米×1194 毫米 16
印 张: 44 字 数: 1107 千字
版 次: 2010 年 10 月第 1 版 2010 年 10 月第 1 次印刷

书 号: ISBN 978-7-5103-0299-2

定 价: 98.00 元

版权专有 侵权必究

举报电话: (010) 64242964

前 言

服务外包作为一项国家战略，在中国的经济舞台上扮演着越来越重要的角色。以软通动力为代表的“中国服务外包六小龙”越来越多的在国际竞争中展现中国服务外包的形象。企业和学校共同合作，推出服务外包系列丛书，就是希望能够同时发挥学校的学术专长和企业的实践经验，能够为中国的有志之士提供一套优质教材，从而为中国的服务外包业培养更多的合格人才，使中国服务外包在国际竞争中立于不败之地。

Java 语言的出现，可以说引起了一场软件革命，对软件服务外包的发展更是起到关键的作用。它是一个广泛使用的网络语言，是一种新的计算机概念。鉴于 Java 本身具有的独特特点及其在服务外包行业中广泛使用的前提背景，使用 Java 工具进行开发已经成为当今软件行业的主流。Java 书籍也如雨后春笋般出现在各大书店的热销书架上。然而，以往的书籍更多地将讲授技术的重点放在理论知识的讲解上，重要的“实践”性环节涉及极少。

为了弥补目前软件书籍市场的这一空白，应广大学生以及用人单位的需求，这样一本兼具理论性和实践性的《ITO 之路—奔跑在 Java 的路上》应运而生，期待一场关于 Java 授业解惑的“软件革命”。我们已经启程，并且希望能和大家共同分享我们的知识、心得和体会。

本书特点在于：

一、资深开发人员编写，“实践”与“知识”完美结合

本书编著者均在 Java 领域有数年开发经验，在开发中积累了宝贵的经验，加之均是讲师与程序员的双重身份，即为开发项目同时参与到企业内部的用人培训中来。得力于这种独特的经营模式，笔创团队积攒实力在编书过程中将“实践”与“知识”完美结合。

二、利用案例进行讲解，深入浅出

引用开发过程中常见的具有代表性的案例，进行讲解分析，传授方法。知识点详细，易于理解。通过真实案例培养学生的动手能力，达到真正的“学以致用”。

三、重点突出，快速提高

重点、难点突出。将开发过程中经常遇见的问题融入到知识点当中，逐个击破，迎刃而解，提高读者解决各类问题的能力，最终真正的做到融会贯通。

笔者寄语：

古人云：读一本不适合自己的书不如不读书。可见读一本适合自己的书的重要性。本书可以说是为软件相关专业学生和工作者的，Java 爱好者及业余钻研者量身定做，针对性强，实践性强，对于广大读者来讲可以称得上是一本好书。但因笔者都是首次将经验付诸于文字写作成书，我们尚处于探索阶段，书中可能有不足之处。我们竭诚欢迎大家批评指正，使之进一步完善。

“路漫漫其修远兮，吾将上下而求索”。让我们一起“奔跑在 Java 的路上”，携手前进！

编者背景介绍：

软通动力信息技术有限公司（iSoftStone）是领先并立足中国，服务全球的业务咨询、ITO 及 BPO 服务提供商，拥有中国本土服务外包行业“六小龙”之一的美誉。软通动力是 ISO27001（BS7799）信息安全认证企业，并于 2006 年 3 月通过 CMMI5 级认证。是“2009 Red Herring 全球百强”企业、“2009 年中国服务外包十大领军企业”（商务部）、“2009 中国软件业务收入前百家企业”（工信部），并连续四年入选“德勤中国高科技、高成长 50 强”。

天津对外经济贸易职业学院是由天津市政府批准，国家教育部备案，天津市唯一一所经贸外语类的全日制高等职业学院，隶属于天津市商务委员会和天津市教育委员会领导。鉴于当前 IT 服务外包产业人才短缺和大学生就业难的现实，为了增强服务外包企业和高校之间的交流以达到教育与时俱进，增强毕业生在人才市场上的竞争力的目的，2007 年 12 月天津对外经济贸易职业学院正式被天津市商务委员会批准为天津市首家服务外包培训基地。该基地拥有目前最完善的实训环境和雄厚的师资力量，为学员提供了以实训为主，并提供企业软件工程师的技术培训，业务流程外包（BPO）培训、IT 服务等多方面的培训服务。2009 年 2 月获批教育部与天津市政府共建的滨海新区技能型紧缺人才培养基地。几年来，为 IT 外包企业培养出具有知识面广博并且专业实践能力强，职业素质好的实用型、应用型人才，颇为企业青睐。

目 录

第一章 面向对象思想	1
1. 什么是面向对象思想	1
1.1 软件开发的方法学——面向对象思想提出的背景	1
1.2 面向对象的物质基础——类与封装	3
1.3 抽象与代码复用——抽象类与概念抽象	3
1.4 模块的开关——接口与功能抽象	4
1.5 同名但不同参——多态与重载	5
1.6 具有面向对象思想语言的优势	7
第二章 Java 语法基础.....	8
2.1 如何学好 Java 语法	8
2.2 基础知识准备	9
2.3 数据类型.....	13
2.4 常量与变量.....	20
2.5 运算符与表达式.....	25
2.6 语法与语句.....	37
2.7 数组与字符串.....	63
第三章 类和对象	72
3.1 什么是类.....	72
3.2 类的定义.....	75
3.3 方法的定义.....	76
3.4 构造器.....	95
3.5 方法超载.....	99
3.6 递归	103
3.7 理解包 (package)	106
3.8 控制对类成员的访问	111
3.9 类的嵌套	113
3.10 finalize () 方法.....	115
3.11 final 关键字.....	116

第四章 类的继承和扩展	123
4.1 类的继承	124
4.2 类的组合	129
4.3 选择基础类访问属性	136
4.4 类的多态	136
4.5 多极继承	141
4.6 使用 final 修饰符	142
4.7 抽象类	143
4.8 通用超类	144
4.9 对象的转型	150
4.10 类的设计	153
4.11 匿名类	157
4.12 接口	158
第五章 异常处理	167
5.1 异常处理机制概述	168
5.2 运用 Java 异常处理机制	171
5.3 Java 异常类	180
5.4 用户定义异常	183
5.5 异常处理原则	187
第六章 IO 流	191
6.1 输入流	191
6.2 输出流	197
6.3 对象序列化和反序列化	202
6.4 标准 IO 流	204
第七章 集合框架	207
7.1 Java1.2 之前的容器类库	207
7.2 Java2 中的容器类库	215
7.3 未来的 Java 容器类库	227
第八章 多线程	230
8.1 预备知识	230
8.2 Java 多线程程序设计	237
第九章 Socket 程序设计	263
9.1 预备知识	263
9.2 Socket 程序设计	267
第十章 数据之桥——JDBC 数据库访问技术	280
10.1 JDBC 的历史	280

10.2	JDBC 技术概述	281
10.3	JDBC 常用接口的使用介绍	285
10.4	事务 (Transaction)	293
10.5	JDBC 使用实例	296
第十一章	Web 开发及其基本概念	300
11.1	web 介绍	300
11.2	Html 与 Css 样式	303
11.3	Web 应用程序的概念	310
11.4	JavaScript 实用技术	312
11.5	Tomat 服务器	327
第十二章	J2EE IDE 的使用	333
12.1	Eclipse	333
12.2	NetBeans	341
12.3	Jbuilder	346
12.4	Jcreator	348
第十三章	Servlet 核心技术	349
13.1	Servlet 简介	349
13.2	Servlet 的主要功能和特点	350
13.3	Servlet 的生命周期	351
13.4	Servlet 入口点	353
13.5	Servlet API	353
13.6	使用 HttpServlet 处理客户端请求	375
13.7	Servlet 的过滤功能	389
13.8	Servlet 的请求调度程序	391
第十四章	JSP 核心技术与应用	392
14.1	动态网页技术	392
14.2	为什么要使用 JSP	396
14.3	JSP 基本语法	398
14.4	JavaBean	412
第十五章	Struts 2 的框架技术	420
15.1	MVC 思想	421
15.2	Struts 2 框架技术概述	424
15.3	Struts 2.0 的常用标记	440
15.4	处理一个 form 多个 submit	455
15.5	struts.xml 常用配置解析	458
15.6	验证数据方式	463

15.7	在 Action 类中获得 HttpServletResponse 对象的四种方法	473
15.8	国际化	475
15.9	文件上传	480
15.10	拦截器概述	486
第十六章	Hibernate 数据库持久化技术	496
16.1	Hibernate 入门	496
16.2	对象/关系数据库映射基础	503
16.3	Hibernate 的检索方式	505
16.4	Struts2 与 Hibernate 整合实例	512
第十七章	Spring 框架	523
17.1	Spring 概述	523
17.2	术语介绍	525
17.3	Spring 快速起步	526
17.4	Spring 核心技术简介	530
第十八章	AJAX 技术	547
18.1	AJAX 简介	547
18.2	AJAX 模式	548
18.3	时刻想着用户	548
18.4	几种工具和技术	548
18.5	Ajax 和 Java Script	549
18.6	XMLHttpRequest	549
18.7	CSS	549
18.8	服务器端	550
18.9	开发工具	550
18.10	浏览器兼容性	550
18.11	XMLHttpRequest 对象的使用	551
18.12	XMLHttpRequest 对象概述	551
18.13	方法和属性	552
18.14	交互示例	554
第十九章	WebService	562
19.1	WebService 概念	563
19.2	动机和特征	564
19.3	WebService 的操作模型	565
19.4	XML	566
19.5	SOAP	584
19.6	WSDL 和 UDDI	596

19.7 使用 Axis 开发 WebService	597
第二十章 应用 SSH 实现 CRM 管理系统	605
20.1 SSH 框架项目简介	605
20.2 CRM 项目简介	607
20.3 建立数据库表结构	609
20.4 建立 Web 项目工程	611
20.5 设置项目的属性	611
20.6 进行分层方式开发	614
20.7 配置 Spring 文件	638
20.8 开发 Web 页面	641
20.9 配置 Struts2	644
20.10 部署并运行程序	645
附件一 JDK5 的新特性	647
附件二 JDK1.5 工具类介绍	651
附件三 常见问题	666
参考文献	694

第一章 面向对象思想

本章的主要内容有：

1. 软件开发的方法学——面向对象思想提出的背景
2. 面向对象的物质基础——类与封装
3. 抽象与代码复用——抽象类与概念抽象
4. 模块的开关——接口与功能抽象
5. 同名但不同参——多态与重载
6. 具有面向对象思想语言的优势

1. 什么是面向对象思想

目前，面向对象的思想被软件开发界广为追捧，其实，面向对象思想不是高深的理论，而是根据前人大量编程项目总结出来的一套分析和解决编程方面问题的方法。

以面向对象思想为指导，可以优化 Java 代码的结构，更可以让数据库访问模块变得富有灵活性——即让代码和模块能更好地适应项目需求的频繁变化。

1.1 软件开发的方法学——面向对象思想提出的背景

计算机语言刚出现时，采用的编程语言是面向硬件的机器语言，即用一大堆计算机硬件能识别的 0 和 1 来编写代码。写这种机器代码需要很高的技巧，它们调试起来比较困难，所以当时的程序员都是精英级别的人物。

为了提高编程语言的适用性，提升代码编写的效率，专家们开发了以汇编语言为代表的低级语言，即用一些助记符来代替晦涩的机器语言，例如用 `add a, b`，来计算 `a + b` 的值。这类语言直接与内存和其他硬件打交道，以使用最小系统资源和谋求最大硬件利用率为开发准则，在当时低配置的硬件环境下是有很大的实用意义的。

但是，用这种编程语言开发出来的代码的结构非常混乱，并且很难读懂。这样的情况可能在几百行规模的代码量下并不明显，如果当项目规模扩大，代码量越来越多的时候，这个问题就突显出来了。更有甚者，作者自己在开发结束后，往往会忘记自己写的代码的结构和语意，这样的代码很难维护和阅读。

为解决这类问题，提高代码的逻辑性，软件语言方面的专家提出了初步融合面向对象思想的结构化的面向过程语言。

结构化语言的重要特征是，在代码里分离了代码逻辑和程序数据，并使用了分支（IF）和循环（While 和 For）等关键字来优化代码的逻辑结构。由于结构化的语言使用子模块顺序调用的方式控制代码流程，所以又叫面向过程语言，以 C 语言为代表。

和汇编语言相比，面向过程语言能很大程度地改善代码结构，提高代码的逻辑性。具体表现为以下两点：

一是，由于在面向过程的语言里，可以把功能点以函数（过程）的形式封装，所以代码的重用性能会得到一定程度的提高，即后继项目可以通过调用具有相同功能的函数来利用现有的代码，这对项目的二次开发或者是功能升级是非常有利的。

二是，由于面向过程语言摒弃了使用 `goto`（或者是 `jump`）语句的跳转方式，而使用了分支和循环等结构性语句，所以代码的结构能得到大幅度的改善，从而能在一定程度上提高项目的可读性和可维护性。

然而，在面向过程语言的使用过程中，程序员们发现其依然有本质上的缺陷，具体表现为：

其一，在模块间的函数调用时，由于无法屏蔽模块里不希望被访问修改的关键变量，这将导致模块间有很高的关联度，这对代码维护是非常不利的，往往会修改一处而触动全部代码。

其二，代码的重用级别仅仅是局限于函数级别的，这导致无法大量有效地利用已开发完成的代码成果，这对代码的重复性利用非常不利。

其三，函数的定义是针对具体动作和具体对象的。比如，我们要定义打印的方法，如果我们使用的打印机型号不同，那我们不得不根据不同的型号，定义多个不同名的方法。也就是说，面对业务逻辑（业务上要做的事情）相同但业务逻辑处理对象不同的需求，必须要写成多个不同名的函数，这导致了代码过于注重细节，而无法考虑逻辑性、重用性和维护性等大局方面的问题。

为了解决上述问题，软件语言的专家们吸取面向过程语言中“结构化”和“模块化”等优秀方法，提出了面向对象的指导思想。

面向对象思想参照了现实生活中观察问题、解决问题的思维方式，将应用程序中的一切模型以类的形式，整合成对象，并通过归纳和抽象对象，提取出一套解决问题的方法和思路。

所以，面向对象思想可以通过优化模块间的结构和加固模块间的坚固性，来指导构造软件项目行业“摩天大楼”。

1.2 面向对象的物质基础——类与封装

类是基于面向对象思想编程语言的基础，程序员可以把具有相同业务性质的代码封装到一个类里，通过接口方法向外部代码提供服务，同时向外部代码屏蔽类里服务的具体实现方式。

对象是类的实例，类一般是个抽象名词，比如“人类”，而对象则是具体的物质存在，比如“张三”这个人。

在现实生活中，我们经常会遇到“类”和“对象”这些概念，比如，封装了能实现“用微波加热材料”功能的微波炉类。这里，用户可以使用面板上的按钮，使用该类里提供的加热等服务，并且，由于该类实现了封装，所以在使用的时候，用户可以不用了解其中的微波加热原理以及实现机制。

我们可以看出，类是对现实生活中真正存在的对象的描述，并且这些对象都具有共同的属性和行为。但是，根据不同的系统需求，同样的一种对象会被描述成具有不同属性和行为的类。比如，对于银行系统，人的这个类应该具有账号、存款余额的属性和存钱、取钱的行为，每个人都有自己不同的账号和相关金额，但是他们存钱取钱的流程是相通的；而对于电信系统，同样的人这个类应该具有手机号码、卡内余额的属性和充值、扣费的行为。因此，我们要注意类的定义和它所封装的行为是否能够正确的反映实际系统的需求。

1.3 抽象与代码复用——抽象类与概念抽象

抽象类也是类的一种，它同样拥有属性和行为，但是它与普通类最大的区别就在于抽象类里面允许有一些抽象的行为，抽象类只从概念的角度出发定义了类的行为，但是它没有描述行为的实现细节，而是把这个任务交给了它的子类来实现，它本身充当了父类的角色。因此，可以说子类不仅有从父类继承过来的属性和行为，而且它可以在父类原有内容的基础上做一些补充和发挥。父类只是做了一个概念的抽象。

图 1-1 说明了这种抽象的必要性：

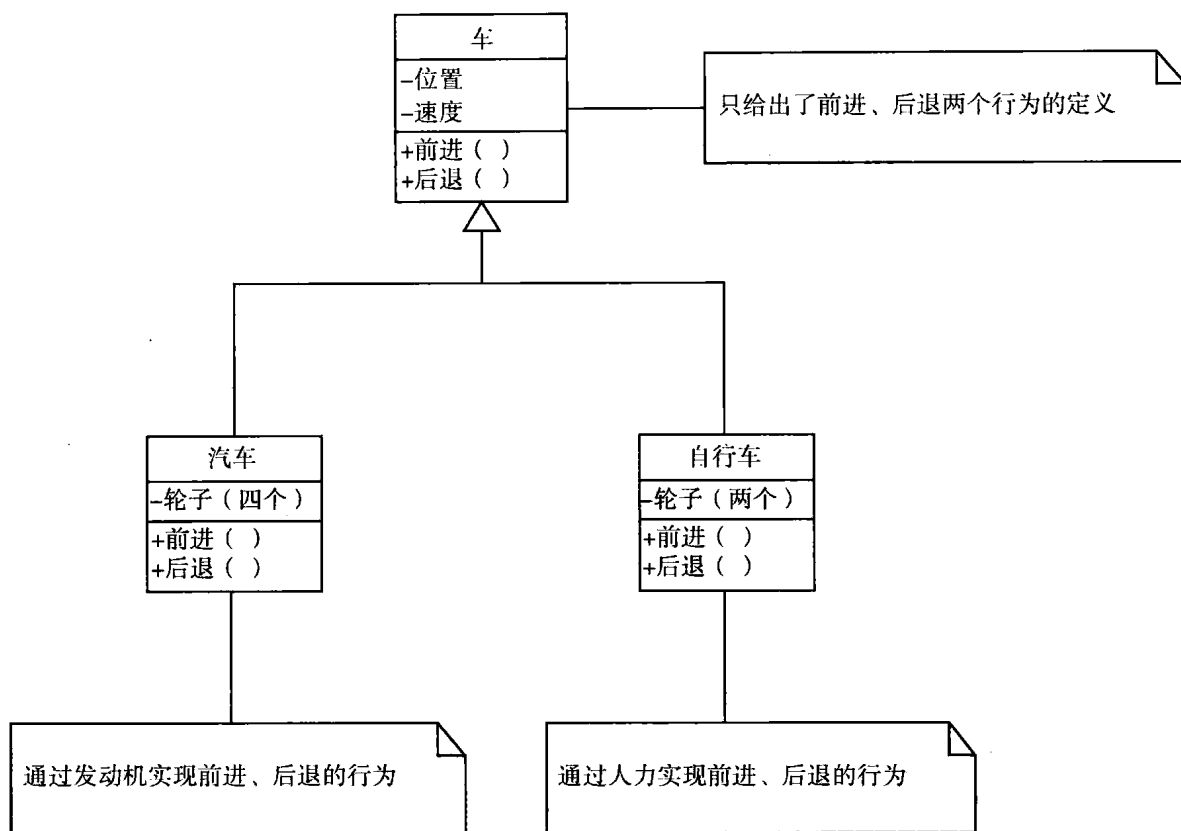


图 1-1 抽象类示意图

对于汽车、自行车这两个类来说，它们都是父类车的子类，具有位置、速度属性和前进、后退行为，但是汽车有四个轮子，自行车有两个轮子，汽车的运动是靠发动机来实现的，而自行车的运动是靠人力来实现的。因此，我们在车这个父类里面只定义了关于运动的两个方法，具体的实现细节交给了它的两个子类来完成。这样我们不仅避免了在两个相近的类之间定义重复的属性和行为，而且还解决了允许不同的类具体的实现细节有所变化的问题。

抽象类可以帮助我们更加有效地解决代码重复问题，并且，它又不失灵活性，又同时达到了概念抽象的目的。这里我们又通过抽象类的概念引出了类之间，一个重要关系，就是子类与父类的继承，当然不仅抽象类可以作为父类，其他的一般类同样可以充当为某一子类的父类。只要它有这种继承的需求我们就可以这样来定义。

1.4 模块的开关——接口与功能抽象

接口的功能与抽象类相似，但是接口只能定义行为，这些行为的具体细节在实现了接口的类中描述，而抽象类还可以为这些类定义一些相同的属性，因此我们可以说，接口比抽象类要更加抽象。接口中只定义公共的行为，这些行为也就是实现它的那些类的功能，不同的类可以有不同实现细节，

这里的原理与抽象类相同。

图 1-2 说明了接口是如何做到功能抽象的：

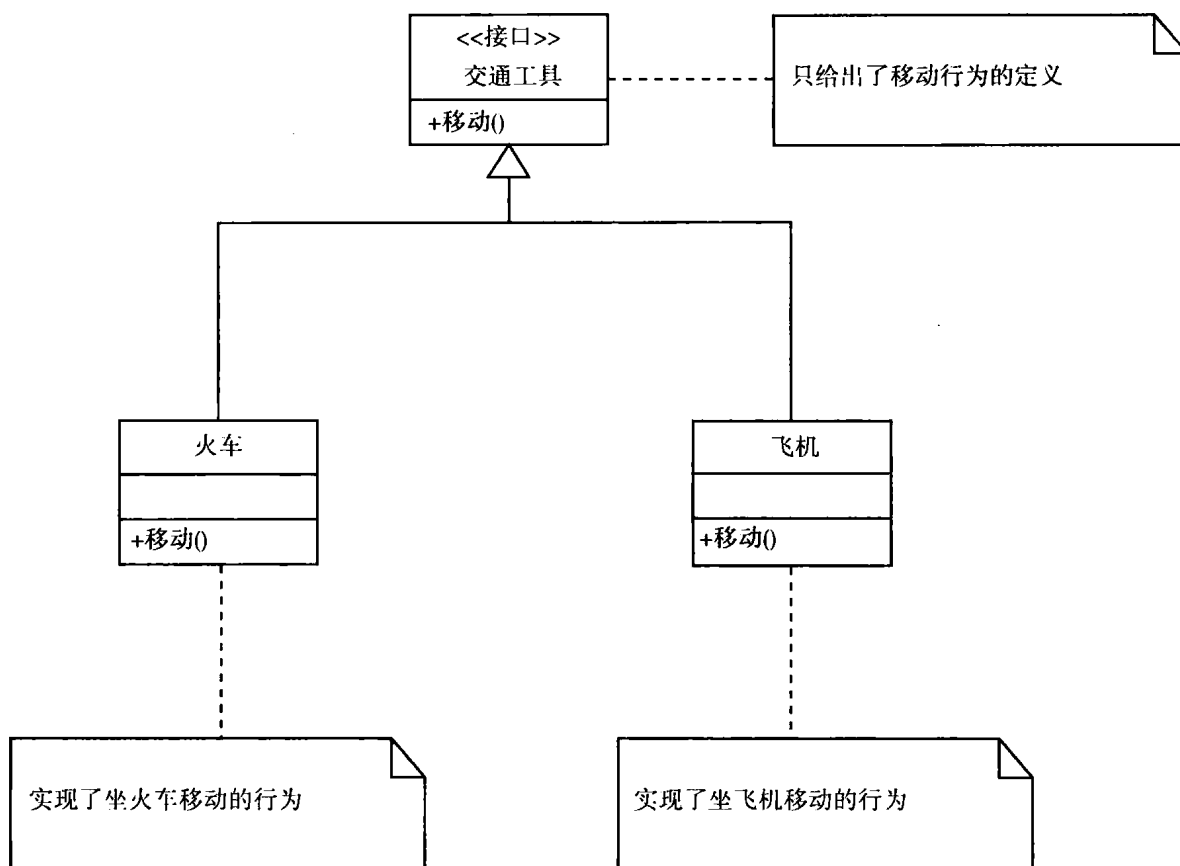


图 1-2 接口示意图

我们定义了交通工具这个接口，在这个接口里面定义了移动这个行为，但是行为的具体实现细节是在火车和飞机两个类里面完成的，并且是由不同的方式来实现的。假设某个用户有了一个从上海到北京的行程安排，并且由于时间紧，不能确定选择哪一种具体的交通工具，只知道要发生移动这个行为。那么，他可以先选择交通工具这个接口的移动行为，是坐火车还是坐飞机我们可以根据实际的情况来定。

接口是单单从行为的角度出发，把功能做了概念上的抽象，使得不同的类通过不同的机制实现名称相同的行为，方便使用者调用，隐藏了具体的实现细节。

1.5 同名但不同善——多态与重载

多态是面向对象编程的一个重要特征，从字面意思来看就是指具有多种形式或形态的情形。在面向对象的编程中，多态主要是指针对于类的行为有多种不同的变化，用相同的语言结构，来表示

不同类型的实体或对不同类型的实体进行操作。即“相同的界面，不同的内部实现形式”。通常通过重载和覆写来实现多态。

重载是让类以统一的方式处理不同类型数据的一种手段。就是在类中可以创建多种行为，这些行为有一个特征是它们具有相同的名字，但具有不同的参数和不同的表现形式。调用行为时通过传递给它们的不同个数和类型的参数来决定具体使用哪个方法，这就是多态性的一种表现，虽然行为的名称相同，但得到的是不同的表现形式。

重载在我们的现实生活中也很常见。比如：保姆类有洗的这个行为，但是我们可以给她衣服或餐具等不同的对象，她会根据不同的输入实现不同的洗的细节，如果是衣服的话就放在盆里洗，如果是洗碗就在水池里洗。这带来最大的好处就是对于命令的发起者主人来说不需要记住太多的洗衣服还是洗碗的方法，只要把这些脏东西交给保姆并且发出一个洗的指令，保姆就会自动地进行处理，屏蔽了细节实现，不仅方便了主人，同时也提高了自己的价值。

让我们谈谈另外一种多态的实现——覆写。还记得我们上面提到过的子类和父类吗？是的，子类不仅可以实现父类里面的只给出具体定义而没有实现细节的行为，子类同时也可以改变父类中的行为，这个我们可以称之为“有所继承，有所发挥”。当你发现老爸那一套做法已经落伍了的时候，你是否会去改变它呢。我们把这种在子类中改变父类行为的具体实现细节，但是名字和接受的输入对象不变的动作就称之为覆写。

覆写在我们的现实生活中同样有着重要的意义。还拿上面的那个例子来说，原先的那个保姆带了一个徒弟，因为徒弟从师傅那里学会了如何做个好保姆，特别是如何洗东西，做保姆对于她来说可能更有优势。

但是时代进步了，我们的主人家里添置了洗衣机、洗碗机等家用电器。徒弟发现用这些东西会更加方便，既省时又省力，于是她快速地学会了如何操作这些家用电器来为主人提供更加优质的服务。对于她来说同样可以洗衣服和洗碗，只不过用的是洗衣机和洗碗机，而不是像她的师傅放在盆里或水池里洗了，并且现在洗得更快更好了。

对于任何一个保姆来说，主人只要给她脏东西并且发出洗的命令，她就会根据实际情况进行相应的操作，这时在这个洗的行为上发生了重载。主人把同样的碗这个对象交给保姆师傅和保姆徒弟来洗，她们也会进行相应的操作，师傅放在水池里洗，而徒弟用洗碗机洗，此时在这个洗的行为上却发生了覆写。

总结一下，类的行为可以有多种不同的表现形式，这叫做多态。对于那些具有相同的名字但是不同操作对象的行为来说，我们称之为重载；对于那些不仅拥有相同的名字而且操作的对象也相同，但是在子类与父类里面具有不同实现细节的行为来说，我们称之为覆写。

在编程语言中，那些名字相同参数类型不同的方法就叫做重载，那些分别处在子类父类里面名字相同参数类型也相同的方法叫做覆写。

1.6 具有面向对象思想语言的优势

传统的程序设计方法称为面向过程的程序设计方法，它是以具体的业务逻辑为基础，以实现方法为主要目的。它的缺点是：程序往往缺乏对代码重用性的支持，并且缺乏统一的接口，使得当代码的规模达到一定程度时，程序变得难以控制和管理，它的复杂性大大增加了开发人员的工作量。

面向对象的程序设计方法以一种更接近人类一般思维的方式去看待世界，按照现实世界的特点来管理复杂的事物，它根据实际的情况把生活中的对象抽象成相应的属性和行为组成的实体，对象之间通过互相调用来协调工作。

面向对象编程的基本单元是类，一个类描述了对应的特性，并把数据和操作数据的行为（称为方法）封装在一起。类可看成是对应的模板，基于这些模板我们可以派生出对应的对象，我们的工作就是把它们有条不紊地组织在一起。而且这样的组织方式更加符合人类认知世界的过程。我们把现实世界中的事物，根据它们的特征和关系定义到了我们的系统中，方便了操作和管理。

面向对象的语言大大提高了程序的重用性，降低了程序的复杂度，使得计算机可以处理更复杂的应用需求。Java 不仅具备面向对象语言的所有特征，而且相对于别的面向对象的语言，它还屏蔽了许多繁琐的操作，简化了难度，降低了开发人员的学习曲线，在面向对象的编程领域中占有举足轻重的地位。