

大学程序设计基础

——系统化方法解析&Java描述

沈 军·编著

本书特点

- 基于系统化思维策略，解析程序设计本质及其三个基本要素和相互关系。
- 从语言学角度解析程序设计语言机制与自然语言机制之间的认识通约性，降低程序设计语言学习的认知难度。
- 基于现代认知科学理论，解析程序设计中的若干基本应用模式及其建构方法，建立学习程序设计的有效方法。
- 强调计算思维的具体应用。



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

大学程序设计基础

——系统化方法解析 & Java 描述

沈 军 编著

 东南大学出版社
SOUTHEAST UNIVERSITY PRESS

· 南京 ·

内 容 提 要

本书采用基于系统化思维的解析方法,首先论述程序设计的本质并给出程序设计的三个基本视图(第一部分)。然后,从语言、环境和应用三个方面分别解析程序设计的内涵,使读者掌握程序设计应有的系统化思维、相关技术思想和方法,对程序设计有一个系统化的认识(第二部分至第四部分)。最后,通过丰富的实例介绍了程序设计中的若干基本应用模式及其建构方法(第四部分)。书中所有思想和方法均通过 Java 语言进行描述。

本书可以作为普通高校计算机相关专业、软件学院的本科教材,也可作为理工科各专业的公共课教材,同时也适合对程序设计感兴趣,希望培养系统的程序设计认识能力和计算思维能力的软件从业人员做参考。

图书在版编目(CIP)数据

大学程序设计基础:系统化方法解析 &Java 描述/
沈军编著. —南京:东南大学出版社,2015. 7
ISBN 978 - 7 - 5641 - 5699 - 2
I. ①大… II. ①沈… III. ①JAVA 语言—程序设计—
高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2015)第 089332 号

大学程序设计基础——系统化方法解析 &Java 描述

出版发行 东南大学出版社
社 址 南京市四牌楼 2 号
邮 编 210096
出 版 人 江建中
责任编辑 张 煦

经 销 全国各地新华书店
印 刷 江苏凤凰数码印务有限公司
开 本 787 mm×1092 mm 1/16
印 张 29
字 数 724 千字
书 号 ISBN 978 - 7 - 5641 - 5699 - 2
版 次 2015 年 7 月第 1 版
印 次 2015 年 7 月第 1 次印刷
定 价 68.00 元

(本社图书若有印装质量问题,请直接与营销部联系,电话:025—83791830)

前 言

现代计算是以计算机为核心,基于图灵机理论模型和冯·诺伊曼体系结构模型的计算机,是通过程序控制电路工作来实现计算。因此,程序设计成为利用计算机进行工作的关键。

程序是人类思维的产物,是人类思维火花的时间定格,呈现静态特征。作为产生程序的过程——程序设计,却是动态的,反映了人类思维的规律和模式。这种规律和模式正是计算思维的具体表现。因此,程序设计中思维能力的培养极其重要。

程序设计是针对计算机特殊环境的思维表达,这种思维表达通过计算机语言描述并受到计算机系统环境的影响和制约。在此基础上,形成各种各样的方法、技术和基本应用模式。目前,针对程序设计的教材基本上分为以程序设计语言为主、以算法和数据结构为主、以开发环境使用为主和以面向特定应用的案例解析为主四种基本形态。第一种教材在介绍语言的同时,非系统性地穿插介绍一些程序构造基本概念和方法以及一些简单的用于体现语言机制具体使用的样例,其弊端是忽略了语言应用能力的培养。针对第一种教材的缺点,第二种教材强调了应用基础,但是,它忽略了知识梯度带来的认知迁移难度,破坏了知识的系统性认知,即算法和数据结构本身具有相对系统的知识体系,它属于方法层面,这些方法可以用程序设计语言来描述。但是,因为学时的问题,要么只能支离破碎地介绍一些,要么大幅度提升认知难度(即将后续课程内容强加给程序设计课程)。第三种教材往往结合一种具体的开发环境,以开发环境如何使用为主线,穿插介绍程序设计语言及体现语言机制和环境机制具体使用的样例。这种教材的缺点是就事论事,适合于技能训练班使用。第四种教材本质上是由应用驱动,强调特定应用的特征及其处理方法,淡化语言本身的学习。这种教材适合于拓展学习。基于上述分析可见,目前的程序设计类教材没有处理好语言本身和语言应用之间的关系,语言与算法交织在一起;或过分热衷于算法,而忽略了基本程序构造方法的系统化思维的训练;或将程序开发环境从程序模型及其基本程序结构形态和基本程序构造方法中割裂开来,对环境支持的编程模型以及环境对语言的影响(调整或拓展)不能够给予重视。尽管目前的教材开始关注语言、环境和应用,但缺乏系统性,并且,对各个方面的介绍没有深入到方法论层面,缺乏应有的思维能力培养。

程序设计涉及语言、环境和应用三个视图。三者之间相辅相成,缺一不可。环境给出了基本的程序运行模型,从而也就间接约束了在其上运行的程序的基本结构,进而也就决定了程序的基本构造方法和技术。语言作为描述的基础和手段,用于描述特定问题的解决思路。在各种特定的环境中,同一种语言会有所改变。应用是整个程序设计的核心,它关注的是特定问题如何解决的思路 and 方案。显然,这种思路 and 方案必须考虑到环境因素、语言因素,不同的环境、不同的语言对同一个问题的解决方案会有不同的形态。更深入地,问题的解决思路具有一定行之有效的模式,这种模式是一种思维活动的产物,

独立于具体的环境和语言,可以映射到具体的环境和语言。可见,应用对于思维的要求比语言和环境对思维的要求更高,因为应用的抽象层次比语言 and 环境的抽象层次要高。

本书针对程序设计的三个视图——语言、环境和应用,从特殊性和普遍性的辩证关系出发,分别阐述了各自对这种辩证关系的诠释(纵向视角)。同时又对三个视图之间的辩证关系给予了论述(横向视角)。本书的目标是传播和阐述程序设计的基本思维和方法,这种思维和方法建立在认识论理论框架基础上,独立于具体计算机语言和开发环境的使用,将具体计算机语言和环境等作为基于思维教学的教学思想演绎的一个案例。也就是说,如果将 Java 语言换成其他语言,不影响教材的本质和教学目标。

基于本书的目标,对下列几部分内容不做展开:

- 对 Java 语言中各种包的内容、各个类或接口的具体细节等并没有给出详细描述,因为只要掌握包、接口和类的相关概念,以及使用它们的方法,也就掌握了其精髓。而具体细节在不同语言中会有不同的具体表现形态,这要根据具体应用时选定的语言确定。程序设计的学习,本质是掌握一种思维方法,这种思维方法可以映射到各种具体语言之中。也就是说,思维和方法不变,语言可以是各种各样的。再者,各种具体语言的细节,可以由其他手册类书给出描述或直接查阅相关电子文档。

- 对于开发环境的具体菜单项的功能,各个对话框的参数细节等,也没有给出详细的解析,因为同一类开发环境具有多种多样的具体产品,但其蕴涵的思想和方法本质上都是一致的。具体某种开发环境的细节,可以参阅该开发环境相应的手册或电子文档。

本书中的观点和思想,源于作者自身对程序设计、计算机学科的感悟和认识,难免存在一些片面性,甚至错误,希望读者能够批评指正。作者将在本书再版时给予重视和纠正。作者的电子邮箱是 junshen@seu.edu.cn。另外,本书也引用了某些名家和同行的观点,这些观点都是经过自身的理解再给出,凡是想到的都将相应的文献列入参考文献之中。由于作者嗜好读书,内容涉及多个学科,时间跨度较大,有的观点记不起具体出自哪本书,或者已融入了作者的主观意识,因此,可能会遗漏一些参考文献。在此,向这些作者表示衷心的感谢!也恳请这些作者与本书作者联系,以便在本书再版时将其列出。

创作是艰苦的,一本书的出版,绝不是一个人能够完成的。尽管本书只署作者一人之名,但本书的出版凝结了许多人的心血。在此要特别感谢东南大学非计算机专业计算机教学指导委员会的全体成员,他们将本书倡导的教学思想和方法直接用于教学实践。感谢董永强副教授、杨鹏副教授、沈单炜副教授和宋爱波副教授,他们为本书出版做了相关的素材整理工作。感谢高文、冒佳明、丁德林、何雯、姜峰、袁明明、万逸、吉祖勤、朱晓建、彭殷路、韩涛等同学为本书所做的截图、样例程序调试及部分编辑工作。感谢所有与我共同试验新教学思想的学生们。感谢东南大学出版社的大力支持,感谢张煦编辑为本书出版付出的辛勤劳动,她对本书的出版提出了一些宝贵意见。

最后特别感谢我的夫人朱凌和我的儿子沈凌翔,他们给予我极大的鼓励,使本书能够顺利完成。

作 者

2014 年 12 月 30 日于古都金陵

目 录

第一部分 程序设计内涵及其学习策略

第1章 认识程序及程序设计	(3)
1.1 什么是程序	(3)
1.2 什么是程序设计	(3)
1.3 程序设计的三个基本视图	(3)
1.4 深入认识程序设计	(4)
1.5 本章小结	(6)
第2章 程序设计的学习策略	(7)
2.1 认识学习的相关概念及基本理论	(7)
2.1.1 显性知识与隐性知识	(7)
2.1.2 演绎策略与归纳策略	(8)
2.1.3 学习理论	(8)
2.2 如何学习程序设计	(9)
2.2.1 概述	(9)
2.2.2 三个视图的学习要点	(10)
2.2.3 学习导图及其解析	(11)
2.3 本章小结	(12)

第二部分 语言

第3章 程序设计语言与语言学	(15)
3.1 语言的基本体系结构	(15)
3.2 程序设计语言的基本体系结构	(15)
3.3 程序设计语言与环境	(16)
3.4 程序设计语言与应用	(17)
3.5 基于语言视图的程序设计学习导图	(17)
第4章 基本符号集与词汇	(20)
4.1 基本符号集	(20)
4.2 词汇概述	(20)
4.2.1 数据类型(词性)	(21)

4.2.2	Java 的基本数据类型	(21)
4.2.3	Java 的数据类型转换(词性转换)	(22)
4.3	保留词(基本词汇之一)	(23)
4.4	标识符(基本词汇之二)	(23)
4.5	常量(基本词汇之三)	(24)
4.6	变量(基本词汇之四)	(25)
4.7	运算符(基本词汇之五)	(26)
4.8	分隔符(基本词汇之六)	(31)
4.9	数组(复合词汇)	(31)
4.9.1	数组的声明和定义(复合词汇的复合说明)	(31)
4.9.2	创建数组(建立具体的复合词汇空间)	(32)
4.9.3	初始化数组(设定复合词汇的初始化值)	(32)
4.9.4	多维数组(以复合词汇数组作为基本词汇,再进行复合。即数组的数组)	(33)
4.10	学习策略	(34)
第5章	短语	(36)
5.1	短语概述	(36)
5.2	表达式(短语的具体表现形态)	(36)
5.3	算术表达式(短语之一)	(36)
5.4	关系表达式(短语之二)	(37)
5.5	逻辑表达式(短语之三)	(37)
5.6	表达式的混合与嵌套(复合短语)	(38)
5.7	学习策略	(39)
第6章	句子	(41)
6.1	句子概述	(41)
6.2	语句(句子的具体表现形态)	(41)
6.3	注释语句	(42)
6.4	数据组织语句	(43)
6.5	输入输出语句	(43)
6.6	计算赋值语句	(44)
6.7	空语句	(45)
6.8	处理流程控制语句	(45)
6.8.1	Java 语言的分支流程控制语句(分支逻辑控制结构的具体实现)	(46)
6.8.2	Java 语言的循环流程控制语句(循环逻辑控制结构的具体实现)	(49)
6.8.3	Java 语言的特殊循环流程控制语句	(52)
6.8.4	Java 语言的例外流程控制语句(例外逻辑控制结构的具体实现)	(54)
6.9	语句的堆叠与嵌套	(55)

6.10 学习策略	(57)
第7章 段落	(58)
7.1 段落概述	(58)
7.2 函数(段落的具体表现形态)	(58)
7.3 函数的定义与使用	(59)
7.3.1 Java 语言中的函数定义	(60)
7.3.2 函数的使用	(62)
7.3.3 函数使用时的参数传递	(63)
7.3.4 函数使用时的结果返回	(65)
7.3.5 函数的递归使用	(65)
7.3.6 函数使用时的变量生命周期与作用域	(67)
7.4 学习策略	(69)
第8章 描写(逻辑段落)	(72)
8.1 描写概述	(72)
8.2 抽象数据类型(描写的具体表现形态)	(73)
8.3 基于抽象数据类型的若干概念	(73)
8.3.1 对象	(73)
8.3.2 类	(74)
8.3.3 实例	(74)
8.3.4 属性	(75)
8.3.5 方法	(75)
8.3.6 构造方法	(76)
8.3.7 析构方法	(77)
8.3.8 消息	(77)
8.3.9 封装	(78)
8.3.10 继承	(78)
8.3.11 多态	(80)
8.3.12 接口	(82)
8.4 Java 语言中抽象数据类型的支持机制	(83)
8.4.1 接口的定义	(83)
8.4.2 类的定义	(84)
8.4.3 关于类修饰符的进一步说明	(85)
8.4.4 对象实例的创建	(85)
8.4.5 对象实例的引用	(86)
8.5 Java 语言中类的组织——包	(86)
8.5.1 包的定义	(87)
8.5.2 包的查找	(88)

8.5.3 包的封装特性与访问	(88)
8.5.4 包的使用	(88)
8.5.5 Java 语言中标准包简介及基本应用	(89)
8.5.6 Java 语言中的事件包及其基本应用	(96)
8.6 一个综合样例——面向对象概念的 Java 语言视图	(106)
8.7 学习策略	(111)
第 9 章 文章	(114)
9.1 文章及其结构	(114)
9.2 Java 语言的程序结构	(115)
9.2.1 独立程序结构	(116)
9.2.2 小程序结构	(121)
9.3 Java 语言程序样例	(125)
9.3.1 界面应用	(125)
9.3.2 多媒体应用	(135)
9.3.3 网络应用	(139)
9.3.4 数据库应用	(142)
9.4 学习策略	(148)
第 10 章 多主线文章	(149)
10.1 为什么需要多主线文章	(149)
10.2 多主线文章的基本构造思想与结构	(149)
10.3 多线程程序构造相关的基本概念	(150)
10.3.1 程序与程序的执行(进程)	(150)
10.3.2 可执行程序的内存在视图及程序执行的基本过程	(151)
10.3.3 进程和线程	(152)
10.3.4 多线程程序构造的核心问题及其解决方法	(153)
10.4 Java 对多线程程序的支持机制	(154)
10.4.1 Java 中的进程与线程	(154)
10.4.2 Java 线程模型	(154)
10.4.3 Java 线程实现	(155)
10.4.4 Java 线程管理	(157)
10.5 Java 多线程程序范例及解析	(163)
10.6 学习策略	(186)

第三部分 环境

第 11 章 环境概述	(191)
11.1 运行环境与开发环境	(191)

11.2	运行环境与程序设计	(192)
11.3	开发环境与程序设计	(193)
11.4	虚拟机环境	(194)
11.5	Java 环境概述	(195)
11.6	环境的学习策略	(196)
第 12 章	分离式开发环境 JDK	(198)
12.1	JDK 概述	(198)
12.2	JDK 的下载、安装与配置	(199)
12.2.1	JDK 的下载	(199)
12.2.2	JDK 的安装	(200)
12.2.3	配置	(203)
12.3	JDK 中的主要开发工具介绍	(205)
12.3.1	Java 编译器:javac. exe	(205)
12.3.2	Java 解释器:java. exe	(206)
12.3.3	小应用程序浏览器:AppletViewer. exe	(207)
12.3.4	Java 反编译器:javap. exe	(208)
12.3.5	Java 文档生成器:javadoc. exe	(208)
12.3.6	Java 调试器:jdb	(209)
12.3.7	Java 归档程序:jar	(211)
12.4	通过 JDK 构建 Java 应用程序	(211)
12.4.1	构建独立应用程序(Application)	(211)
12.4.2	构建小应用程序(Applet)	(213)
12.5	学习策略	(215)
第 13 章	集成式开发环境 JBuilder	(217)
13.1	集成式开发环境概述	(217)
13.2	JBuilder 9 的安装	(219)
13.3	JBuilder 9 开发环境工作方式简介	(223)
13.4	使用 JBuilder 9 开发独立应用程序	(229)
13.5	使用 JBuilder 9 开发小应用程序	(238)
13.6	学习策略	(241)
 第四部分 应用		
第 14 章	应用的内涵及解析	(245)
14.1	应用的本质	(245)
14.2	技术与技术思想的领悟	(246)
14.3	应用模式的发掘及其建构方法的建立	(246)

14.4 应用的学习策略	(247)
第 15 章 应用构造的相关概念与基本方法	(249)
15.1 程序模型及其作用	(249)
15.1.1 程序模型及其演化	(249)
15.1.2 程序模型与程序基本结构形态	(256)
15.1.3 应用体系结构及其演化	(282)
15.1.4 应用体系结构与程序结构	(287)
15.2 应用的基本构造方法	(288)
15.2.1 建模概述	(288)
15.2.2 面向功能的分析、设计与描述	(290)
15.2.3 面向对象的分析、设计与描述	(293)
15.2.4 模式在应用构造中的使用	(299)
第 16 章 基本应用模式及其建构	(301)
16.1 模式的描述	(301)
16.2 基本应用模式及其建构	(301)
16.3 模式及其建构思想的认识	(371)
第 17 章 案例解析	(386)
17.1 概述	(386)
17.2 案例说明	(386)
17.3 程序构造基本模型及软件体系结构	(387)
17.4 程序结构	(387)
17.5 数据库设计	(388)
17.6 程序解析	(389)
17.7 总结	(415)
附录 元集成开发环境 Eclipse 及其基本使用	(417)
f.1 概述	(417)
f.2 下载	(417)
f.3 安装	(420)
f.4 启动 Eclipse	(420)
f.5 使用 Eclipse 开发程序	(423)
f.6 MyEclipse 简介	(429)
f.6.1 概述	(429)
f.6.2 安装	(430)
f.6.3 MyEclipse 基本使用	(434)
参考文献	(453)

第一部分

程序设计内涵及其学习策略

基本要点

- 程序的概念
- 程序设计的概念
- 程序设计的三个基本视图
- 程序设计的思维特性及其本质
- 程序设计的学习策略

第1章 认识程序及程序设计

本章主要内容:什么是程序,什么是程序设计;程序设计的三个视图及其关系;对程序设计的深入认识。

本章重点:程序与程序设计的区别和关系;程序设计的三个视图及其关系;程序设计的内涵与本质。

1.1 什么是程序

所谓“程序”,广义而言,是指对“程”的处理步骤的一种“序”化安排,即“程的序”。也就是说,“程”相当于某件事的处理过程,该过程涉及多个处理步骤,为了有效地处理完这件事,就需要有一个针对处理步骤的具体安排,这种安排就是处理这件事的“程序”。狭义而言,“程序”是指计算机程序,即预定的一组计算机工作指令流,可以直接被计算机执行。其中,“一组指令”表示多个处理步骤,“指令流”表示对一组指令的“序”化安排。显然,计算机程序是通用程序概念在计算机中的一种映射。依据现代计算机的工作原理,运用计算机处理问题,必须首先给出处理该问题的“程序”(即“预定”一组指令流)。

1.2 什么是程序设计

所谓“程序设计”,是指获得“程序”的过程,即如何对问题的处理步骤进行“序”化。

显然,相对于“程序”,“程序设计”的内涵要丰富得多。首先,“程序设计”涉及问题和计算机两个方面,即既需要对所要处理的问题进行分析和理解,明确其需求,又要对计算机技术的发展现状有一定的认识,以便针对需求给出有效的计算步骤。其次,“程序”仅仅是一种结果,是静态的;而“程序设计”则是一种动态的过程,这个动态过程中融入了人的智慧和创造。再次,“程序设计”对思维有一定要求,这种要求主要体现在对问题理解与计算机技术运用两者相结合的认识方面。因为,针对同一个问题,采用同样的计算机技术,不同的人会有不同的设计,这就取决于每个人的思维。最后,“程序设计”本质上也强调了计算思维能力的培养。也就是,相对于“广义程序”的设计,“计算机程序”的设计是一种基于计算机环境和技术约束的特殊程序的设计,它必然受到计算思维的作用和指导。因此,计算思维能力直接决定了“程序设计”的能力。

1.3 程序设计的三个基本视图

计算机程序设计一般涉及三个基本视图,其基本模型如图 1.1 所示。三者之间相辅相成,缺一不可。

在此,环境是指程序设计语言赖以作用的基础,不同的环境对语言有不同的要求,同

一种语言在不同的环境中也会有不同的修正和扩展,以适应环境的需要。环境可以分为程序执行支撑环境和程序开发环境两大类。程序执行支撑环境就是指计算机系统,包括硬件系统和系统软件。程序开发环境主要是指辅助程序开发的工具集。尽管计算机语言由机器语言、汇编语言发展到高级语言,与低层的机器越来越远,与高层的人越来越近,然而,它并没有完全掩盖计算机系统的痕迹,一种语言以及基于该语言的程序,只有与环境有机地糅合在一起,才能产生强大、高效的作用,才能拓展出计算机的无穷威力。

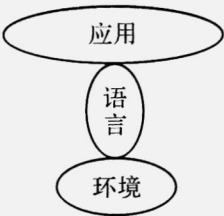


图 1.1 程序设计的三个基本视图

应用是指对具体问题的解决方法,问题的解决方法是多样的、有规律的。然而,在计算机这个特定的环境里,问题解决的方法首先要考虑到计算机系统的特性与体系结构以及用于构建程序的模型、技术和方法。也就是说,程序设计中的应用可以理解为人类思维在计算机环境中的投影,针对特定的环境,人类的思维必然受到限制和约束,这种约束和限制由两方面组成,即环境和程序模型及其延伸的各种构造应用的技术和思想。因此,应用与环境是分不开的,脱离具体的环境,应用是不能最终实现的;同一个应用在不同的环境下将会呈现不同的形态。另一方面,应用与程序模型、技术又是密切相关的,程序模型和技术规定了应用构造的方法,以及应用应有的体系。环境是应用赖以存在的基础,而模型和技术则是应用建立的方法和策略。特别是在约束条件下,可以寻找到各种思维规律,这些规律即反映了特定环境下的应用模式。多种思维规律的灵活应用又可以产生更多的新型思维定式,使得应用的开发始终处于一个创造性的无限循环之中。在此,灵活应用实际上就是指对各种应用模式的组合方法,具有明显的创造特征和个性化特征。从而,也就潜移默化地形成了程序设计所必需的计算思维能力。应用模式的发掘及其逻辑组合(即模式建构)就是程序设计的精髓所在。

语言是一种黏合剂,可将应用和环境连接起来。一方面,语言必须将应用中的各种应用模式及其逻辑组合描述清楚,也就是将思维形式化并记录下来。另一方面,语言本身又考虑了环境的特征,语言的各种机制都体现出环境的痕迹和特点,从而将应用问题过渡到具体环境,实现最终的应用问题求解。

1.4 深入认识程序设计

程序设计是一种创造性劳动,作为程序设计的结果——程序,则是人类思维的直接映射,反映了程序员的思维模式。针对同一个问题,不同的程序员会有不同的程序设计,甚至同一个程序员在不同的时期,对于同一问题也会有不同的程序设计策略。

图 1.2 是图 1.1 的拓展模型,深层次地折射出程序设计与思维的渊源。思维方式是由文化内涵衍生的,不同的文化,营造了不同的生活及工作环境,奠定了不同的思维和行为模式。从世界文化范畴来看,有三种重要的逻辑体系,即中国古典逻辑、古印度古典逻辑和古希腊亚

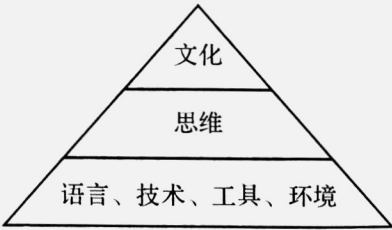


图 1.2 拓展视图

里士多德逻辑。从总体上看,古印度古典逻辑和古希腊亚里士多德逻辑属于同一种类型,基本上是一种形式逻辑,以认识方式为主体,含三个核心内容,即人工语言构成的概念框架、模型化处理和专门的形式体系研究,用形式化(公理系统)表达人的思维过程。而中国古典逻辑是一种超越了的形式逻辑,即辩证逻辑,以辩证关系为主体,其核心是强调思辨,具体的外在表现形式体现在没有统一的形式化系统,依靠感悟。两种思维模式是相反的。

从文化内涵来看,上述两种思维模式正是中西方文化的一种具体表现。西方思维的基本特征是演绎思维,而东方思维的基本特征是归纳思维。前者强调整体概念框架,由此演绎各种具体技术。后者则强调认识,由各种具体的技术归纳或建立整体的概念。计算机诞生于西方,自然地程序设计以及由此而诞生的各种软件工具、各种教材等,都明显地带有这种文化的痕迹,体现了西方文化的思维特征。目前,所有的软件开发工具都提供了基于演绎的框架式可视化设计方法,其蕴涵的思想其实就是一种西方思维的体现特征,即要求使用的人只要学会使用即可,按照它的设计思路做即可。他们认为这样做是顺理成章的,事情就应该如此,没有为什么。也就是说,在两个人配合默契、互相了解的前提下,每个人该干什么、说什么等,一切都在不言之中,一切都是自然的。然而,对于东方文化熏陶下的人而言,深层次的应用却并不简单。因为首先要接受西方文化的思维模式,或者说要了解各种技术、工具的产生背景和思想,然后才能深入理解技术、工具本身,进而灵活应用技术、工具,最后融入东方文化内涵,创造性地进行技术、工具的扩展与应用。

事实上,尽管东方文化的思维特征对于形式化的概念比较生疏和薄弱,但却有较高的感悟能力。形式逻辑可以说是按部就班的思维特征,只要知道是什么(What);而辩证逻辑要求的是原理理解,需要的不仅仅是知道是什么(What),更是要知道为什么(Why),只有通过知道为什么,才能自如的知道是什么。可见,归纳思维解决的是认识本体问题,而演绎思维解决的是在假设前提下的表面已知问题。

特别地,从本质上讲,东方文化对于程序设计应该是有益的,是得天独厚的,这也就是为什么中国学生(在此是指已经熟悉了西方思维的学生)能力比较强的原因所在。然而,中国的软件产业仍然落后,IT人才的流动性特别频繁,导致这些现象的产生固然有各方面的原因,但作者认为,深层原因在于目前的程序设计课程教材体系只是定位于语言和部分算法,沿用的是西方文化的逻辑体系,按西方文化思维习惯演绎,没有充分挖掘东方文化及其思维特征的内涵,基于教材所演绎的教学思维、教学策略等,制约了东方文化对程序设计的作用力,延长了程序设计学习的时间。更为严重的是,忽略了元认知能力培养的时间和生理属性,导致了学习方法、思维方法的畸形发展,从而直接影响了程序设计能力的提高,导致了创新能力的枯竭,丧失了学习的能力,最终也制约了软件产业的发展。

另一方面,尽管每个人的思想不同,但人类的认识活动则存在着一定的规律,人类的认识活动和某个具体个体的认识活动可以看成是普遍性与特殊性之间的辩证关系。普遍性是指共性和规律,可以映射为程序设计活动中的各种规律和经验模式,是从大量程序设计个体活动中抽象出来的、有意义的、有效的、基本通适的问题处理方法和思想。特殊性是指在普遍性指导下,个体认识活动的具体展开过程,是个体根据具体问题对共性规律进行灵活应用的过程和能力。特殊性可以映射为程序设计活动中每一个程序员的具体设计过程,该设计过程显然应该是在普遍性原则指导之下。

因此,程序设计的本质和内涵在于对两种文化及其思维框架的认识,从认识论的高度去理解程序设计。正确理解两种文化及其衍生的思维框架,以及由此诞生的各种技术和工具,并注重捕捉设计活动中的经验规律和应用模式,是程序设计的精髓。

图 1.3 演绎了程序设计的本质与内涵。从各种技术、工具出发,理解其背后隐藏的技术思想和原理,更进一步由各种技术思想感悟其背后的文化内涵及其思维逻辑特征,从而深刻体会和感悟程序设计的丰富内涵——编程之“道”。如此,就会使人以少胜多,降低学习新思想、新技术、新工具的成本,笑傲瞬息万变的 IT 世界;就会油然而生“会当凌绝顶,一览众山小”的专业顿悟,完成知“道”、体“道”、得“道”的过程。正如老子在《道德经》中所说:“为学日益,为道日损,损之又损,以至于无为,无为而无不为。”

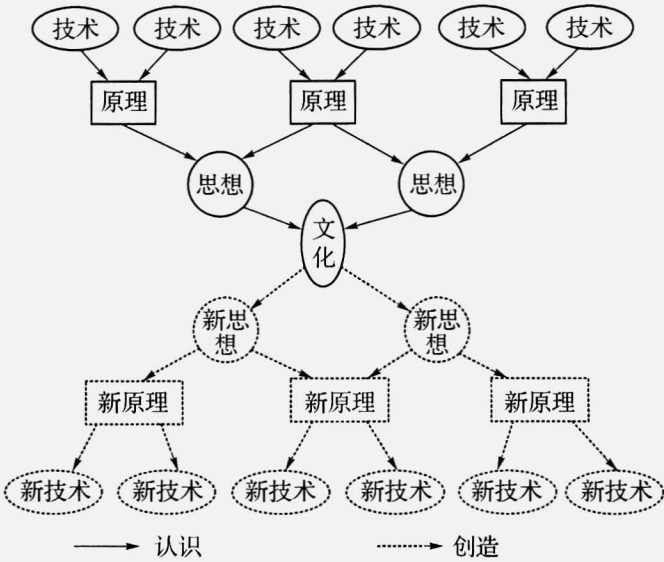


图 1.3 程序设计的内涵

1.5 本章小结

本章首先解析了什么是程序,什么是程序设计,以及两者的关系。然后,给出了程序设计的三个基本要素。最后,从认识论层面深层次解析了程序设计与思维和文化的关系,为如何有效学习程序设计给出一个铺垫。

习 题

- 1. 什么是程序? 对于一个具有 N 个处理步骤的“程”,其“序”有多少种? 其有效的“序”有多少种?
- 2. 什么是程序设计? 它与程序的关系是什么? 相对于程序,程序设计有什么特殊的地方? 程序与程序设计有什么本质不同?
- 3. 解析程序设计的三要素及其关系。
- 4. 解析程序设计与思维、文化的关系。
- 5. 对于程序的阅读理解和程序的构造,你认为两者之中哪个相对较难? 为什么?
- 6. 解析演绎思维和归纳思维的特征及其关系。