

论程序设计“非结构化→结构化→对象化”技术进步与教育创新

周启海

(西南财经大学经济信息工程系 成都610074)

On Technique Progresses and Blazing New Trails of Education for Program-Design's "non-structured→structured→objectized"

ZHOU Qi-Hai

(Southwestern University of Finance & Economics, Chengdu 610074)

Abstract In this paper, the programs' control structures are understood and re-understood, the program-design's "non-structured→structured→objectized(i. e. object-oriented)" are back-looked and known back, the program-design's education is evaluated, the distinctions and relations of the isomorphic nature and isomorphic representation of both an algorithm and its programs are researched and studied. Then, the creative new education ideas and new teaching-reform countermeasures of "the objectized conformity and isomorphized structured program-design" and "isomorphization of objectized program-design" are advanced.

Keywords Control structures, Structured program-design, Objectized program-design, Technique progresses, Teaching-reform, Blazing new trails of education

1 引言

计算机,是人脑的仿生与物化——“人脑的物化,物化的人脑”。

计算机语言,是人言的仿生与异化——“人言的异化,异化的人言”。

计算机程序设计(含算法设计),是人智的仿生与(模)拟化——“人智的拟化,拟化的人智”。

计算机程序设计教育,是人脑驾驭电脑的文明与文化——“人类新文明,世界新文化”。

这就决定了始于1946年的程序设计技术进步,必然对程序设计教育产生深远的根本影响;也就决定了从结构程序设计到对象程序设计(即对象化程序设计的简称;其旧称、习称为面向对象程序设计。注意:从程序设计发展史来看,采用术语“对象化程序设计”远比“面向对象程序设计”即 Object-Oriented programming,能更历史、真实、简练、精当地刻画出所谓“面向对象程序设计”的技术特色和历史地位;故作者倡导并只用此术语)的技术进步与教改创新,必定存在着深刻的本质联系。努力弄清、深入研究、科学总结这些历史影响和本质联系,将事关本质认识程序设计及程序设计教育的历史经验、发展规律,会对借鉴昨天、把握今天、探索明天的计算机程序设计教学改革与教育创新,影响深远,意义重大。

2 程序设计技术进步的认识和反思

半个多世纪来,程序设计技术进步不断发展、日益深化、渐趋完善,其进步标志是计算机控制结构的历史发展序列:非结构化→结构化→对象化。

2.1 控制结构的认识与再认识

算法(及其程序)设计离不开控制结构,反之亦然。今天,有必要进一步用21世纪的发展审视眼光,认识和再认识控制结构及其历史进步。

竖看历史,控制算法(及其程序)执行流程的控制结构,是控制算法(及其程序)执行流程的神经组织结构,也是程序设计及其教改发展知识体系的基本内容框架。进而,控制结构是传统控制结构与对象控制结构的全部总和。

传统控制结构,是顺序结构、选择结构、循环结构、子算法结构以及并行结构的总称。基本结构,是顺序结构、选择结构、循环结构的统称,是构成子算法结构、并行结构的基础。核心结构,是顺序结构、选择结构、循环结构、子算法结构的合称,是构成并行结构的核心。非顺序控制结构,是除顺序结构外的选择结构、循环结构、子算法结构、并行结构的别称,是比顺序结构复杂得多、也重要得多的传统控制结构,是人们学习、研究、应用的重点。

为了进一步简化算法(及其程序)设计、提高设计效率、保障软件安全,只采用传统控制结构是远远不够的(因为这只能支撑传统的结构程序设计,即结构化程序设计),还必须扩展到采用“对象”作为新型的控制结构(因为只有它才能支持现代的对象程序设计)。从根本上说,所谓对象控制结构就是:把算法(及其程序)中具有“各处(指位于任何处)同构(指其构造本质相同)”的行为(它对应着类的行为特性元素,简称行为元素),连同它们所处理的“结构同质”的数据(它对应着类的数据属性元素,简称数据元素),进行整体封装,并提升抽象为“各处通用、双重封装、可以继承、允许多(种状)态、可供复用”的一个“类”,且用“类”来定义和取代这些分离状初级形态的

*)本文得到教育部重点课题(课题批准号:DIA010319)与西南财经大学校管课题(课题批准号:02K05)资助。周启海 教授,硕士导师,教育部通讯评审专家,国家自然科学基金委同行评议专家,四川省有突出贡献优秀专家。主要从事软件设计方法学、软件设计自动化、中文信息处理、经济信息系统开发、计算机教育学、技术经济学等的教学与科研。

行为和数据的对象。这种具有“类决定对象,对象统帅行为,行为依托方法,方法处理数据,协同实现功能”集成控制特性的高级程序控制结构,称为对象控制结构。因此,应有“传统控制结构+对象控制结构=控制结构”。

由此可见:传统控制结构是对象控制结构的内核基础,而对象控制结构是传统控制结构的发展结果。类,把数据类型和传统控制结构(尤指类内的各子算法)封装在一起,具有继承性和多态性。传统控制结构与对象控制结构的同构化本质关系可抽象为“对象 \subset 类 \equiv 封装(数据类型,传统控制结构)+继承+多态”;式中,“封装(数据类型,传统控制结构)”表示“对数据类型和传统控制结构施行封装化处理”,“+”表示结合,“ \equiv ”表示恒等。

2.2 程序设计技术进步的回顾与反悟

20世纪中叶以来,控制结构的科学进化与历史演变,主导着程序设计及其教学教改发展历程“非结构化(Nonstructured)→结构化(Structured)→对象化(Objectized)”的三大历史性技术进步,反映了“传统控制结构与对象控制结构有着深刻的发展联系和进步必然”。

非结构化程序设计发祥最早,为计算机1946年的诞生与早期发展,建立了不可磨灭的历史贡献。但它滥用 GOTO 结构控制程序流程,也为日后令人震惊的“软件危机”埋下祸根。

20世纪60年代,为解决由非结构化程序设计导致的“软件危机”而提出的结构化和对象化程序设计思想,是程序设计发展史上最有影响的两大思想方法。

结构化程序设计发展较早、捷足先登,其方法精髓是“基于功能,面向数据,结构驱动”,宜于设计小规模程序。当年,它曾为解决软件危机立下汗马功劳。它可概括为公式:“结构程序设计=算法+数据结构+结构化软件方法学+结构化计算机语言”(式中,加号表示结合,等号表示统一),意即——结构程序设计是“算法、数据结构、结构化软件方法学、结构化计算机语言”四者有机结合的统一体。

对象化程序设计发展较晚、后来居上,是在结构程序设计基础上发展起来的,其思想核心是“基于(子)算法,面向对象,消息驱动”,长于中、大规模的程序设计。如今,它正为软件发展开辟广阔道路,已发展成为世界各国所公认的现代程序设计基本方法与主流方向。它可概括为公式:“对象=数据结构+(子)算法,对象程序设计=(对象+对象+…+对象)+对象化软件方法学+对象化计算机语言”,意即——对象,是“数据结构集及其作用于它的子算法集”的高级封装体;对象程序设计,是“活化(即处于激活状态的)对象群、对象化软件方法学、对象化计算机语言”三者有机结合的统一体。

应当强调指出:在对象程序设计中,对象的重心是“施加于封装在对象内部的对象数据(即对象的数据属性元素,简称数据元素)之上的对象行为(即对象的行为特性元素,简称行为元素)”;而用以描述、处理、实现对象行为的程序设计基本方法,其实质仍然是结构程序设计。由此可见,结构程序设计与对象程序设计的内在关系为“结构程序设计,是对象程序设计的底层基石;对象程序设计,是结构程序设计的上层建筑”。这表明,现今程序设计已发展成结构程序设计与对象程序设计的有机融合,人们绝不能错把两者割裂开来,更不能误使两者对立起来。

此外,也应当说明并遵循程序设计经济学原则:开发中、大规模程序,对象程序设计无疑具有结构程序设计无可比拟的巨大优越性。但在设计小规模程序时,不要拘泥于总想采用

对象程序设计;因为此时,这好比“杀鸡焉用牛刀”——对象程序设计原本胜过结构程序设计的技术优势将恰似“英雄无用武之地”,会表现得并不显著或难以发挥,使其时空效率、研制成本、生产周期有可能反不及结构程序设计而变得已不经济且低效率,这正如“打麻雀,用鸟枪比用导弹更合算、省事、快捷”。

不过,也应当说明并活用程序设计教育经济学原则:在对象程序设计基础普及教育中,为了简明、高效地讲述对象程序设计的基本原理、技术与方法,所采用的示例尽管往往为小规模示意性程序(因而解决它们并不是“非用对象程序设计不可”),但却特许对这些示例实行“例外原则”——不受上述程序设计经济学原则的约束,而一律采用对象化方法进行程序设计。

3 程序设计教育发展的评价与反思

纵观历史、环顾现状,程序设计教育发展尚较欠缺合理性、科学性、先进性,其所以如此的根本症结在于:对作为程序设计教育思想基础的计算机教育学,认识不足,重视不够,研究不深,未获突破。

实际上,计算机教育学是事关创新人才“以计算机程序设计为基础的计算机应用能力”基本素质培养的重要新兴交叉学科。但迄今为止,全球计算机科学家几乎无暇顾及而各国教育家又鲜给予关注的计算机教育学,远远落后于其母源学科——计算机科学与教育科学的发展进步,至今还较薄弱,不少方面还是空白。这就决定了国内外高等院校的程序设计课程,一直沿用“发展迟滞、默守陈规、缺乏创新”的落后教学模式(包括其教育思想、教学内容、教材建设、教学方法、教学方式、教学手段等),长期存在着三大根本弱点:

一是“移花不接木”——只着眼于把程序设计技术成果之花直接生搬硬套地简单移用作程序设计课程教学内容之木,而未能在教学中实现“从计算机科学的技术成果到计算机教育学的课程内容”的整合、转化、创新;因而,不能顺应高校学生的认知规律和满足师生的教学需求。

二是“授鱼不授渔”——仅局限于用某种当时流行的计算机语言(如 BASIC、PASCAL、FORTRAN、C、C++之一)展开程序设计教学,而未能在教学中揭示各种计算机语言程序设计的共性本质;因而,不能使高校学生掌握可指导各种计算机语言程序设计的共同原理,做到“举一反三”。

三是“叶落不知秋”——总拘泥于把面向过程的结构程序设计教学与面向对象的对象程序设计教学硬性割裂开来,分成互不相干的两部分(如国内外现行做法:不是分别作为各自独立的两门课程、两种教材,就是在一门课程、一本教材中作为各自独立的两大内容),而未能在教学中体现本应使结构程序设计与对象程序设计有机统一的客观规律;因而,不能使高校学生从对象化的科学视角来认识、学习和掌握结构程序设计与对象程序设计的辩证统一,浪费了高校教学资源,降低了国家教育效益。

正因为如此,程序设计教育发展虽长期以来总与程序设计技术进步结伴前行,但程序设计教学、教改、教育却远未能达到应有的合理性、科学性、先进性。这种历史与现状再也不能继续下去了!

4 结构程序设计的对象化整合与同构化创新

归根结底,程序设计无非是设计可解决给定问题的算法

及其程序。弄清算法与程序的关系,有助于认识从结构程序设计到对象程序设计的技术进步的历史必然与本质联系。

4.1 算法与程序的同构本质与异构表现

算法,是人们为了借助计算机解决给定问题,而模拟计算机按照“人一(计算)机”系统所认可的操作方式和控制方式,具体施行的有穷操作的有序集合的准自然语言描述与亚计算机语言表征。

程序,是人们在已设计出的算法基础上,进一步用选定的某种计算机语言,把算法译成计算机可接受、理解和执行的有穷操作的有序集合的计算机语言描述。众所周知,同一个事物,可用不同自然语言表述而有不同说法;同理,同一个算法可根据设计者实际需要和编程爱好而采用不同的计算机语言进行编程,并得到“同构(于同一算法本质)异形(出不同语种形式)”的不同程序。

1986年,笔者提出“凡同类事物,客观上总存在着使其结构本质与构造实质趋于或达到一致的自然规律”的同构化基本原理;2000年,提出“凡异类事物,客观上总存在着使其结构本质与构造实质趋于或达到不同的自然规律”的异构化基本原理;2001年,提出“凡事物,客观上既存在着使其结构本质与构造实质趋于或达到一致的同构化自然规律,也存在着使其结构本质与构造实质趋于或达到不同的异构化自然规律”的特构化基本原理。

按照同构化基本原理与异构化基本原理,算法显然远比程序深刻得多、根本得多、重要得多,且两者的同构化关系是“算法是程序之母,程序是算法之子”,而两者的异构化关系是“算法是程序的同构化之根,程序是算法的异构化之叶”。

4.2 结构程序设计的对象化整合

结构程序设计,因它对现实事物数据属性和行为(或称操作、服务)特性的割裂性描述和处理,存在着自身难以克服的重大缺陷:造成数据结构对其算法的被动性,形成算法囿其数据结构的局限性,导致大中型软件设计的隐患性。因而,必须对它进行深刻改造与重大革新。

正是在结构程序设计基础上的深化革新,才取得“基于算法,面向对象,消息驱动”的程序设计大进步:催生了数据结构、算法从彼此分离走向深层封装与结合的崭新数据类型——“类”,促成了可灵活组装程序的基本元件——“对象”,铸就了对象、对象程序设计方法学、对象化语言所凝成的当今主流程序设计技术——对象程序设计:“对象=数据结构+算法,对象程序设计=(对象+对象+…+对象)+对象程序设计方法学+对象化语言”;式中,括号表示优先,加号表示结合,等号表示统一。

不同于结构程序设计倚重传统控制结构——“在顺序结构、选择结构、循环结构、子算法结构(以及并行结构)的控制下,实现程序功能”,对象程序设计注重对象控制结构——“在消息驱动对象、对象作出响应(即对消息进行释义和处理)的调控下,实现程序功能”,揭开了程序设计发展的历史新篇章。但应强调指出:在对象程序设计中,封装在对象内部的用以描述、处理、实现对象行为属性的程序设计基本方法,其实质仍然是结构程序设计。

据此,人们可以而且应当把结构程序设计所倚重的传统控制结构,科学地、完全地、自然地纳入对象程序设计的理论框架与设计规范。

如果说,循环结构与子算法结构是结构程序设计最常用、最关键的基本概念;那么可以说,类与对象就是对象程序

设计最重要、最基本的核心概念。从软件方法学来看,类及其对象是其数据元素与行为元素的模块化集成体;而类的对象化技术支柱,则是封装(Encapsulation)、继承(Inheritance)、多态(Polymorphism)三大技术,其主导思想主要是“重用”(Reuse)。

封装技术,既是一种高强度的信息隐蔽技术,也是一种模块集成化的程序代码动态重用技术。它把类的数据元素、行为元素封装成一个整体来作“黑箱处理”:在行为元素的内核实现模块中,使类元素的抽象行为功能(力求单一)与其具体实现方法(可以多样)得以分开;在类整体化的外壳集成模块中,使类的静态定义与动态表现(指“接收消息、作出响应、表现为、实现功能”)能够分开。故封装使类及其对象的行为元素得到“外壳层”与“元素层”的双重封装和强化保护,极大地避免了外部对类及其元素的可能侵扰,从而增强了类及其元素处理的安全性、可靠性和易维护性。

继承技术,是一种程序代码静态重用技术。它表达了类与类之间的层次关系——父类可以派生出子类;子类既能遗传得父类若干元素,也可变异出自己的若干元素。故子类可重用(注意:这是重复利用,而不是重新编写!)处理这些遗传元素的程序代码,只需另行处理那些变异元素编写程序代码即可,从而提高了软件开发效率。继承技术有利于采用快速原型法进行软件开发,有助于用类及其对象的可重用元素构造或扩充软件系统。

多态技术,既是一种信源(即信号源,例如:运算符、对象、消息)自适应强化技术,也是一种程序代码自适应动态重用技术。它是使同一信源及其处理程序代码,在不同情况下可有释义与处理均不同的响应。它是通过静态的重载(Reloading)技术与动态的绑定(Dynamicbinding,也称联编、结合)技术来实现,从而降低了软件开发难度和成本。

值得注意的是:支撑对象程序设计的这三大技术支柱,无论从它们所封装、继承、多态化处理的重用内容,还是实现它们自身的内核设计方法,仍然都或多或少地离不开结构程序设计。

事实上,从同构化程序设计高度来审视:在结构程序设计中,循环结构只是“同处同构”操作块的最佳分立控制结构,子算法结构已是“异处同构”操作块的最佳分立控制结构;而在对象程序设计中,类与对象则进一步优化为“各处同构”数据结构及其上各子算法的最佳集成控制结构。

因此,类的技术本质,是一种崭新的集成控制结构,是计算机系统中“源数据结构及其上所有子算法(包括过程子算法和函数子算法)”的集成化封装控制手段:类的数据元素,就是在类中定义的供本类内外使用的“类辖变量”,其本质是结构程序设计中记录字段的推广;而类的行为元素,乃是在类中定义的供本类内外调用的“类辖子算法”(包括:类辖子算法的“首部”,及实现其行为功能的“方法”),其本质是结构程序设计中普通子算法的强化和升华。

当然,也应指出:从形式和作用看,对象程序设计中下使用消息,类似于结构程序设计中调用子算法,但有三大区别。

第一、对象程序设计中下使用消息,所调用的是类辖子算法;而结构程序设计中调用子算法,所调用的是自立子算法。

第二、子算法总得“对子算法调用有求必应”,而对对象则可以“对消息使用不作响应”。

第三、调用子算法时的参数相同,其子算法调用结果应当相同;而使用消息时的接口消息相同,其消息响应结果未必相

同。

由此观之,对象程序设计是结构程序设计的改革性继承、创新性发展;而两者的同构化关系为“结构程序设计,是对象程序设计的底层基石;对象程序设计,是结构程序设计的上层建筑”。这表明,今天的程序设计,实际上是结构程序设计与对象程序设计的有机融合、新型统一。显然,任何把两者割裂开来甚至对立起来的观点和做法,在理论上是错误的,在实践上是有害的。因此,作者首创的“把结构程序设计有机融入对象程序设计”的新观点、新思路、新方法,理所当然地是人们学好、用好对象程序设计的新捷径。

5 程序设计教育创新的对策构想

程序设计普及教育、教学改革与教育创新的长期实践表明:如果未能反映程序设计“非结构化→结构化→对象化”的技术进步与教改发展,如果未能体现“结构程序设计”与“对象程序设计”历史协同和教育创新的客观要求与发展规律,就必定不能实现“结构程序设计对象化与对象程序设计同构化”的有机融入、科学结合、和谐统一,就肯定不能使受教育者事半功倍、举一反三。

在对策核心理念上,笔者坚持“凡事物,客观上既存在着使其结构本质和构造实质趋于或达到与同类事物相一致的同构化自然规律,也存在着使其结构本质和构造实质趋于或达到与异类事物不一致的异构化自然规律”的创新研究独特视角,针对国内外程序设计课程教学现行模式三大弱点,展开对程序设计课程教育思想、教学内容、教材建设、教学方法、教学方式、教学手段的创新研究,依托和反映“教育思想、教学内容、教材建设”的理论研究与“教学方法、教学方式、教学手段”的实验实证有机结合的计算机程序设计教学创新研究:

——针对“移花不接木”弱点,笔者坚持“科技上合理的事物未必在教育上也同样合理”的唯物辩证观,深刻认识“计算机教育学=计算机科学+教育科学”的交叉学科特色,正确运用认知论、课程论、教学论相结合的研究方法,认真研究程序设计技术与程序设计教学的根本区别、内在联系、交叉融合与学科转化,积极探索从程序设计技术历史成就向程序设计教学现实需要的交融创造和转化创新,催生“程序设计教育学=程序设计科学+教育科学”的课程论重组整合、教学论移植转化、学科性交叉改造,以顺应高校程序设计课程教学中的学生认知规律和师生教学需求。

——针对“授鱼不授渔”弱点,笔者坚持“特殊与一般、共性与个性”的辩证统一观,探索和揭示寓于各种计算机语言程序设计中的、可指导各种计算机语言程序设计的共性本质和同构原理,并进而创著出“思想先进、内容新颖,深入浅出、循序渐进,富于启迪、举一反三,便于教学、利于教改”的新一代程序设计创新教材,以适应科技进步与终身教育所要求的“举一反三”。

——针对“叶落不知秋”弱点,笔者坚持“今天的程序设计教学内容、教材架构,绝不是程序设计进步历史轨迹的简单克隆、剪裁拼盘、直接翻版,而应是更深化的再认识与再创造”的

历史统一观,研究和探求可使结构程序设计自然、合理、科学地融入对象程序设计的创新方法和认知捷径,使高校学生能直接从对象化的统一视角来深刻认识、快捷掌握结构程序设计与对象程序设计,以提高高校师生教学效能与学生创新能力。

总之,笔者坚信:以上述基本对策构想为“结构程序设计的对象化整合与对象程序设计的同构化创新”教学教改与教育创新的灵魂,其成果必将填补国内外程序设计教育学研究空白,提高我国高校程序设计课程的教学质量、水平,提升我国高等教育(特别是计算机程序设计基础教育)的国际地位、学术影响,促进高校学生创新意识、能力的培养,进一步丰富和深化现代课程论、教学论的研究、发展和创新;因而,无疑有重要的社会、经济价值与理论、现实意义。

参考文献

- 1 Zhou Qihai. On the Improvement of Flowchart Techniques for Teaching of Computer Basic Course. in THE Asia- Pacific Conference On Computer Education, Oct. 1988
- 2 周启海. 走中国特色的计算机普及教育改革之路. 软件报, 1988 (39)
- 3 周启海. 计算机基础教育改革的思路与实践. 教材通讯, 1989(4)
- 4 周启海. 计算机结构程序设计原理. 北京: 高等教育出版社, 1989
- 5 Zhou Qihai. An Improved Graphic Representation for Structured Program Design. in Journal of Computer Science & Technology, 1991, 6(2)
- 6 周启海. The New Shortcut of Instruction for Computer Program Design. 见: 中国计算机学会教育与培训专业委员会1991年年会论文集, 1991
- 7 周启海. 既授人以鱼, 更授人以渔(等16篇, 系列讲座文章). 计算机世界(周报), 1992(1~4)
- 8 周启海. 计算机同构化程序设计原理及其应用导论. 北京: 清华大学出版社, 1993
- 9 周启海. Zhou 图——同构化系统设计表现技术. 计算机学报, 1995 (7)
- 10 周启海. 试论中国计算机基础普及教育改革若干问题. 见: 第三届全国计算机应用学术交流大会论文集. 北京: 电子工业出版社, 1995
- 11 周启海. 谈谈开创性的同构化程序设计. 计算机科学, 1997(特辑)
- 12 周启海. 世纪之交计算机基础教育教材建设的若干启示. 全国高校计算机基础教育研究会2000年论文集. 北京: 清华大学出版社, 2000
- 13 周启海. C语言程序设计新捷径. 上海: 复旦大学出版社, 2000
- 14 周启海. 论结构程序设计与对象程序设计的同构化关系. 见: 中国计算机学会2001全国软件技术研讨会论文集. 大连: 大连出版社, 2001
- 15 周启海. 21世纪高校计算机程序设计教育及其信息化技术的创新思考. 见: 第八届联合国际计算机会议论文集. 杭州: 浙江大学出版社, 2002
- 16 周启海. An Isomorphized New Thinking for Automatic Program Design and Program Transplant. 见: 第八届联合国际计算机会议论文集. 杭州: 浙江大学出版社, 2002