

以培养计算思维为目标的表达式教学研究与实践

张化川, 陈 君, 张 悦, 张力生

(重庆邮电大学软件工程学院, 重庆 400065)

摘 要: 表达式是程序设计语言的核心, 涉及计算机中很多基本概念和方法。但在目前的教学中, 这些内容并没有很好地被融入, 导致大部分学生没有从计算机层面理解表达式, 最终影响学生对程序设计和编程的兴趣。依据图灵机、冯·诺依曼机、上下文无关文法等计算理论及逆波兰式等编译方法, 融合程序设计和计算机运行实现中的内容, 文中设计了确定运算符求值顺序的递归算法和表达式求值的循环算法, 以CDIO中“做中学”教育理念为指导, 设计了对这两个算法的纸上演算学习方法。依此方法, 学生在演算的过程中不仅掌握表达式及相关程序设计的知识, 还有效培养程序思维、递归思维, 以及“0和1”思维等计算思维。

关键词: 计算思维; 计算机表达式; CDIO

中图分类号: TP311 **文献标识码:** A

Research and Practice on Expression Teaching Aiming for Cultivating Computational Thinking

ZHANG Huachuan, CHEN Jun, ZHANG Yue, ZHANG Lisheng

(School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: Expression is the core of programming language, which involves many basic concepts and methods in the field of computer. However, in current teaching environment, it is not applied well so that a large portion of students cannot comprehend expressions from the level of computer implementation, leading to lack of interests in the study of programming. Based on calculation theory of Turing machine, Von Neumann machine, context-free grammar, and reverse Polish notation, the paper integrates the knowledge of programming with that of computer operation, and then designs a recursive algorithm to determine the order of operators and a cyclic algorithm to produce the result of expression. Besides these two algorithms, a teaching method is proposed based on the education concept of Learning by Doing in CDIO. In such way, students can learn these algorithms by calculating on paper. In this approach, students can not only master programming skills, but also develop their computational thinking effectively.

Keywords: computational thinking; expression teaching; CDIO

1 引言(Introduction)

工程人才短缺和工程教育质量问题是世界面临的共同课题^[1]。计算机专业的教育是工程教育, 计算机课程的教学质量, 特别是计算机基础课程的教学质量直接影响计算机工程人才培养质量。在我国, 高校计算机基础课程普遍存在告诉学生“这是什么”, 而不是“为什么是这样”的现象^[2]。计算机基础教学面临着新的改革, 既要有相对稳定的、体现计算机学科思想和方法的核心内容, 又要突出思维方法的训练^[3]。在2010年九校联盟(C9)计算机课程研讨会上, 与会专家就“如何在新形势下提高计算机基础教学的质量、增强计算思维能力的培养”, 明确提出将“计算思维”作为计算机基础教学的核心任务^[4]。

在计算机基础教学中, 计算机语言是其中的重要内容, 而要学好计算机语言, 学好计算机表达式是基础。因此, 本文从计算机表达式的教学改革入手, 探讨计算机基础教学中如何落实计算思维能力的培养。

2 计算机表达式(Computer expression)

计算机表达式是程序的重要组成部分, 它来源于数学表达式, 是数学表达式在计算机中的应用; 然而与数据表达式又不尽相同, 它更侧重的是数据在计算机中的表示和存贮方法、实现运算的方法、构建和分解表达式的算法等应用内容, 学习重点是实现方法及计算机基础理论和知识。

2.1 数据的表示和存贮

根据冯·诺依曼机理论, 计算机在处理数据时, 先将数

据存贮到内存中,为了使用方便,从数学中引入了变量的概念,标识数据对应的二进制串存放在内存中的位置。从本质上讲,变量表示内存中一组连续的存贮单元,用首地址和偏移表示其在内存中的位置^[5]。图1描述了数据在计算机中的表示和存贮,为了便于初学者理解,用“长度”替换了“偏移”这个计算机术语。

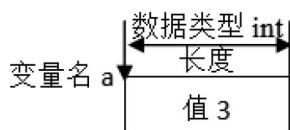


图1 int a=3在计算机中的表示和存贮

Fig.1 The representation and storage on int a=3

2.2 运算

在数学中定义了数的集合,然后在数的集合上定义运算,并用一个符号表示这个运算,这个符号称为运算符,运算涉及的数称为操作数。

计算机运算符来源于数学运算符。为保证计算机能存贮和处理数据,计算机对不同类型的数据使用不同的二进制表示方法,从而产生了数据和数据类型的概念。计算机对数据进行运算表示的是计算机处理数据的操作序列,即在冯·诺依曼机上处理数据的操作序列。

从本质上讲,运算是定义在数据类型上的,与数据类型紧密相关,虽然有些运算可以处理不同类型的数据,但具体实现的方法是不同的。若操作数的数据类型与运算规定的的数据类型不同时,先要对操作数进行类型转换^[6]后再进行计算。因此,必须有数据类型的概念和意识才能够理解计算机中的运算。初学者需要进行大量的训练,才可以建立这种概念和意识。

2.3 教学和学习重点

前文已经明确从表达式和运算两个层次学习计算机表达式。在表达式层次上,核心内容是构造一个需要的运算序列,学习重点定位于掌握确定表达式中运算的执行顺序的方法,着重培养计算思维^[7]中的递归思维和程序思维。在运算层次上,核心内容是构造一个能在冯·诺依曼机上运行的操作序列,由于计算机中增加了很多计算机特有的运算,涉及到大量的计算机知识,以及计算理论中的原理和方法,初学者不可能一次性学习并掌握所有的运算,所以学习重点定位于掌握在冯·诺依曼机上执行运算的方法,学习和掌握基本、常用的运算,提升学习能力,着重培养计算思维^[7]中的“0和1”思维。

3 计算机处理表达式(Processing for expressions by computer)

培养和推进计算思维需要学生深入掌握计算机解决问题的思路^[8]。如图2所示,描述计算机中编译表达式和计算表达

式的过程,在此过程中体现了计算基本原理和方法。

3.1 编译过程

对计算理论中的下推自动机、编译原理中的逆波兰式等基础理论和技术进行分析,总结抽象出计算机编译表达式的递归过程,并形成流程图如图2所示。

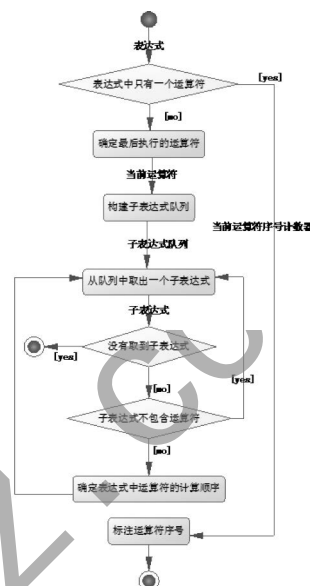


图2 计算机编译计算机表达式的过程

Fig.2 The process of compiling computer expressions

(1)若表达式中只有一个运算符,直接将序号计数器(序号计数器初始值为1)中的序号标注给这个运算符,序号计数器加1。

(2)确定最后执行的运算符。表达式中有多个运算符时,找到优先级最低的运算符,若有多个优先级最低的运算符,根据优先级最低的运算符的结合性确定最后执行的运算符。

(3)构建子表达式队列。如果在第(2)步确定的运算符是多目运算,则按照规定的操作数计算顺序或编译器规定的计算顺序构建子表达式列表;若是单目运算,则直接构建子表达式队列。

(4)从子表达式队列中取出一个子表达式。采用递归方法依次确定各子表达式的运算符执行顺序。从队列中取出一个子表达式,如果没有子表达式,则计算机编译完毕。如果取出的子表达式不包含运算符,继续从队列中取下一个子表达式;如果取出的子表达式包含运算符,则递归调用本算法,确定子表达式中运算符的顺序。

3.2 计算过程

根据图灵机及冯·诺依曼机等理论可知,CPU采用循环方式执行程序的指令序列。为了便于理解,使用如图3所示的循环算法,描述计算机计算表达式的过程。按照上节确定的运算符计算顺序,从第1个运算符开始,依次执行所有的运算符,直到结束。

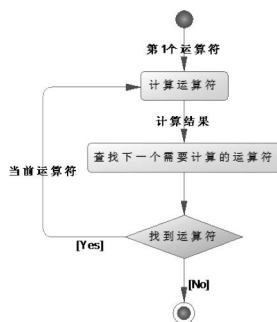


图3 描述计算机表达式的过程

Fig.3 The process of describing the computer expression

(1)计算运算符。运算符的语义规定了在冯·诺依曼机上运行的操作数序列，无论运行在什么类型的计算上，CPU执行的指令序列都是为了实现冯·诺依曼机的操作数序列。根据运算符的语义将操作数进行计算，得到计算结果。

(2)查找下一个需要计算的运算符。根据上级运算符的语义和前面的计算结果，依次判断下一个运算符是否需要计算。

(3)判断是否找到了需要计算的运算符。若找到，则返回第一步，计算运算符；若未找到，则遍历了运算符序列。

计算表达式结束后，最后计算的结果即为表达式的计算结果。

4 表达式的教学实践(Practice on expression teaching)

对初学者而言，可按照类似数学演算的方法，在纸上分编译表达式和计算表达式两步对表达式在计算机的处理过程进行演算，有效地将数学中的学习方法与能力平稳迁移到程序设计的学习中，降低学习难度，提高学习效率。

4.1 运算符

目前大多数教材虽涉及了运算符的语法、语义、优先级、结合律等运算相关的内容，有些教材^[6]也给出了计算机表达式中运算符的运算表，但大多数教材都没有对运算符的语义进行准确描述，对运算符的优先级和结合性也只有简单的描述，没有深入讲解，导致学生很容易对运算符理解错误。如有些教材中出现的“按运算符的优先级别高低次序执行”^[9]的表述，导致学生误解为计算机一定先执行优先级高的运算符，再执行优先级低的运算符。

目前大多数教材，无论是内容的完备性还是组织形式，都不足以支持学生以计算机的方法计算表达式。针对这一现状，对相关知识进行梳理，设计了包含运算符的运算名、结合性、语法和语义等的运算表，如表1所示。

在表1的运算表中只列举了部分运算符的相关内容，表中运算符按优先级由高到低的顺序依次列出，若优先级是相同的，则用相同的底纹显示。只有逻辑运算、条件运算符和逗号运算符等少数运算符规定了计算顺序，其余运算符的计算顺序由编译器决定，或从左至右，或从右至左，但在同一个编译器中，运算符的计算顺序是唯一确定的。为了便于讲解，在本文中，除了已经规定了操作数计算顺序的运算符，其余运算符统一为从左至右的计算顺序。

表1 运算表

Tab.1 Operation table

运算符	运算名	结合性	语法	语义/操作序列
()	Bracket	Left to right	(exp)	提高表达式exp中运算符优先级
*	Multiplication	Left to right	exp1*exp2	计算exp1得到值v1，计算exp2得到值v2，v1乘以v2得到值v3
/	Division	Left to right	exp1/exp2	计算exp1得到值v1，计算exp2得到值v2，v1除以v2得到值v3
%	Modulus	Left to right	exp1%exp2	计算exp1得到整数类型的值v1，计算exp2得到整数类型的值v2，v1除以v2取余数整数类型的值v3
+	Addition	Left to right	exp1+exp2	计算exp1得到值v1，计算exp2得到值v2，v1加v2得到值v3
-	Subtraction	Left to right	exp1-exp2	计算exp1得到值v1，计算exp2得到值v2，v1减v2得到值v3
=	Assignment	Right to left	exp1=exp2	计算exp1得到变量x，计算exp2得到值v，将v按照变量x类型规定的方式存储到变量x指定的内存，得到变量x

学生在对一个实际的表达式进行分析时，首先确定运算符的计算顺序，然后再根据确定的运算符计算顺序对表达式进行求值，具体步骤如下。

4.2 确定运算符的计算顺序方法

将表达式抄到一张纸上，表达式一定要写在同一行，且下面需要有足够的空行，然后按照如下步骤，确定表达式中运算符的计算顺序。

(1)在表达式下面画一条横线，表示要计算这个表达式。如果表达式只有一个运算符，则直接在运算符上面标注运算符的计算序号。

(2)如果表达式有多个运算符，根据优先级和结合律找到最后计算的运算符。从横线标注的表达式中提取所有运算符，然后在运算表中逐个查找这些运算符的优先级，找出最低优先级的运算符，若只有一个，则此运算符就是最后计算的运算符；若有多个，则根据结合性找到最后计算的运算符。在最后计算的运算符下面画一条竖线，并与横线垂直相交，以表示该运算符，并将横线标注的表达式分解为多个子表达式。

(3)确定子表达式列表顺序。在运算符表中查询当前运算符的形式化语法，确定形式化的“exp”代表的实际子表达式，按表中的操作数运算顺序，确定子表达式的有序列表。

(4)采用递归方法依次确定各子表达式中运算符的计算顺序。按照子表达式列表的顺序依次取出子表达式，找到此子表达式中最后计算的运算符，在运算表中找到此运算符的形式化语法，确定形式化的“exp”代表的实际子表达式，按表中规定的操作数运算顺序，形成另一个子表达式有序列表。若此子表达式除了最后计算的运算符再无其他运算符，则顺序取出子表达式列表中的另一个子表达式，确定运算符的计

算顺序。

(5)直到表达式中所有的运算符标上都执行序号。

表达式 $x=c*f+(f/u-i\%3)c$ 、 f 、 u 、 i 分别为char、float、unsigned、int类型的变量，按照上面的方法最终确定的运算符计算顺序如图4所示。更详细过程可参照文献[10]。

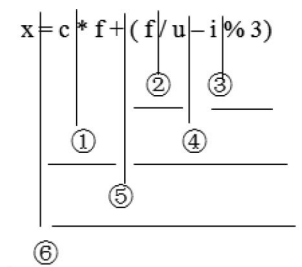


图4 确定 $x=c*f+(f/u-i\%3)$ 运算符的计算顺序

Fig.4 The calculation order on $x=c*f+(f/u-i\%3)$

4.3 表达式求值的演算方法

在上图中，根据以下步骤直接演算表达式求值过程。

(1)根据画出的运算符执行顺序图，找到未执行序号中，执行序号最小的运算符。

(2)计算执行序号最小的子表达式。根据运算符表中给出的运算符形式化语法，找到“exp”对应的实际数据，将实际数据代入到对应的语义中替换“exp”，替换后根据语义进行计算，得到子表达式①的计算结果。若运算的操作数不满足对数据类型的要求，还需要增加类型转换操作，再进行计算，最后将计算的值和数据类型写在执行序号下面的横线处。

(3)根据运算符的语义和前面的计算结果判断是跳过下一个运算符继续查找运算符，还是计算这一运算符。当找到了需要计算的运算符，循环第(2)步。

(4)当没有了需要计算的运算符时，表达式计算完毕，最后一个被计算的运算符下的数据类型和值即为最后的计算结果。

表达式 $x=c*f+(f/u-i\%3)$ 按照上面的方法对表达式求值的顺序如图5所示。

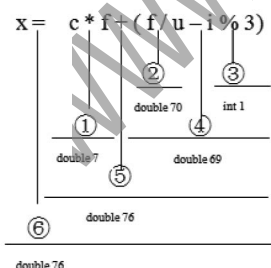


图5 表达式 $x=c*f+(f/u-i\%3)$ 的计算过程

Fig.5 The calculation process on $x=c*f+(f/u-i\%3)$

需要注意，确定表达式的运算符计算顺序并不是表达式求值的顺序，在有些表达式中，会直接跳过一些运算符的执行顺序对表达式求值，如表达式 $c=c>='A' \&\&c<='Z'?c+32:c$ 。

5 结论(Conclusion)

本文基于CDIO工程教育思想，以培养计算思维为目标，

对计算机表达式提出教学改革的方法。从图灵机、冯·诺依曼机、上下文无关文法等计算理论，以及逆波兰式等编译方法中，总结、抽象出确定运算符执行顺序的递归算法和按顺序计算运算符的循环算法，并根据教学的需要，设计出确定运算符的计算顺序方法和表达式求值演算方法，设计出与演算方法配套的包含形式化语法和半形式化语义等内容的运算符表。在计算机存储数据的教学中培养了学生的“0和1”思维；由于在计算机处理表达式的教学中使用了递归思想进行分析，所以学生在纸上演算推理表达式的过程中培养了递归思维，达到了“做中学”的目标；同时，在学生分析计算机处理表达式中理解计算机执行的过程，即培养学生的程序思维。

该教学改革方法于2010年应用于我校软件工程专业，经过8年多的教学实践证明，学生自主学习程序设计的时间明显增加，能读懂数据结构中算法的学生比例，以及具有较好编程能力的学生比例明显提高，近五年麦可思中国大学就业报告显示，我校软件工程专业毕业生的就业质量处于重庆市前列。

参考文献(References)

- [1] 李竹林.基于CDIO教育理念的高校计算机专业课程教学改革探讨[J].教育与职业,2012(20):127-128.
- [2] 周虹,富春岩,刘越.大学计算机基础课程的创新与实践[J].教育探索,2014(3):39-40.
- [3] 冯博琴.对于计算思维能力培养“落地”问题的探讨[J].中国大学教学,2012(9):6-9.
- [4] 佚名.九校联盟(C9)计算机基础教学发展战略联合声明[J].中国大学教学,2010(9):4-9.
- [5] Bjarne,Stroustrup.王刚,译.C++程序设计原理与实践[M].北京:机械工业出版社,2010.
- [6] 钱能.C++程序设计教程(第二版)[M].北京:清华大学出版社,2005.
- [7] 陈国良,董荣胜.计算思维的表述体系[J].中国大学教学,2013(12):22-36.
- [8] 谭浩强.研究计算思维,坚持面向应用[J].计算机教育,2012(21):45-49.
- [9] 谭浩强.C++程序设计[M].北京:清华大学出版社,2004.
- [10] 张化川,杨林,黎在万.基于计算思维的计算机表达式教学方法实践[J].现代计算机,2017(05):7-10.

作者简介:

张化川(1979-),男,硕士,实验师.研究领域:计算机基础教学,软件建模及测试方法.

陈君(1981-),女,硕士,讲师.研究领域:软件工程,数据挖掘.

张悦(1993-),女,硕士生.研究领域:软件工程.

张力生(1965-),男,硕士,教授.研究领域:计算机基础教学,数据挖掘与大数据分析.