



例谈学生计算思维培养的教学策略

□ 孙秀芝 胡泉镐

【摘要】逻辑不清、脱离算法、思维受限是“算法与程序设计”模块初学者身上的常见问题,主要原因在于其对算法的关键步骤、程序的关键指令、作品的关键差异把握不准。实践证明,通过梳理关键步骤、寻找关键指令、聚焦关键差异三个教学策略,能够改善课堂状况,促进学生计算思维的培养。

【关键词】计算思维;算法与程序设计;教学策略

计算思维是个体运用计算机科学的思维方法来解决一系列思维活动,是信息技术学科核心素养的四个核心要素之一。随着教材的再次修订,体现信息技术学科本质内容,指向计算思维培养的知识模块的比重大幅提升,这也给教师开展算法与程序设计教学带来了更大挑战。在实际教学中,逻辑不清、脱离算法、思维受限是“算法与程序设计”模块初学者身上的常见问题,主要原因在于其对算法的关键步骤、程序的关键指令、作品的关键差异把握不准。

鉴于此,教师从“算法与程序设计”模块教学的三个关键点入手,通过梳理关键步骤、寻找关键指

令、聚焦关键差异,采用合适的教学策略改进当前课堂现状,帮助学生逐步养成计算思维。

一、梳理关键步骤,培养分解思维

对于算法的关键步骤,没有厘清其逻辑结构,是导致分析问题思路混乱的主要原因。虽然教师会列举生活中学生耳熟能详的案例,但是学生在独立思考时还是会遇到障碍。因此,教师需要引导学生对问题进行分解,抓住解决问题的关键步骤,并用自己的理解加以描述,以此使思维更清晰。

(一)要素分解,用自然语言描述关键步骤

对于逻辑思维处于发展阶段的初学者来说,让

教师可以与学生一起讨论活动计划,让学生来选择活动开展的方式、内容以及小组同伴。同时,活动的过程推进要体现全域育人。活动过程是学生情感、态度、价值观、学习能力、团队协作水平等因素全方面展现的过程。关注学生在活动过程中的表现不仅可以给教师了解学生提供翔实的方向,还为教育学生,使学生在原有的基础上有所提升提供了充分的机会。活动中,要让个体差异融入整体优化过程。教师应从学生的日常生活中捕捉有教育价值的内容,兼顾群体需要和个体差异,根据学生的

实际表现提出积极的、有针对性的建议,调动他们的学习主动性,让学生获得成就感,增强自信心。

参考文献:

[1]杨梅.小学道德与法治可以这样教[M].长春:东北师范大学出版社,2019.

[2]达蒙.儿童道德心理学[M].上海:上海社会科学院出版社,2020.

[3]杜时忠.德育研究[M].福州:福建教育出版社,2019.

(浙江省杭州市勇进实验学校 310016)

其分析计算机程序的算法并不容易,即便是相对简单的程序也是如此。因此,教师在引导学生分析一个具体问题的算法时,要把握两个要点:一是分析具体问题中包含的主要角色;二是将主要角色所要进行的关键步骤进行罗列,而对于次要信息予以暂缓考虑。分析过程可以借助一定的可视化工具来支持,比如思维导图、表格等,但学生须用自己的语言来表达解决问题的思路。

【案例1】《询问交互》一课算法分析

课堂上,教师提供了《中国诗词大会》的视频片段,请学生利用表格分析一次问答程序经历的步骤。(提示:只考虑关键交互过程)具体如表1。

表1 一次问答的关键步骤表

角色		
关键步骤	计算机系统角色	用户
1	询问问题	
2		输入回答
3	判断用户回答与答案是否一致	
4	如果一致,输出答对提示;否则,输出答错提示	

学生通过反复观看视频,理解计算机与用户交互的过程,在表格中罗列出目标程序所涉及的两个角色——“计算机系统角色”“用户”。由于“用户”就是使用者本身,不需要在计算机中设置,所以只需要添加一个角色——“计算机系统角色”。接着,学生对于实现目标程序所需要的关键步骤进行分解:第1步是“计算机系统角色”出题,第2步是“用户”输入回答,第3步是“计算机系统角色”判断用户回答是否与答案一致,第4步是“计算机系统角色”根据判断结果输出不同的结果提示。

通过主要角色、关键步骤的分析训练,学生初步学会了抓住问题的核心,将主要精力集中于解决程序中最重要“交互”问题,将次要问题(角色的初始化、程序的提示语、结果提示时的角色造型变化等)的处理调整到后续优化环节,以此保证课堂的高效性。

(二)图形填空,用流程图表示基本的算法

用流程图描述算法,能更形象、直观地表示算

法,便于学习者理解,也便于指导学习者进行程序的编写。在学生初学流程图这一描述方式时,教师依然需要引导学生抓住关键步骤,以此降低学习难度,从而帮助学生顺利理出一个相对完整的流程图。教学时,为了引导初学者入门,教师可以借助图形填空的方式降低难度,让学生将关键步骤组合成程序的流程图。

【案例2】《询问交互》一课算法分析

仍以问答程序中的算法分析过程为例,教师为学生提供学习单,请学生将右侧的步骤拖到左侧的线框内,完成一次问答程序的流程图,如图1。

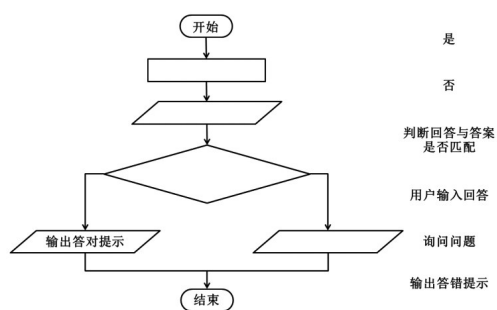


图1

借助这样的半成品流程图,学生参与绘制的难度大大降低,绝大部分学生都能顺利完成任务。在拖动的过程中,学生一边理解算法,一边熟悉流程图符号。

实践证明,利用图形填空的方式,能更有效地帮助初学者掌握利用流程图表示算法的方法,帮助学生建立对程序关键步骤的理解。有了程序流程图,程序的编写就更容易操作。需要注意的是,教师在设计学习单时,可以有意识地将流程图的关键步骤进行“挖空”处理,从而引起学生对关键步骤的关注,为后续教学关键指令做好铺垫。

二、寻找关键指令,提升算法思维

初学计算机课程的学生尚未建立流程图与程序之间的对应关系,对于流程图中的关键步骤使用何种关键指令存在认知上的困难,从而导致程序编写的过程脱离流程图。因此,教师需要帮助学生在程序流程图与计算机程序的各条指令间建立联系,将算法真正落实到程序的实际脚本当中。

(一)正向匹配,经历关键指令的探究过程

随着算法越来越复杂,程序设计中所涉及的新指令也随之增多。新指令的学习是除了算法分析外的另一个教学难点。因此,教学中,教师需要把握两个要点:一是要给予学生充分的时间学习新指令,让学生经历探究指令的过程;二是要将流程图与指令进行一一对应,使知识产生前后的联系。

【案例3】《列表存储》一程序设计

在分析完问答程序的升级版——随机出题的问答程序后,流程图随即形成。教师让学生选择所需要的指令到脚本区。学生新建了“i”“得分”两个变量,“问题”“答案”两个列表,“在()和()之间取随机数”“询问”“回答”“如果…那么…否则……”等指令,将其拖到脚本区。

教师进一步引导:变量i的取值范围是多少?

生:有多少题目,就取到多少。

师:用“在()和()之间取随机数”怎么表示?

生:表示成“在(1)和(题目数量)之间取随机数”。

师:“题目数量”有没有更好的表示方法?请你去“变量”类指令的面板区找一找。

经过一番查找,学生找到了关键指令。

这时,教师请学生勾选变量“i”前面的方框,使其显示,再单击指令观察变量“i”的值有什么变化。

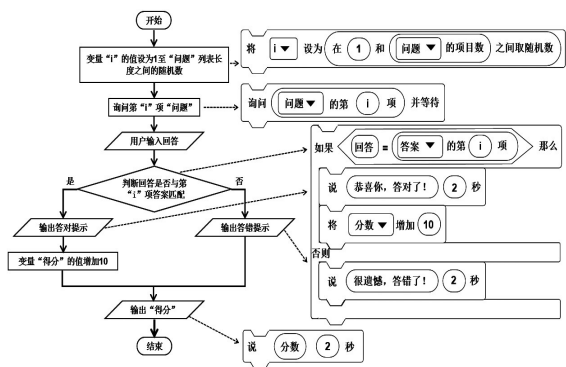


图2

通过教师的提问引导,学生提出猜想、验证猜想,理解了新指令“随机数”的含义。同样,对于其他几个关键的新指令,教师也可让学生先行探究、尝试,从而实现知识的内化。

在这个过程中,教师以流程图为蓝本,一步步

引导学生对其中的关键步骤所需要的指令进行探究、尝试、验证。学生经历了新知识的发现过程,将流程图与指令、脚本联结在一起,整体梳理了程序设计的思路。

(二)反向解析,强化对关键指令的理解

教师也可以通过让学生给程序脚本添加注释的方法,检查学生对关键指令的理解程度。一般的程序设计软件都带有给脚本添加注释的功能。按照教师的要求,学生可以在关键指令处添加注释,写明所用指令的用途。

【案例4】《列表存储》一程序设计

以《列表存储》一课为例,关键指令的注释如图3。

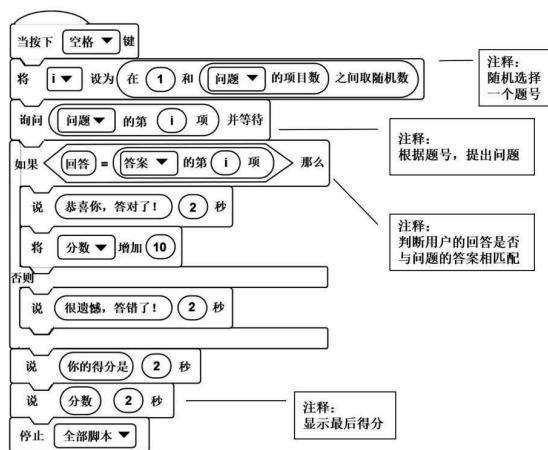


图3

通过正向将流程图与关键指令进行匹配,反向给关键指令添加注释的方式,学生经历了两次程序的梳理,对流程图和程序脚本之间的对应关系有了更深刻的理解,对关键指令的使用方法也更明确,知识得以内化。

三、聚焦关键差异,培养迭代思维

缺乏能激发真实需求的情境是导致学生较难提出合理化改进建议的原因之一。一方面,受到思维局限,学生对于自身制作的程序设计作品难以进行自我更新;另一方面,学生对于一些新指令的陌生感则限制了想象力的发挥。因此,在作品的改进设计环节,教师要有意识地创设能激发真实需求的情境,多角度发散学生的思维,鼓励学生说出自己的改进思路。

(一)自我修正,借助多维优化更新程序

对于程序设计的优化,学生思维受限的原因之一是缺少创新角度的点拨。课堂中,如果教师没有提前设计学习支架,有效提示学生进行程序脚本的改进,而是直接让学生自由发挥,往往很难达到预期的效果。在程序设计的拓展环节,教师可以设计能激发学生探索新指令、解决新问题的合理情境或任务,让学生找出关键差异,然后带着问题进一步探索。

【案例5】项目学习“有趣的图形”中程序设计的改进

项目学习“有趣的图形”中,学生通过循环嵌套的方式,学会了画出一个由基本图形旋转而成的组合图形。在后续的项目学习中,教师引导学生逐步优化程序,并通过一组图形的变化,寻找关键差异,学习新指令,从而不断地将“过程”这一程序设计中的重要概念学实、学透。

1. 图形组的第一次变化(关键差异:基本图形)

教师请学生说说脚本应该如何修改。

学生:将每一段画图形的脚本中“内层循环”改成画正方形的脚本,即重复4次,右转90度。

教师:这样修改需要改动多处,有些麻烦,使用“过程”,一次修改即可。

(学生学习“过程”指令的运用)

学生:把三角形这一“过程”的名称改成“正方形”,然后把过程脚本修改一次,主程序中的“过程”指令就自动改过来了。

2. 图形组的第二次变化(关键差异:基本图形及其边长)

学生已经可以使用学过的“过程”指令来修改基本图形的变化,但边长的变化学生尚未掌握。

学生:每个正五边形的边长都不一样,每次都要去修改正五边形这个“过程”。

教师:“过程”是一段固定的、可多次使用的脚本,一经定义,一般不随意修改。尝试添加一个参数“边长”,有什么发现?

学生:有了参数,就可以在带参过程“正五边形(边数)”里输入数字,自由控制边长。

3. 图形组的第三次变化(关键差异:基本图形的边长、边数)

通过前期优化程序的经验,学生能够想到再增加一个参数“边数”,使用带参过程“正多边形(边长)(边数)”来解决这个问题,其对“过程”的认识进一步丰富。通过三次优化,学生经历了发现差异、学习新指令、修改脚本的过程,从而实现了程序迭代优化。学生对于“过程”这一新知识点的认识由表及里,由浅入深,对于“过程”的应用也更加得心应手。

(二)分享互学,借鉴他人之长优化程序

在优化程序的过程中,学生思维受限的另一个原因是缺少同伴间的交流分享。在分享作品、改善作品的过程中,抓住关键差异,聚焦优化程序的切入点是非常重要的。此时,教师需要精心设计互学分享的支架,帮助学生打开思路,表达创意。

【案例6】《创编游戏》一课中的程序优化

在教学《创编游戏》一课时,教师请学生先自主尝试,在完成课本要求的基础上,改进自己的游戏程序。一段时间后,教师请学生带着学习单找找其他同学的作品,记录变化。如图4,生1在游戏中设置了多条路径;如图5,生2在游戏中添加了障碍物。

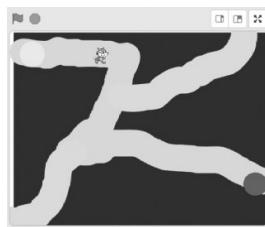


图4

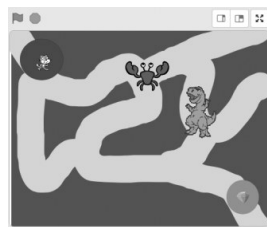


图5

通过观察记录,学生发现了同伴设计中的亮点,记录下了互学体验,新的创意在交流中不断迸发,作品在相互借鉴中不断臻于完善。

实践证明,运用梳理关键步骤、寻找关键指令、聚焦关键差异等策略,能够在一定程度上锻炼学生的分解思维、算法思维、迭代思维,从而帮助学生更高效地达成学习目标,形成计算思维。

(浙江省杭州市笕桥小学 310021

浙江省杭州市采荷实验学校 310020)