

提高中学生计算机编程能力的研究

刘明星

(宜宾市翠屏区成外学校, 四川 宜宾 644000)

摘要: 教育数字化背景下, 编程教育越来越受重视, 与培养数字化人才、发展创新型国家联系起来。编程教育发展时间短, 在教育开展中受到工具论等的影响, 难免剑走偏锋, 部分学校甚至将编程教育等同于程序设计语言学习、简化为程序的模仿练习, 甚至将其异化为编程测试题训练。编程能力的培养应当以发展编程创新性思维、掌握编程应用技能、理解编程理论知识为目标, 着眼于编程特点和规律, 发展符合学生认知规律的教育模式, 为学生数字化能力的培养提供坚实保障。

关键词: 编程能力; 中学生; 教学模式; 教育策略

在智能时代, 技术革新催生了新业态、新局势, 使得社会朝着更加智能化的方向发展, 人机协同成为未来社会发展的趋势之一, 因此数字化思维、编程能力成为未来人才需要掌握的关键性能力之一。中学教育应当关注人与数字环境的关系, 重视数字化生存技能培养, 将信息技术课程中编程能力培养作为编程教育发展的重要任务。

一、编程教育的现存问题

编程能力是用数字化思维方式去思考, 用程序驱动的数字设备以编程解决问题的能力, 是数字素养与技能体现。伴随着我国建设创新型国家和世界科技强国的步伐, 编程能力培养在中学教育中越来越重要, 2017年国务院印发《新一代人工智能发展规划》其中提出要在中小学阶段设置人工智能相关课程, 逐步推广编程教育。《义务教育信息科技课程标准(2022年版)》中明确提出要让学生尝试设计求解算法, 并通过程序进行验证, 让学生体验编程, 发展计算思维、数字化素养。

(一) 局限于理论与技能培养, 忽视思维和兴趣发展

编程能力培养不局限于编程理论与技能学习, 更强调计算思维、数字化学习能力, 要求学生通过抽象、分解、建模、算法设计等思维活动形成用程序解决问题的能力。也就是说, 编程教育既包括编程理论与技能学习, 同时也包括计算思维、创新思维、数字化学习能力的培养。学生不只是学习程序设计语言中的变量、运算符、表达式、循环语句等内容, 还掌握用程序设计语言解决实际问题的方法。脱离问题情境, 只关注程序设计知识技能的编程能力培养工作是片面的、错误的, 将编程教育与生活场景、现实问题割裂的编程教育不仅失去了教育的本来意义, 更抹杀了编程学习的趣味性, 使得编程教学编程枯燥的记忆性课程, 抹杀了编程教育的魅力。

(二) 片面化为模仿练习, 忽视自主思考和探究

编程能力发展的过程是一项不断“试误”和逐步完善的过程, 持续性地对编程方案进行优化和迭代, 在修正、完善的过程中强化编程能力, 直至完成任务需求。因此, 编程教育要始终坚持以学生为中心, 创新创新教学模式, 引导学生自主参与到编程中, 自主探究, 经历描述问题、完成任务、建构知识和运用知识的全过程, 直至学生掌握用编程解决问题的方法。然而, 以生为本的教学实践难度比较大, 在教学实践中, 编程教育仍主要以教师为中心, 以模仿训练方式为主。只打造编程活动情境, 但是忽略了让学生思考、探究, 未能让学生经历问题分析、方案设计、试误与优化探究过程的编程教育也失去了本来意义, 由此导致了编程教育陷入误区。以模仿练习为主的编程教育并不能真正让学生生成编程能力, 一旦更换问题情境, 学生就难以找到问题解决方案。

(三) 功利化问题突出, 增加了学生负担

编程能力培养的本质在于通过让学生掌握编程方法、工具和

资源, 推动学生自主地参与到认识和改造世界的现实活动中, 逐步引导学生成长为掌握技术、具备创新思维的技术反思者。但是, 编程能力培养受到功利思维的影响, 受“选拔指挥棒”的影响, 让学生在选拔考试中获得好成绩, 部分学校将编程教育异化为编程测试题训练, 通过“题海战术”强化学生对编程知识与技能的熟练掌握程度, 提高学生应试能力。这严重偏离了编程教育的本质, 不仅丧失了编程教育的本来意义, 还给学生增加了学习负担, 让编程教育变成高强度、重复性做题训练。

二、提高中学生计算机编程能力的有效策略

(一) 开展项目化教学, 打造开放创新的学习环境

引入项目化教学模式, 设计编程项目。联系真实问题设计编程项目, 构建真实化情景, 组织学生结成学习小组, 通过编程解决问题或完成任务, 在此过程中构建知识体系, 发展编程能力。项目化教学不仅能使学生有效掌握编程知识与技能, 更能培养学生的创造性思维 and 实践能力, 让他们得以应用编程方法去创造性地解决现实问题。例如, 联系生活中的浇花场景, 设计“智能浇花系统”项目, 通过编程实现土壤湿度判断和自动浇花指令下达。在项目中, 学生需要描述自动浇水系统需解决的问题, 设计问题解决方案, 通过编写程序实现功能。项目体现了情境性、问题性和开放性的特征, 让学生发展了创造能力、编程能力。

创设情境。联系真实场景, 将学生引导到解决问题活动中, 开展项目。情境为学生提供了探索真实任务的空间, 产生了认知冲突, 并激发学生的创造力、探究力。情境将编程知识与技能与问题解决联系起来。

提出问题。提出需要解决的问题是编程项目得以继续发展的关键性步骤。问题让学生调动经验、学习知识与技能, 并在感知、发现和掌握中形成解决问题的编程方案。在教育实践中, 问题让学生持续优化、迭代编程方案, 经历螺旋上升的认知发展过程, 最终掌握并创造性地应用编程知识与技能。可见, 问题是学生掌握编程能力的动力。

项目开放化。编程项目方案的设计与执行过程是学生抽象项目关键特征, 按照算法思维分析已知条件、描述问题、设计问题解决步骤, 通过编程验证、调试和持续完善项目方案并创新项目作品的过程。在此过程中

项目主题应具有开放性, 学生可依据主题要求灵活选择活动方向、设计实施过程、创新项目成果。开放性使得学生得以发散思维, 创新方法, 为形成创造性编程方案提供了空间和平台。例如, 在“智能浇花系统”项目活动中, 不拘泥于某一特定类型的传感器、编程思路, 只要能实现智能浇花功能即可, 让学生得以充分发挥创造力。

项目化教学具有情境性、问题性和开放性的典型特点, 引入到中学编程能力培养中, 可以为学生打造广阔的探究空间、系统

的问题支架和充满创造力的学习氛围。

(二) 关注编程实践参与, 让学生亲历编程全过程

培养编程能力必须让学生经历完整的编程过程, 要让学生能够完成编程任务、设计编程方案, 而不只单纯让学生记住或者会操作程序设计语言的语法知识和语句结构, 更要引导学生经历编程探究的全过程, 解决实际问题, 完成编程作品。编程过程可以分为四个环节:

其一, 明确需要解决的问题。根据项目情境展开需求分析, 界定需要解决的核心问题, 抽象问题的关键因素, 描述所需解决的问题, 并列示预期结果。例如, 首先抽出情境中的核心问题, 之后针对这一复杂性问题的, 采用编程过程中的分解思维方式, 将问题分成若干个小问题, 通过逐步求解小问题最终得到整个问题的解决方案。

其二, 设计并实施编程方案。通过编程方式创新解决问题的过程与步骤, 打造一个完整的问题解决方案。编程方案需要反复优化, 直至最后得到最优解, 能够创新性地解决实际问题。要描述编程方案, 将每一个步骤、环节通过自然语言或流程图以顺序、分支和循环控制结构进行明确、清晰地描述, 为有序、清晰地执行解决问题提供参考。

其三, 编程验证方案。即选择合适的编程环境将问题解决方案的逻辑和步骤转化为可执行的程序, 通过执行程序实现问题解决。该过程既是学习程序设计语言知识的过程, 也是应用编程环境创新编程作品的过程。

最后, 分享实施方案, 完善创新成果。完成编程方案后, 展开测试和运行, 检验程序是否正确执行, 对程序运行中出现的问题提出解决方案, 并对程序运行方案进行评估, 将其反复优化, 可通过寻求同伴和他人的建议寻找编程优化思路, 直至程序能够流畅、快速地运行, 能够高效解决问题。

让学生在经历整个编程过程, 从描述项目问题、设计编程方案、执行编程方案到最后的程序调试与运行, 学生切实发展了编程能力, 得到一个完整的编程作品。

(三) 持续优化编程成果, 完善编程方案

编程优化是借助编程环境, 通过优化算法、应用系统, 提升程序的稳定性与实用性的过程。编程优化需要学生分析程序运行和应用效果, 整合用户建议, 并从功能特征、技术性能和用户体验等方面对程序进行迭代和优化。

功能优化。对照问题描述与预期结果分析, 检测编程成果是否达到预期目标, 对未能实现的功能优化设计并实施编程方案。例如, 在“智能浇花系统”项目中, 对土壤湿度的判断不准, 这就需要进一步优化方案, 增加更多土壤湿度判断程序, 判断结果互证, 得出多方互证结论。

技术性能优化。程序作品的运行效果和响应速度是编程方案质量的重要体现。编程项目活动过程中, 学生还需进一步考察程序作品的运行效率, 从算法设计、空间存储、语句结构等方面进一步优化程序作品, 提升自身的编程知识和能力的同时, 使得创造性的编程成果得到进一步的提升。

用户体验的优化。用户体验包括人机交互界面、程序作品使用流畅度、生成报告或结果的可用性等。在程序设计创新的过程中, 往往会出现一些事先没有预见到的问题, 而通过用户体验, 可以发现新的需求, 从而有针对性地对其进行改进。在程序创新成果初步形成之后, 可以邀请其他学生或者教师扮演用户对程序进行测试, 通过成果展示、交流、研讨, 听取他们的意见, 根据其体验和意见, 不断地对编程创新成果进行迭代和优化。

(四) 打造试错空间, 螺旋上升发展编程能力

编程能力培养要秉承“做中学”的理念, 突出开放性、情景化等特征, 突破了传统编程教学以教师为主体、以模仿训练为主的模式, 打造一个具有创新性、提供反复试错空间的学习平台, 让学生能够反复对编程方案进行优化与改进。

做好问题分析和描述。在编程能力培养中, 分析和描述问题是一个非常重要的过程, 它不仅要对问题的定义、描述, 而且要预设问题解决结果。然而, 由于情境的复杂性和需求的多样性, 很多时候很难一次性准确完整地对问题进行分析, 这是在编程创新的过程中, 通过对问题的反复思考和深刻的理解, 对前期没有充分考虑的问题进行进一步的细化和改进, 从而获得的。比如, 我们在使用各种软件的过程中, 可以发现软件的版本总是在持续地更新, 随着开发者对问题的理解越来越深刻, 程序功能也持续完善。因此, 程序应用的功能和性质也越来越强。

设计并实施编程方案。当用户对问题的认知程度、功能需求和技术效率等方面的需求变化时, 程序的执行计划也会随之变化。在此基础上, 项目提出了一种新的、面向用户的、基于不同应用场景的编程方案, 并通过对该方法的不断完善, 来提升程序设计的创新性结果的质量。

持续优化程序。程序要通过反复的试错和优化来不断完善, 进一步对照问题描述和预设的实施结果, 验证程序实施方案, 逐步优化程序。在此基础上, 学生可以针对新的需求对程序进行改进, 以增强程序运行的稳定性与安全性。从这一点上可以看出, 要想提升学生的程序设计创新能力, 除了要给学生提供一个探究问题的空间之外, 还要给他们提供一个开放的、不断探究的途径, 从而逐步提升程序设计的创新能力。

三、结语

在编程能力培养中, 经常发现这样的情况: 学生上课听得懂教师讲解的例题, 甚至也能自己看懂一些源程序。但一旦转化场景让学生自主设计编程, 学生就不知从何下手。学生的编程往往只停留在依教师的例题“依样画葫芦”的水平, 一旦稍有改变, 便不知从何入手。如何提高学生的编程能力成为每位教师在程序教学中面对的突出问题。如何改变编程能力培养中课程枯燥无聊的问题, 使难以理解、复杂的计算机语言更加形象, 提高学生的学习兴趣, 使学生逐渐掌握编程的基本原则、方法, 树立编程理念, 增强综合能力和计算思维, 突出信息技术课程的价值, 本文展开了系统论述, 讨论了开展项目化教学、关注编程实践参与、注重编程方案的持续优化、打造试错空间四个教学策略, 希望为编程教学工作发展提供一些借鉴和参考。

参考文献:

- [1] 韩雪婧, 汪基德, 王孝培. 编程教育对中小学生学习效果的影响——基于国内外36项实验与准实验研究的元分析[J]. 数字教育, 2022, 8(06): 56-62.
- [2] 杨国荣, 崔忠伟, 余丽静. 基于Scratch的创客型信息技术师范生编程能力培养模式研究[J]. 中国新通信, 2021, 23(10): 178-179.
- [3] 徐锦霞, 钱朱银. 人工智能视域下我国中小学编程课程的构建与实施[J]. 汉字文化, 2021(08): 157-158.
- [4] 杜晓敏, 史松竹. 人工智能时代中小学编程课程体系构建的实践探索[J]. 基础教育课程, 2020(23): 20-24.
- [5] 张琦旋, 刘扬, 徐瑞祺等. 基于“编程一小时”与Scratch培养中小学生编程能力[J]. 软件, 2020, 41(08): 229-232.

* 四川省教育信息化与大数据中心的课题—创客理念下趣味编程在课后服务中的研究(立项号: DSJ2022273)