

我国青少年编程教育课程标准探讨

孙丹 李艳

(浙江大学教育学院, 杭州 310028)

[摘要] 编程能力被认为是智能时代人才必备的重要技能之一,青少年编程能力的培养更是寄托了国家的希望和民族的未来。然而,与发达国家相比,我国青少年编程教育的实践和研究仍处于起步阶段,在体系、内容和师资等方面还需要开展大量工作。其中,当务之急是开发相对规范的青少年编程教育课程标准。本研究在分析《英国国家课程计算学习项目》(2013版)、美国《K-12 计算机科学框架》(2016版),以及我国《普通高中信息技术课程标准》(2017版)的基础上,结合国内主流青少年编程教育机构的实践与研究,探讨我国青少年编程教育课程标准的核心内容,包括基本理念、理论基础、核心素养、课程目标和具体内容等,重点介绍入门、初级、中级和高级四个学习阶段的具体目标以及与之匹配的内容和案例示范,研究最后对编程教育课程研发和实施提出了建议。本研究希望通过对课程标准的研讨,更好地引导青少年编程教育领域的研究和实践。

[关键词] 青少年;计算;编程;编程教育;课程标准

[中图分类号] G423

[文献标识码] A

[文章编号] 1007-2179(2019)05-0099-11

一、引言

随着智能时代的到来,世界上很多国家已经将人工智能列入国家发展战略,试图通过人工智能技术的研发以及在行业内的广泛应用占领全球科技竞争的新高地。未来人才的计算思维和编程素养培养问题因此备受各国学界关注,编程课程学习的重要性日益凸显(刘敏娜等,2018)。在过去几十年里,英、美等发达国家一直引领着全球计算机产业的发展,它们在计算机科学教育领域的实践和研究同样领跑全球(寇光杰等,2012)。

然而,随着时代的推移,英国教育主管部门发现,青少年接受的信息技术教育很难与未来从事的

技术工作所需能力匹配(BCS, 2011)。为此,英国教育部于2013年9月颁布的《英国国家课程计算学习项目:关键阶段1和2》和《英国国家课程计算学习项目:关键阶段3和4》,界定和阐述了中小学阶段“计算”(computing)能力的培养及编程教育的内容。

美国进入21世纪以来,K-12阶段的计算机科学教育实施存在诸多挑战,如计算机科学教育的内容选择、计算思维测评等(Google & Gallup, 2015)。为了响应各州对计算机科学教育指南性文件的迫切需求,全美各领域专家经研讨于2016年11月发布了《K-12 计算机科学框架》(简称《框架》)(K-12 Computer Science Framework Steering Committee,

[收稿日期] 2019-06-22

[修回日期] 2019-08-11

[DOI编码] 10.13966/j.cnki.kfjyyj.2019.05.010

[基金项目] 2019年度浙江省哲学社会科学新兴(交叉)学科重大扶持课题“面向学生批判性思维提升的心理-技术-课程的交叉协同研究”子课题“利用现代教育技术开展学生批判性思维培养的理论与实践研究”(19XXJCZD-2)。

[作者简介] 孙丹,博士研究生,浙江大学教育学院,研究方向:智能教育、编程教育、学习分析、数字化学习等(11803011@zju.edu.cn);李艳,博士,教授,博士生导师,浙江大学教育学院,研究方向:数字化学习、远程教育、信息技术教育、科技教育、媒介教育、智能教育、教育创新传播(yanli@zju.edu.cn)。

2016),指导美国 K-12 计算机科学教育实践,其中包含对青少年编程教育的内容建议。在实践层面,美国以 STEM 教育为切入口,在青少年编程教育领域进行跨学科的课程规划和实践(赵中建,2015)。美国新媒体联盟发布的《2017 年地平线报告(基础教育版)》也指出,在未来几年,编程作为一项专业素养将逐渐成为驱动基础教育发展的关键要素(NMC Horizon Report,2018)。

2013 年,我国教育部启动了高中信息技术课程标准修订工作,《普通高中信息技术课程标准(2017 年版)》(简称《课标》)应运而生(中华人民共和国教育部,2017),其中包含编程教育的内容。2017 年 7 月,中国国务院发布的《新一代人工智能发展规划》提出,要“广泛开展人工智能科普活动,在中小学设置人工智能相关课程,逐步推广编程教育”(国务院新闻办公室,2017)。2019 年 3 月 13 日,教育部公布的《2019 年教育信息化和网络安全工作要点》指出,“今年将开始推动在中小学阶段设置人工智能相关课程,逐步推广编程教育”(教育部办公厅,2019)。

国内中小学校的编程课程主要采用综合实践课、兴趣拓展课等形式,已成为中小学校建设特色课程的重要组成部分(李玉阁等,2018)。青少年编程课程的推广推进了中小学地方课程、校本课程建设和实施,为中小学校打造校本特色、区域特色的课程提供了解决方案(杨帆,2019)。我国校外青少年编程教育的发展近年来更是如火如荼。2018 年 6 月,亿欧智库(2018)发布的《2018 中国少儿编程教育行业研究报告》显示,国内青少年编程教育正向快速发展阶段过渡。艾瑞咨询(2018)发布的《2018 年中国少儿编程行业研究报告》显示,截至 2018 年 10 月,我国少儿编程行业市场规模约 30 亿至 40 亿元,用户约 1550 万,行业规模 5 年内将达到 300 亿。编程课程是学科课程的延伸、融合与提升,既是各学科课程基础知识的拓展应用,也是对学生综合素质及能力的实践检验。因此,青少年编程教育课程的设置,为提升学生的综合素养意义重大。

然而,我国青少年编程教育整体上仍处于起步阶段,课标制定、课程建设、教材研发,以及人才评价等都需要投入大量人力和物力(孙丹等,2019)。为了更好地助推我国青少年编程教育的发展,本研究

尝试分析当前英、美、中三国的计算机科学教育相关课标内容,从课程设计的基本理念、理论基础、核心素养、课程目标和具体内容等纬度探讨中国青少年编程教育课程标准的核心内容,以期对该领域的研究与实践提供参考。

二、英、美、中计算机科学教育 编程教育内容比较

(一)《英国国家课程计算学习项目》(2013 版)的编程教育内容

英国青少年编程教育通过“计算学习项目”(computing programmes of study)实施。英国教育部 2013 年发布的《英国国家课程计算学习项目:关键阶段 1 和 2》和《英国国家课程计算学习项目:关键阶段 3 和 4》规定,计算学习项目融入英国中小学的所有学段,并设定了该项目的四大学习目标:1)确保所有学生理解和应用计算机的基本原理和概念;2)能用计算术语分析问题,有编写计算机程序的实践经验;3)能评估和应用新的信息技术;4)成为对技术负责的、有能力的、自信的和有创造力的用户。

计算学习项目在英国基础教育有四个关键学习阶段,分别是 1-2 年级、3-6 年级、7-9 年级和 10-11 年级,每个关键阶段的编程学习目标都有细致的描述(见表一)。此外,该项目还描述了每个关键阶段结束时学生应理解、掌握和应用课程规定的概念和技能等(Department for Education, 2013)。学者拉克(Larke,2019)指出,高质量计算机教育能使学生利用计算思维和创造力理解和改变世界,“计算”的提出旨在为年轻人提供充分参与日益数字化的世界所必需的技能、知识和思维方式。

(二)美国《K-12 计算机科学框架》(2016 年版)的编程教育内容

美国《K-12 计算机科学框架》(2016)界定了 K-12 阶段计算机科学课程的五大核心学习模块,即计算机系统、网络和互联网、数据与分析、算法与编程和计算机影响。其中,“算法与编程”模块专门介绍编程教育的内容。《框架》将美国 K-12 阶段分为四个学段,即 1-2 年级、3-5 年级、6-8 年级和 9-12 年级。《框架》描述了每个学段学生不同学习模块应掌握的核心概念(见表二)。由这些核心概念不难发现,抽象、人机交互、隐私与安全、交流与协作等概

表一 英国国家课程计算学习项目的阶段目标

关键阶段	学习目标
第一关键阶段 (1-2 年级)	<ul style="list-style-type: none"> 了解什么是算法,如何利用程序实现; 创建和调试简单程序; 使用逻辑推理预测简单程序; 安全使用技术,保护个人信息的私密性及如何获得帮助等。
第二关键阶段 (3-6 年级)	<ul style="list-style-type: none"> 学会设计、编写和调试程序; 将问题分解成更小的部分解决问题; 使用顺序、选择和循环方法编程; 使用变量和各种形式的输入和输出; 使用逻辑推理解释简单的算法及调试算法和程序错误; 安全和负责任地使用技术; 认识可接受/不可接受的行为等。
第三关键阶段 (7-9 年级)	<ul style="list-style-type: none"> 设计、使用和评估计算机抽象概念模拟现实世界问题和物理系统的状态和行为; 理解反映计算思维的关键算法,如排序和搜索算法;使用逻辑推理比较解决相同问题的最优算法; 使用两种或多种编程语言(至少一种是文本语言)解决各种计算问题;适当利用数据结构,如列表、表格或数组;设计和开发使用流程或功能的模块化程序; 理解简单的布尔逻辑,如“与”“或”和“非”,及其在电路和编程中的应用;了解如何用二进制表示数字,并能利用二进制数字进行简单运算,如二进制加法和二进制与十进制之间的转换; 了解组成计算机系统的硬件和软件,以及它们如何相互通信和与其他系统通信; 理解如何存储和执行计算机指令;了解各种类型的数据(包括文本、声音和图片);如何以二进制表示和处理数字; 开展创造性的项目,包括选择、使用和组合多个应用程序,最好是跨系列设备,以实现挑战性目标,包括收集和分析数据以及满足已知用户的需求; 为给定受众创建、重用、修改和重新利用数字产品,关注可信度、设计和可用性; 了解安全、尊重、负责任和安全地使用技术的方法,包括保护在线身份和隐私;识别不适当的内容、联系和行为,并知道如何报告问题。
第四关键阶段 (10-11 年级)	<ul style="list-style-type: none"> 发展学生的计算机科学、数字媒体和信息技术的能、创造力和知识; 开发和应用他们分析、解决问题、设计和计算思维技能; 了解技术变化如何影响安全,包括保护在线隐私和身份的新方法,以及如何识别和报告问题。

表二 美国《K-12 计算机科学框架》不同学段的核心学习模块和核心概念

学习阶段	核心学习模块与核心概念				
	“计算机系统”模块	“网络与互联网”模块	“数据与分析”模块	“算法与编程”模块	“计算影响”模块
1-2 年级	人机交互、交流与协作、系统关系	人机交互、隐私与安全、交流与协作	抽象、系统关系、隐私与安全、人机交互	抽象、系统关系、人机交互	人机交互、隐私与安全
3-5 年级	抽象、系统关系、交流与协作	抽象、交流与协作、隐私与安全	抽象、系统关系、人机交互	抽象、系统关系、人机交互、交流与协作	系统关系、人机交互、隐私与安全
6-8 年级	抽象、系统关系、人机交互、隐私与安全、交流与协作	抽象、隐私与安全、交流与协作	抽象、人机交互、隐私与安全	抽象、系统关系、人机交互	系统关系、人机交互、隐私与安全、交流与协作
9-12 年级	抽象、系统关系、人机交互、隐私与安全、交流与协作	抽象、系统关系、人机交互、隐私与安全、交流与协作	抽象、系统关系、人机交互、隐私与安全、交流与协作	抽象、系统关系、人机交互、隐私与安全	抽象、系统关系、人机交互、隐私与安全

念几乎贯穿所有学段和学习模块,从中可见,美国计算机科学教育强调人机交互和隐私安全等内容。

值得注意的是,《框架》不仅提出了五大核心学习模块和核心概念,更重要的是明确了每个模块的具体目标并对其进行了详细描述。针对“算法与编程”学习模块,《框架》先界定了核心概念,如算法是为完成特定任务而设计的系列步骤;创建有意义且高效的程序开发过程包括选择信息、处理和存储信息、分析问题、重组解决方案及分析不同解决方案等。《框架》还提出,“算法与编程模块”的五大子概

念,即算法、变量、控制结构、模块化及程序开发,并给出了四个学段中这些核心子概念的具体学习内容示例。如二年级结束时,学生应了解程序可以表示为计算机可以理解的算法;五年级结束时,学生应能选择解决具体问题的最优算法;八年级结束时,学生应能测试、调试、改进算法;十二年级结束时,学生应学会基于性能、可重用性、易用性等评估和选择算法等。这些内容构成了美国 K-12 阶段编程教育的核心。此外,《框架》提出了围绕计算思维的核心实践,包括创建全纳的计算文化、通过计算开展合作、

识别和定义计算问题、发展和使用抽象思维、创造计算产品、测试和改善计算产品以及计算的沟通。赵中建(2017)指出,《框架》的面世是信息社会技术发展的必然需求,是美国教育倡导公平的一项举措,符合一门学科存在的内在要素和逻辑。

(三)中国《普通高中信息技术课程标准》(2017版)的编程教育内容

我国2017年颁布的《普通高中信息技术课程标准》(简称《课标》)不仅确立了高中信息技术学科的核心素养(包括信息意识、计算思维、数字化学习与创新及信息社会责任),梳理了信息技术学科的大概念,还重构了高中信息技术课程的内容与模块(邱美玲等,2018)。《课标》建议的课程内容包含两个必修模块(数据与计算、信息系统与社会)、六大选择性必修模块(数据与数据结构、网络基础、数据管理与分析、人工智能初步、三维设计与创意以及开源硬件项目设计)和两个选修模块(算法初步、移动应用设计)。

从《课标》设置的两个必修模块可以发现,与编程相关的教学内容主要设置在“数据与计算”必修模块的“算法与程序实现”中。《课标》要求学生“掌握一种程序设计语言的基本知识,使用程序设计语言实现简单算法。通过解决实际问题,体验程序设计的基本流程,感受算法的效率,掌握程序调试与运行的方法”。同时,《课标》注重学生在虚拟空间和现实空间培养生存与发展能力、提升信息安全和社会责任意识。这份标准的修订依据数字环境发展需要和学生认知特征,对高中阶段信息技术学科的建设与发展提供了重要的指导意义(李锋等,2016)。

(四)英、美、中计算机科学教育的编程教育内容比较

计算机科学/信息技术教育已成为英、美、中三国中小学课程的重要组成部分。不过,英、美、中三国对学生所需掌握技能的要求不同:中国将编程教学作为必修模块并提出总体要求,英国在四大目标的基础上对中小学生的编程能力提出了总体要求,美国则是将算法和编程能力分为五个方面并各自提出要求,但三国青少年编程教育的基本内容相同,都包含学习设计、编写、调试等基本能力,包含顺序、选择、循环等算法基本结构,以及安全、负责任使用技术等使用规范。

在课程设置的阶段性上,英美两国均将编程教育课程贯穿于K-12阶段,分四个学段实施,并界定了各阶段结束时应掌握的编程知识与技能。其中,美国最详尽。同时,英、美两国课标设计体现的理念和内容,尤其是对计算机思维、创新能力等核心素养的培养,与我国信息技术高中新课标界定的要求有较高的一致性(赵蔚等,2017),这为我国目前尚缺的初中、小学阶段编程课程标准的设置提供了参考。卢蓓蓉等(2017)提出,美国K-12《计算机科学框架》对改进我国基础教育信息技术课程标准的修订、提升标准的修订质量、提高教材的开发水平及促进课程教学都有借鉴意义。

三、我国青少年编程教育课程编制的基本理念和理论基础

根据国内外课程标准制定的一般做法,中国青少年编程教育课程标准在编制具体内容之前需要思考该课程编制的基本理念和理论基础,基本理念是指导课程标准设计的重要依据,理论基础体现课程标准的根基和依据。中国青少年编程教育课程应遵循与时俱进的教育思想和理念,需参照国际前沿的学科实践和研究成果,依据编程学习的自身规律及遵循青少年的身心发展特征,同时还需其他教与学理论基础的支持。例如,从建构主义角度思考“什么是编程学习”与“如何进行编程学习”;从体验式学习角度建构“如何通过教学实践助力学生思维发展”等。基于这些思考,我们提出青少年编程教育课程编制的基本理念和理论基础。

(一)青少年编程教育课程的基本理念

1. 以立德树人为课标设计的指导思想

和其他课程一样,青少年编程教育课程也要体现“立德树人”的思想。课程标准将引导学生不断思考技术对生活与社会带来的挑战,个人在智能时代需承担的责任与义务,帮助青少年成长为积极的技术使用者、创新的技术开发者以及遵循道德规范的技术守护者;同时培养学生对技术发展,尤其是编程技术发展的敏感度与适应性,帮助学生有效地利用编程技术分析、开发与创造丰富多样的工具,优化学习和生活,推动中国智能时代的发展。

2. 参照国际编程教育研究与实践的前沿成果

青少年编程教育课程的建设要依据我国基础教

育的国情,同时结合国际技术变革的前沿热点与发展趋势,帮助学生掌握扎实的基础性编程技能,引导学生成为既掌握编程核心理念,又能与前沿交流对话;课程内容设置要有助于学生连接已有的学习经验与未来智能化的社会生活,从解决实际问题与展望未来的角度促进学生习得编程知识与技能、掌握编程方法及树立正确的价值观。

3. 依据编程学习的自身规律

青少年编程教育课程的设置要理论性、工具性和实践性并重,灵活设计教学情境,在项目学习中培养学生的编程与高阶思维能力。同时,课程设置要体现各阶段课程设置的层次性、多样性与丰富性。初级课程要从兴趣入手,帮助学生建立实际生活与虚拟世界之间的联系,加强其对虚拟世界的理解;中高级课程应不断渗透编程技术的核心理念,掌握抽象概念与方法,逐步理解编程技术。循序渐进的学习过程可以培养学生对课程的浓厚兴趣、提升对课程了解的深度与广度,为学有所长的学生提供个性化又有挑战的学习机会。

4. 遵循青少年的身心发展特征

与其他学科一样,青少年编程教育需充分考虑青少年的身心发展特征,特别是要理解不同阶段学生的思维发展水平,包括分析、综合、概括、抽象、比较、具体化和系统化等思维能力,还要考虑学生的学习背景与知识储备,倡导以学生为中心、多元化的教学策略;培养学生开放、合作的学习习惯,使学生脚踏实地、注重实际,积极参与技术支持的真实的学习;同时,培养学生运用计算思维等高阶思维能力创造性地提出问题解决的能力;创设技术带来的问题与冲突的情境,促进学生思考技术价值、个人行为对社会生活和人文环境的影响。

(二) 青少年编程教育课程的理论基础

1. 建构主义

国际编程教育的实践证明,建构主义理念可以帮助学生更好地理解编程与掌握技术。建构主义强调学习者将原有的知识经验生成意义、形成对事物的理解(Papert, 1993)。建构主义的教学方法多种多样,其共性是教学环节包含情境创设和协作学习,并在此基础上由学习者最终实现对所学知识意义建构(何克抗,1997)。艾哈迈德等(Ahmet & Susan, 2011)在建构主义理论框架下,研究五年级学生设

计和开发计算机游戏的能力发现,学生能按照规划设计、测试和分享游戏,游戏为学生提供了非正式的知识建构和分享机会。阿迪提等(Aditi et al., 2017)对11名10年级学生的K-12科学教育学习环境设计的研究发现,利用建构主义可以促进学生参与基于探究的科学生产活动;尼可拉斯等(Nikolaos et al., 2017)基于建构主义理论教学框架,研究初中生利用Second Life(一种3D游戏环境)学习基础的编程概念研究表明,基于建构主义的教学设计框架可以帮助学生建立社交和认知能力,培养高阶思维和计算思维能力。

2. 体验式学习

学习指通过阅读、听讲、研究、实践获得知识或技能的过程(Sugarman, 1984)。体验式学习强调通过实践认识周围事物,强调认识论与直接经验联系的重要性,并且在尊重人性、个性和情意的人本主义教育理论中得到继承和发展(庞维国,2011)。国际体验式教育学会(Association for Experiential Education, 2004)将体验式教育定位为“一种教育哲学和方法论,教育者有目的地把学生置于直接经验和专心反思中,使其增长知识、发展技能和澄清价值”。体验式教学模式注重学生的参与,强调教学效果的实用性,倡导建立新型师生关系,实现师生双方能力的提升。体验式学习能有效地帮助幼儿提升学习能力以及培养儿童的计算能力(Rahman, 2018)。

四、青少年编程教育课程的核心素养、 课程目标与具体内容

编写青少年编程教育课程首先需要明确课程核心素养,确立课程目标,使学习者明确课程要实现的具体目标和意图(钟启泉,2007)。青少年编程教育课程的核心素养需与《中国学生发展核心素养》目标一致。青少年编程教育课程的目标一方面需要了解该领域未来人才所需的知识 and 能力以及核心素养,另一方面也需了解国内外研究成果和实践经验。青少年编程教育课程的内容框架既要与课程核心素养及课程目标一致,也要考虑该课程的定位、匹配的教学资源(人力、物力和时间等的投入)以及可操作性。

(一) 核心素养

2016年,我国教育部发布了《中国学生发展核

心素养》,其中提到核心素养指学生应具备的,能适应终身发展和社会发展需要的关键能力和必备品格,它以培养“全面发展的人”为核心,分文化基础、自主发展及社会参与三个方面(李锋等,2018)。青少年编程教育课程的核心素养,涉及信息素养、计算思维、数字化学习与创造力、技术价值和责任感四个核心要素。各要素视角不一,但相互关联,互相支撑,共同促进学生编程基础能力以及信息素养的提升,具体表现为:

1. 信息素养

“信息素养”一词最早由美国信息产业协会主席保罗·泽考斯基(Paul Zurkowski)1974年提出。它指利用大量的信息工具及主要信息资源使问题得到解答的技术和技能(陈维维等,2002)。联合国教科文组织(UNESCO,2003)将信息素养界定为这样一种能力,它能够确定、查找、评估、组织和有效地生产、使用和交流信息,并解决面临的问题。在计算机科学教育和编程教育领域中,信息素养可以指学生能够使用信息工具和资源的能力,包括获取识别信息、加工处理信息、传递创造信息的能力,即灵活地运用信息资源进行问题求解,以及培养批判性思维、创新思维等高阶思维活动的的能力。

2. 计算思维

计算思维最早由周以真(Jeanette M. Wing)教授提出,它指运用计算机科学的基础概念进行问题求解、系统设计,以及人类行为理解的涵盖计算机科学之广度的一系列思维活动(Wing,2006)。图灵奖获得者卡普(Karp,2011)认为,基于计算思维的基本原理和方法论,就是对自然问题和社会问题的本质进行恰当提取,使之成为计算机能够处理的形式。计算思维是人类科学思维固有的组成部分,它以抽象化和自动化为主要特征,通过抽象和分解解决复杂问题,即运用计算机科学的思想与方法解决现实问题和理解人类行为。

3. 数字化学习与创造力

数字化学习与创造力指个体能够掌握常见的数字化资源工具,有效地进行学习过程与资源管理,同时从技术的视角创造性地发现与解决真实世界的问题(曹东云等,2018)。数字化学习与创造力是信息时代人才必备的核心素养,教师应引导学生综合利用数字化资源解决问题,促进知识迁移、培养协作能力、

发展批判性能力与创造力,这也是对新一代人才培养要求的积极回应。

4. 技术价值和责任感

技术价值和责任感指信息社会中个体在自我约束、道德规范和责任担当等方面应尽的义务(马若明,2004)。拥有正确技术价值和高度责任感的学生应能较好地认识和理解与技术相关的文化、伦理和社会问题;能及时规避和远离风险,安全并负责任地使用信息技术;面对技术创新产生的新观念和新事物,能保持积极的学习态度、理性的判断力以及负责任的行为担当。

(二) 课程目标

基于未来人才需求以及编程课程核心素养的内容,结合国内外尤其是美国《K-12 计算机科学框架》“算法与编程”的核心概念和内容,本研究建议中国青少年编程课程的目标为:全面提升中国青少年学生的计算思维与信息素养,培养青少年编程的核心能力,即算法、变量、控制结构、模块化、程序开发以及技术的影响等(见图1)。课程通过提供丰富的资源,为学生提供从基础编程到高阶编程能力培养的课程体系;帮助学生学会合理、规范运用技术,成为数字化时代合格的中国公民。为了符合国际标准,中国青少年编程核心能力的关键概念建议采用美国计算机科学教师协会(CSTA,2016)的定义,具体界定如下:

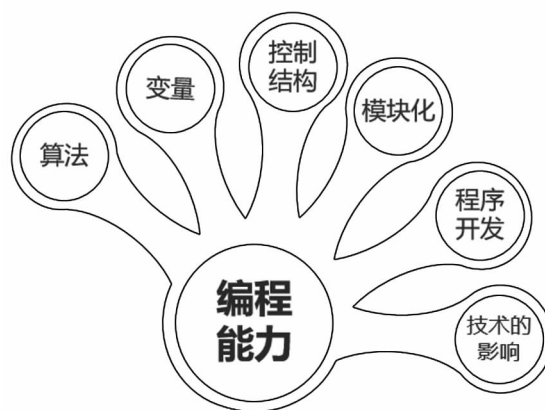


图1 编程能力培养目标

1. 算法

算法是完成特定任务的系列步骤,是解决问题的清晰指令,是用系统的方法描述解决问题的策略

机制。学生先学习现实世界的算法,继而学习计算机算法的开发、组合、分解和评估。

2. 变量

变量是计算机语言中能储存计算结果或能表示值的抽象概念。学生先了解不同类型数据的使用方法,如单词、数字或图片,继而学习复杂变量及其使用方法。

3. 控制结构

控制结构是一种程序运行逻辑,可以指程序中执行指令的顺序。学生先学习顺序结构和其他简单的控制结构,继而学习复杂的控制结构及其组合与嵌套。

4. 模块化

模块化是将复杂系统分解为可管理的模块的方式。模块化涉及复杂任务的分解和简单任务的组合。学生先了解生活实例,知道分解复杂任务及将简单任务进行组合,继而学习识别模式,为常见场景设计解决方案,并以通用的方式描述任务。

5. 程序开发

程序开发是程序员通过设计并调试以得到满意解决方案的过程。学生先学习如何以及为何开发程序,继而学习满足用户需求、注重效率、符合道德规范等复杂决策的程序设计。

6. 技术影响

编程技术的发展影响人们生活的方方面面。个体通过行为和文化影响编程技术的发展,反过来编程技术的发展也推动形成新的文化实践。学生应了解技术对社会的影响,包括促进或破坏公平以及不道德使用技术的危害。

考虑到青少年不同年龄段身心发展和认知规律的差异,英美的编程教育按阶段实施。我国青少年编程教育的课程目标也可以考虑按阶段完成。根据中国的国情和学情,青少年编程课程的目标设计和内容实施可按照12年级四学段分:一至二年级为入门级,三至六年级为初级,七至九年级为中级,十至十二年级为高级。不同学段根据学生身心发展特点,在算法、变量、控制结构、模块化、程序开发和技术的影响六方面设置学段目标(见表三)。

(三)具体内容

1. 内容框架

青少年编程课程的具体内容围绕课程核心素养

和课程目标开展设计和开发,分四阶段,每个阶段围绕六方面(算法、变量、控制结构、模块化、程序开发以及技术的影响)展开不同深度的内容设计与开发,保持内容的延续性和层层深入。课程内容要明确学生在每个阶段、每个方面需要学习的知识点,且通过适当的示例促进知识点的讲解与理解。因文章篇幅的限制,这里以入门阶段的课程内容为例说明。

1)算法:向学生介绍日常生活使用的命令和序列,如小学常出现的开晨会、找教室以及打扫卫生或制作食物、刷牙等,由此告诉学生算法的概念及目的,引导学生学会用算法完成日常任务,通过编写算法,让计算机完成不同的任务。

2)变量:现实世界可以用计算机程序表示。程序可以用于存储和操作数据,如存储数字、单词、颜色和图像。同时,数据类型确定变量的属性和操作,如对于图片数据,可以进行裁剪与重新着色等。

3)控制结构:计算机遵循自动执行任务的精确指令序列,如绘制形状或移动角色。通过重复指令模式和使用事件发起指令,程序执行可以是非顺序的,小学生不要求区分不同类型的循环。

4)模块化:复杂的任务可以分解为更简单的指令,也可以组合简单的指令完成复杂的任务。分解是将任务分解为更简单任务,如为聚会做准备涉及邀请客人、制作食物和摆放桌椅等。这些任务还可以分解,如摆放桌椅包括铺设桌布、折叠餐巾纸以及将餐具和盘子放在桌子上等。组合是将简单任务组合成复杂任务,如为了建造城市,人们需要建造房屋、学校、商店等。

5)程序开发:为了解决某一问题或表达想法,人们需要设计、创建和测试程序。程序开发常被用于创建产品或工具,如开发视频游戏、交互式艺术项目和数字故事等。

6)技术的影响:技术可以对人类产生积极影响,如运动手环可以监测身体数据,也可以带来消极影响,如网络成瘾、社交媒体成瘾、游戏成瘾等。因此,在线发布信息应考虑隐私,避免因长时间的暴露隐私而被他人非法利用;使用技术还要有责任感,如不共享登录信息,保护好密码以及任务完成后注销账号等。

2. 案例建议

为了更好地设计与开发中国青少年编程教育课

表三 青少年编程教育的学段目标

学段目标	一至二年级 (入门级)	三至六年级 (初级)	七至九年级 (中级)	十至十二年级 (高级)
算法	通过算法(步骤/指令)模拟日常生活流程。	比较和选择解决同一任务的最优算法。	了解经典的算法,不同条件下的算法可能有不同的输出;保证算法的可读性。	根据问题情境选择合适的算法,使用经典的算法解决复杂问题(搜索、排序、加密、解密等)。
变量	将信息存储为计算机可处理的数据;了解数据的类型及操作和属性。	掌握常量、变量、数组等数据类型,并进行操作。	为变量分配标识符,提高程序可读性;加深对变量的理解:使用、声明和命名。	了解常见的数据结构。根据功能、存储和性能进行选择,掌握列表的操作。
控制结构	了解计算机机会自动执行精确的指令序列,程序也可以非顺序执行,重复指令模式或用事件可以发起命令。	使用事件、顺序、条件判断和简单循环编写程序。根据情况,选择不同的控制结构。	掌握如何选择以及整合控制结构(例如循环、事件处理、条件以及跳出循环),以创建更复杂的程序。	在选择和组合控制结构时会权衡可实现性、可读性和程序性能;掌握包含事件、顺序、条件、循环和递归的程序。
模块化	了解复杂的任务可以分解为简单的指令,同样简单的指令可组合形成复杂的任务。	将程序分解为更简单的部分,或将简单的程序结构进行重组。	定义过程(函数)组织代码,并可重复使用。将经常使用的代码封装起来,在需要时直接调用。	使用面向对象的编程或模块化方式构建解决问题的方法;使用库或 APIs 或接口开发程序。
程序开发	制定计划描述程序的顺序、目标和期望的结果;运用图形符号或文字展示流程;运用比较和试错解决问题。	通过设计、实施和调试,迭代开发程序,并进行改进和完善。	不断收集用户反馈和约束条件,设计和开发适用于广泛场景的程序。采用以用户为中心的设计,创建解决方案。	使用软件生命周期过程计划和开发面向广大用户的程序;使用控制工具、集成开发环境、协作工具和说明文档;比较不同编程语言并讨论特征及应用场景。
技术的影响	编程技术积极或消极地改变着人们的生活和工作方式;人们可以随时随地访问信息,但他们同时面临网络欺凌和隐私暴露的风险;编程技术可以帮助或损害自己及他人利益,认识并避免有害行为。	编程技术的发展与文化实践之间相互作用、相互影响;编程技术的创新与发展影响各类社会机构;通过互联网发送和接收多媒体信息的便利性为未经授权的使用提供了机会。	编程技术的进步使得社会不断面临日益全球化和自动化选择;人们可以通过编程技术支持的各种通信平台组织和参与交互,发表观点;权衡信息公开和保持信息私密和安全之间的问题。	编程技术的开发和设计,与信息和社会的不平等之间的考虑;技术改变了许多职业的性质和内容;编程技术涉及法律事务,可能会产生利益冲突;法律、道德和国际差异对编程技术存在影响。

程,课程标准可以呈现个别国内外先进的编程课程内容设计与开发案例。本研究建议的案例是由美国哈佛大学教育研究生院 ScratchEd 团队 2019 年 1 月发布的《创意计算》(Creative Computing)课程。该课程包含七个单元,分别是预备、探索与发现、动画、故事、游戏、深入沉浸以及黑客松。

《创意计算》旨在通过培养“创造力”和“计算思维”,提升学生对编程的熟悉度与流畅度,建立学生的创造力和个人兴趣。值得注意的是,编程教育的核心素养体现在活动设计中。这些活动特别鼓励探索关键的计算思维概念(序列、循环、并行、事件、条件、运算符、数据)和关键的计算思维实践(实验和迭代、测试和调试、重用和混合、抽象和模块化)(Creative Computing Curriculum, 2019)。课程内容的设计不但体现了编程的核心能力,还突出跨学科学习的特点。课程内容涵盖计算机科学、工程、语言、艺术、数学、音乐等知识和技能。课程开发团队认为该课程不是针对特定对象开发的,使用领域非

常广泛,包括 K-12、大学以及其他专业发展人员,研发团队鼓励未来课程使用者可以基于不同的教育场景进行全新的活动设计,创造性地使用该课程内容。

五、思考与展望

智能时代,全球教育领域正在掀起一股青少年编程教育的浪潮。巨大市场利益的诱惑让这个行业鱼龙混杂,良莠不齐,一些教学资质和教学条件不达标的培训机构很可能误导学生,它们不仅扰乱行业发展的秩序,更会对学生学习和身心发展产生负面影响。因此,迅猛发展的中国青少年编程教育行业亟需科学的课程标准用以规范实践。

基于对青少年编程教育课程标准的探讨,本研究对青少年编程教育教材编制提出如下建议:

第一,要体现当前国内外编程教育领域先进的教学理念。综观国内外编程教育的优秀实践案例,不难发现,受欢迎的编程教育除了突出科学性和系统性外,还具有非常强的交互性、趣味性以及学科交

又性,突出编程教育与其他学科学习的结合,突出游戏化元素(任务、情境、激励等)融入。这些特点的背后是对当代学习科学理论的综合应用,是对青少年认知和学习规律的尊重,值得借鉴。

第二,兼顾理论性与操作性。教材既要有关键概念及其相互间逻辑关系的陈述,又要有关键概念具体操作的讲解和举例说明,包括编程语言的选择、编程平台的搭建,具体功能的实现,以及指导教师开展师生互动、生生互动等教学策略;确保学习者在掌握概念内涵的同时熟悉操作步骤和要领。

第三,要体现阶段性的差异和联系。各阶段(入门级、初级、中级和高级)的教材内容既要有相对完整性和独立性,体现不同学段学习者特征和认知规律,又要有系统性和连贯性。各阶段的编程教育内容要循序渐进,相互之间有明显的连接和传承。

第四,除呈现主体教学内容(目标、概念、操作、活动、评价等)外,教材应提供丰富的、多媒化的学习资源,如国内外在线学习资源(比如,在线开放课程)、在线学习平台、在线虚拟学习社区、代表性案例、交流和互动平台/空间等,将认知、实践、思维和社交融合,保证内容的新颖性与前沿性。

基于对青少年编程教育课程标准的探讨,本研究对青少年编程教育教学提出如下建议:

第一,编程教育教学要有科学合理的课时规划。根据国际上的作法,小学段编程课程既可融入语、数、英等核心科目,也可以像音体美一样作为独立学科设置;初中段可将编程融入数学、科学等,或者作为选修或社团课设置;高中段除跨学科整合外,可根据学生水平差异设置基础课程和高级课程,同时可以根据实际开设专业选修课。

第二,实际教学要考虑学生实际水平,选择匹配的教学目标、教学资源 and 教学策略。本研究课标虽然给出了四阶段、六维度的课程学习内容和目标建议,但中国地域广阔,城乡之间、东西部之间、学校之间以及个体之间的差异巨大,实际教学应充分了解学生编程学习的知识与能力基础,根据实际适当扩充课程的深度与广度,同时注意及时记录学生的学业水平与学习表现。

第三,提倡中小学校与校外青少年编程教育机构协同合作。当前,青少年编程教育课程还只是中小学校的社团课或选修课,编程教育实施存在师资不足、教

学资源不足等现实困境。比较而言,校外青少年编程教育发展迅猛,吸引了大量科技人才和资本。中小学校可考虑完善与校外青少年编程教育培训机构的合作,校企可基于课程标准提供的内容框架,实践探索教材开发、师资培训、创新教学模式等,为推进青少年编程教育寻找新的模式。

第四,教育主管部门要研究如何对校内外青少年编程教育实践进行质量监管和成效评估。编程教育课程标准可以成为质量监管和成效评估的重要依据。中小学校和青少年编程教育行业机构可以依据本研究建议的课标内容规范自身编程教育。教育主管部门也可以根据建议的课标内容对行业机构加以规范与监管,掌握教育培训机构的教學数据,保障青少年编程教育行业质量。

总体来说,刚刚起步的中国青少年编程教育承载着国家科教兴国的战略使命,吸引了各界人士的极大关注。本研究聚焦中国青少年编程教育课程标准的内容探讨,希望依据国际比较和借鉴厘清中国青少年编程教育课程标准的重要内容,能吸引更多业内人士和学科专家研讨和制定官方的、规范的中国青少年编程教育课程标准,为这个方兴未艾的新兴学科领域划定好边界,提供好准则,明确好内容,推动中国青少年编程教育的实践和研究。

[参考文献]

- [1] Adivi, W., Kate, C. W., & Uri. W. (2017). Bridging inquiry - based science and constructionism; Exploring the alignment between students tinkering with code of computational models and goals of inquiry[J]. *Journal of Research in Science Teaching*, 54(5) : 615-641.
- [2] Ahmet, B., & Susan, M. L. (2011). An investigation of the artifacts and process of constructing computers games about environmental science in a fifth grade classroom[J]. *Educational Technology Research and Development*, 59(6) :765-782.
- [3] 艾瑞咨询(2018). 2018年中国少儿编程行业研究报告[EB/OL]. [2018-12-05]. <http://www.199it.com/archives/781622.html>.
- [4] BCS (2011). Fact sheet, why computing education is important to the UK. [EB/OL]. [2019-08-27]. <http://www.computingschool.org.uk>.
- [5] 北京市人民政府(2019). 国际人工智能与教育大会形成《北京共识》[EB/OL]. [2019-5-19]. <http://www.beijing.gov.cn/ywdt/yaowen/t1587795.htm>.
- [6] 曹东云,邱婷(2018). 创造力发展:设计型学习的功能预见[J]. *电化教育研究*, 39(3) :18-22 +34.
- [7] 陈维维,李艺(2002). 信息素养的内涵、层次及培养[J]. *电*

化教育研究, (11): 7-9.

[8] CSTA (2016). K-12 Computer Science Standards [EB/OL]. [2019-06-20]. http://www.csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf.

[9] Creative Computing Curriculum(2019). [EB/OL]. [2019-08-20]. <http://scratched.gse.harvard.edu/guide/curriculum.html>.

[10] 崔允漭, 夏雪梅(2007). 试论基于课程标准的的学生学业成就评价[J]. 课程. 教材. 教法, (1): 13-18.

[11] Department for Education(2013). The National Curriculum in England [EB/OL]. [2019-6-10]. <https://www.gov.uk/government/collections/national-curriculum>.

[12] Google & Gallup(2015). Searching for computer science: Access and barriers in U. S. K-12 education. [EB/OL]. [2019-8-26]. <http://g.co/cseduresearch>.

[13] 国务院新闻办公室(2019). 新一代人工智能发展规划 [EB/OL]. [2019-05-10]. <http://www.scio.gov.cn/34473/34515/Document/1559231/1559231.htm>.

[14] 何克抗(1997). 建构主义的教学模式、教学方法与教学设计[J]. 北京师范大学学报(社会科学版), (5): 74-81.

[15] 教育部办公厅(2019). 关于印发《2019年教育信息化和网络安全工作要点》的通知 [EB/OL]. [2019-03-15]. http://laws.ict.edu.cn/laws/new/n20190313_56766.shtml.

[16] Karen, B., Laura, P., & Alexa, K. (2019). Creative Computing Curriculum Guide [EB/OL]. [2019-6-7]. https://docs.google.com/presentation/d/1TNWUgUrh-6X3vDboBNcMIHwRl7eiQHF3Q6Mth-68jo/edit?pli=1#slide=id.g4b7c0bb569_2_1759.

[17] Karp R. M. (2011). Understanding science through the computational lens [J]. Journal of Computer Science and Technology, 26(4): 569-577.

[18] 寇光杰, 邹海林, 杨洪勇(2012). 欧美计算机专业教育的特点与启示[J]. 计算机教育, (19): 102-106.

[19] K-12 Computer Science Framework Steering Committee (2016). K-12 Computer Science Framework [EB/OL]. [2019-5-1]. <https://k12cs.org/>.

[20] 李锋, 柳瑞雪, 任友群(2018). 确立核心素养、培养关键能力: 高中信息技术学科课程标准修订的再思考[J]. 全球教育展望, 47(1): 46-55.

[21] 卢蓓蓉, 尹佳, 高守林, 金凯, 廖媛, 任友群(2017). 计算机科学教育: 人人享有的机会——美国《K-12 计算机科学框架》的特点与启示[J]. 电化教育研究, 38(3): 12-17.

[22] 马若明(2004). 培养学生计算机兴趣指向和道德意识的探索: 信息技术教学实践的思考[J]. 电化教育研究, (6): 31-33.

[23] Larke, L. R. (2019), Agentic neglect: Teachers as gatekeepers of England's national computing curriculum. British Journal of Educational Technology, 50: 1137-1150. doi:10.1111/bjet.12744.

[24] 李锋, 赵健(2016). 高中信息技术课程标准修订: 理念与内容[J]. 中国电化教育, (12): 4-9.

[25] 李玉阁, 刘军(2018). 国内中小学编程教育研究现状分析[J]. 中国现代教育装备, (8): 26-29.

[26] 刘敏娜, 张倩苇(2018). 国外计算思维教育研究进展[J]. 开放教育研究, 24(1): 41-53.

[27] Nikolaos, P., & Efstratios, P. (2017). Leveraging Scratch4SL and Second Life to motivate high school students' participation in introductory programming courses: Findings from a case study [J]. New Review of Hypermedia and Multimedia, 23(1): 1-29.

[28] NMC (2017). NMC/CoSN Horizon Report (2016 K-12 Edition) [EB/OL]. [2018-10-05]. <https://www.nmc.org/publication/nmc-cosn-horizon-report-2016-k-12-edition/>.

[29] None(2004). Association for Experiential Education (AEE) [J]. Journal of Experiential Education, 27(2): 244-244.

[30] 庞维国(2011). 论体验式学习[J]. 全球教育展望, 40(6): 9-15.

[31] Papert, S. A. (1993). Mindstorms: Children, computers, and powerful ideas [M]. Basic Books: 20-22.

[32] 邱美玲, 李海霞, 罗丹, 岳赛赛, 王冬青(2018). 美国《K-12 计算机科学框架》对我国信息技术教学的启示[J]. 现代教育技术, 28(4): 41-47.

[33] Rahman, F. (2018). From App Inventor to Java: Introducing object-oriented programming to middle school students through experiential learning [C]. 2018 ASEE Annual Conference & Exposition, Salt Lake City, Utah.

[34] Rick, M. (2018). Literary Analysis Using Minecraft: An Asian American youth crafts her literacy identity [J]. Journal of Adolescent & Adult Literacy, 62(1).

[35] Sugarman, L. (1984). Experiential learning: Experience as the source of learning and development [M]. New Jersey: Prentice Hall.

[36] 孙丹, 李艳(2019). 国内外青少年编程教育的发展现状、研究热点及启示: 兼论智能时代我国编程教育的实施策略[J]. 远程教育杂志, (3): 47-60.

[37] 孙立会, 周丹华(2019). 国际儿童编程教育研究现状与行动路径[J]. 开放教育研究, 25(2): 23-35.

[38] Stufflebeam D. L. (2003). The CIPP Model for Evaluation. In Kellaghan T., Stufflebeam D. L. (eds). International Handbook of Educational Evaluation A. Springer, Dordrecht.

[39] UNESCO, NCLIS. (2013). The Prague Declaration Towards an Information Literacy Society. Information Literacy Meeting of Experts, Prague, The Czech Republic. [EB/OL]. [2019-6-20]. <http://www.unesco.org/new/fileadmin/MULTIMEDIA/HQ/CI/CI/pdf/PragueDeclaration.pdf>.

[40] Wing, J. M. (2006). Computational Thinking [J]. Communications of the ACM, 5(3): 33-35.

[41] 吴飞(2007). CIPP模式在高校课程评价中的运用[J]. 中国高等教育评估, (2): 53-55 + 69.

[42] 杨帆(2019). 初中阶段编程校本课程开发与评价研究[D]. 上海师范大学.

[43] 亿欧智库(2018). 2018 中国少儿编程教育行业研究报告 [EB/OL]. [2018-11-01]. <https://www.iyiou.com/intelligence/report565.html>.

[44]张俊列(2018). 中国课程评价研究40年:历程、主题与展望[J]. 课程. 教材. 教法,38(10):59-66.

[45]赵蔚,李士平,姜强,郎威蒙(2017). 培养计算思维,发展STEM教育:2016美国《K-12计算机科学框架》解读及启示[J]. 中国电化教育, (5):47-53.

[45]赵中建(2015). 美国STEM教育政策进展[M]. 上海:华东师范大学出版社.

[46]赵中建,周蕾(2017). 作为一门学科的计算科学:美国

《K-12年级计算机科学框架》评述[J]. 全球教育展望, 46(04):52-66.

[47]中华人民共和国教育部(2017). 普通高中信息技术课程标准(2017年版)[EB/OL]. [2019-5-23]. <https://wenku.baidu.com/view/5681a5752379168884868762caaedd3383c4b594.html>.

[48]钟启泉(2007). 课程论[M]. 北京:教育科学出版社.

(编辑:徐辉富)

Discussion on the Curriculum Standard of Chinese Youth Programming Education

SUN Dan & LI Yan

(College of Education, Zhejiang University, Hangzhou 310028, China)

Abstract: *Programming ability is considered to be one of the important skills necessary for talents in the intelligent era. As a main force of programming learning, young people's programming ability is based on the hope of the country and the future of the nation. However, compared with some developed countries, the practice and research of Chinese youth programming education is still in its infancy, and a lot of work needs to be done in terms of system integrity, content scientificity and teacher matching. At present, the urgent task of China's youth programming education is a relatively standardized curriculum standards to guide the teaching practice of programming education inside and outside the school. Based on the analysis of Computing Programmes of Study National Curriculum in England (2013), the US K-12 Computer Science Framework (2016), and the National High School Information Technology Curriculum Standards in China (2017), this study also combines the practice and research of major domestic youth programming education institutions, and explores the core contents of China's youth programming education curriculum standards, including basic concepts, theoretical foundations, core curriculum literacy, curriculum objectives and specific contents recommendations, focusing on the specific objectives of the four learning stages: entry level, beginner level, intermediate level and advanced level, and the matching content and case demonstrations, In the final part, the study put forward some suggestions for the development and implementation of the programming education curriculum. The authors hope that the study could better guide the research and practice in the field of programming education for young people through the study of curriculum standards.*

Key words: *youth; computing; programming; programming education; curriculum standards*