

人工智能背景下程序设计课程任务驱动式 思政教学探索*

邵太华** 周玉珍 余文涛 乔贵鸿

信息工程大学数据与目标工程学院, 郑州 450000

摘要 针对生成式人工智能进入课堂后程序设计学习中出现的答案依赖、过程跳步和结果弱验证问题, 对人工智能背景下课程思政的实施路径进行分析。围绕人工智能改变任务完成方式所引发的教学断点, 提出以课堂任务链重构带动课程思政落地的改革思路, 将程序设计学习过程划分为问题理解、规则辨析、流程设计、代码实现和结果验证五个环节, 并据此构建“课程模块、主要断点、任务链着力点、育人要求”相衔接的教学设计。以三种算法基本结构模块为例, 说明问题理解、流程建模、代码实现和结果核验的组织方式, 以及人工智能工具在课堂中的辅助定位。教学观察表明, 学生在任务分析、结果核验和工具使用边界等方面的过程性表现有所改善。程序设计课程中的课程思政应落实于真实编程任务的完成过程, 规则意识、责任意识、求真意识和工程规范需要通过任务过程的再组织加以实现。

关键字 人工智能, 程序设计, 课程思政, 任务驱动, C 语言

Exploring Task-Driven Ideological and Political Teaching in Programming Courses in the AI Era *

Taihua Shao** Yuzhen Zhou Wentao Yu Guihong Qiao

Institute of Data and Target Engineering
Information Engineering University
Zhengzhou 450000, China

Abstract—In response to the problems of answer dependency, process skipping, and weak result verification that have emerged in programming learning since the introduction of generative artificial intelligence into the classroom, this paper analyzes the implementation pathways of curriculum-based ideological and political education in the context of artificial intelligence. Focusing on the pedagogical breakpoints caused by AI's transformation of task completion methods, this paper proposes a reform approach that drives the integration of ideological and political education through the reconstruction of classroom task chains. The programming process is divided into five stages: problem comprehension, rule analysis, process design, code implementation, and result verification. Accordingly, a teaching design framework is constructed that aligns "course modules, key breakpoints, task chain focal points, educational objectives". Taking the three basic algorithmic structure modules as examples, this paper illustrates the organization of problem comprehension, process modeling, code implementation, and result verification, as well as the supportive positioning of AI tools. Teaching observations indicate that students have shown improvement in process-oriented performance in areas such as task analysis, result verification, and awareness of tool-use boundaries. Ideological and political education in programming courses should be implemented through the completion of authentic programming tasks. Rule awareness, responsibility awareness, truth-seeking awareness, and engineering standards need to be realized through the reorganization of task processes.

Keywords—artificial intelligence, programming courses, curriculum-based ideological and political education, task-driven approach, C language

1 引言

生成式人工智能进入程序设计课堂后, 学生获取题目解析、完整代码和纠错建议的门槛明显降低, 程序设计学习的任务完成方式随之发生变化。对初学者

而言, 这种变化一方面提高了学习支持的即时性, 另一方面也使读题分析、规则辨析、流程建模和结果核验等关键环节更容易被前置答案所替代^[1-2, 7-8]。随着新工具不再难以获取, 程序设计课程面新的教学挑战: 如何在工具广泛可得条件下保持学习过程的完整性。

程序设计课程既承担基础语法与编程能力训练任务, 也承担计算思维启蒙和学习规范养成任务。学生在变量设定、条件判断、循环控制、样例推演和边界

*基金资助: 本文得到 2026 年信息工程大学教育教学研究课题“GenAI 赋能的程序设计课程个性化教学模式探索研究”(No.JXYJ2026C074) 资助。

**通讯作者: 邵太华 shaotaihua13@nudit.edu.cn。

检查等环节中形成的规则意识、责任意识和求真态度,正是课程思政在该类课程中的主要落点^[3-6]。因此,程序设计课程的课程思政并不主要依赖知识讲授之外的附加表达,而是根植于真实编程任务的完成过程。《高等学校课程思政建设指导纲要》要求结合专业课程特点把思想政治教育有机融入教学全过程;《教育强国建设规划纲要(2024—2035年)》提出推进教育数字化和智能化。对于程序设计课程而言,二者的交汇点在于:人工智能改变了学生完成任务的方式,课程思政必须随之回到任务过程本身加以重构。

围绕程序设计课程思政建设,近期已有研究主要从思政元素挖掘、案例设计、教学组织和实践路径等方面展开探索^[3-6];围绕人工智能赋能程序设计教学,现有研究则更多关注智能答疑、学习支持、教学辅助和个性化反馈等应用价值^[1-2, 7-8]。现有成果为课程改革提供了重要参照,但两类研究大多分别展开,对人工智能如何改变程序设计任务完成方式,并进一步影响课程思政实施这一问题讨论不足。尤其在程序设计初学阶段,若学生以工具输出替代问题分析,以运行通过替代结果核验,课程思政原本依托任务过程形成的育人功能就会被削弱。如何在人工智能背景下重建程序设计课程的任务过程,使知识训练与价值引导重新在同一课堂链条中展开,已成为程序设计课程教学改革需要回应的现实问题。

基于此,依托开设的C语言程序设计课程教学实际,本文围绕人工智能背景下程序设计课程思政的任务过程断点,提出任务驱动式思政教学的重构思路,并从任务链设计、课程模块映射和典型模块实施三个层面展开论述。

2 人工智能背景下程序设计课程思政的任务断点分析

2.1 生成式人工智能重塑程序设计任务完成方式

程序设计课程的学习原本建立在较为完整的任务过程之上,学生需要围绕题意理解输入输出关系,结合数据类型和运算规则确定求解条件,再将求解思路转化为顺序、选择、循环等基本结构,最后通过样例推演和边界检查确认程序结果。这一过程既是知识掌握过程,也是思维训练过程。

生成式人工智能进入课堂后,学生解析题目、生成代码和检查边界的成本显著降低,程序设计学习的支持方式由教师主导的课堂指导逐步扩展为随时可得的工具支持。对学习积极性较高的学生而言,人工智能能够在思路提示、错误定位和结果比较等方面发挥辅助作用;但对基础尚不稳固的初学者而言,学习起点也容易由“先理解任务”转为“先调用工具”,由

“先组织思路”转为“先接受答案”。程序设计任务的完成时间被压缩,任务过程中的关键思考环节则更容易被跳过。

程序设计课程对此类变化尤为敏感。其一,编程结果具有较强的表面可见性,学生容易将“程序能够运行”直接等同于“问题已经解决”;其二,代码文本具有较强的可复制性,学生容易将“获得完整代码”误判为“已经掌握方法”;其三,初学者对错误来源的辨析能力有限,更容易把工具反馈当作现成结论。由此可见,生成式人工智能并没有取消程序设计学习中的关键环节,但明显降低了学生绕开这些环节的门槛。

2.2 程序设计课程思政的过程生成机理

程序设计课程中的课程思政往往并不依赖知识点之外的附加讲述,而是蕴含在学生完成真实编程任务的全过程之中。数据类型、常量、变量和运算规则要求表达准确、条件明确,对应的是规则意识和严谨态度;顺序、选择、循环结构要求处理步骤有序、判断条件清晰,对应的是流程意识、责任意识和审慎意识;数组、函数和指针等内容要求从整体结构、功能分解和边界控制理解程序运行逻辑,对应的是系统观念、协作观念和工程规范。

程序设计课程中的育人要求与任务过程天然契合。学生只有在读题分析中明确问题边界,在规则辨析中分清条件范围,在流程设计中组织求解步骤,在结果验证中回溯错误来源,才能真正理解程序为何成立、错误因何出现、修改应落在哪个环节。规则意识、责任意识和求真意识正是在这些步骤中逐步形成的。若将课程思政简单处理为知识讲授后的补充说明,就会脱离该课程核心的育人载体。

2.3 人工智能背景下的主要教学断点

基于上述分析,人工智能背景下程序设计课程思政面临的断点主要体现在四个方面。其一,问题理解环节被前置答案挤压。部分学生在尚未厘清输入输出、变量含义和任务要求之前,就直接向工具索取代码,导致后续学习停留在答案识别层面。其二,规则辨析环节被工具输出弱化。数据类型选择、条件范围设定、循环边界控制等原本需要学生独立判断的内容,容易被现代代码所覆盖,学生对“为何这样写”缺乏必要追问。其三,结果验证环节被运行通过所替代。程序一旦编译通过并输出结果,学生往往缺少进一步进行样例推演、反例检验和异常排查的动力,求真意识随之减弱。其四,工具使用边界不清。学生容易把人工智能视为解题主体,而不是学习辅助,由此弱化自身在分析、判断和验证中的主体责任。

因此,人工智能进入程序设计课堂后,需要重构的不是某一个孤立教学方法,而是支撑知识训练和价值引导共同发生的任务过程本身。程序设计课程思政要在新情境下保持有效性,关键在于把被工具前置压缩的学习环节重新显化,并把育人要求重新嵌入课堂任务链。

3 程序设计课程任务驱动式思政教学的重构思路

3.1 任务驱动式思政教学重构目标

程序设计课程任务驱动式思政教学的重构，不是为了排斥人工智能进入课堂，而是为了在工具广泛可得条件下重新确认学生在学习过程中的主体地位。其具体目标有二：一是恢复程序设计学习的基本过程，使问题理解、规则辨析、流程组织和结果验证重新成为学生完成任务的必要环节；二是使规则意识、责任意识、求真意识和工程规范不再停留于抽象要求，而是在任务推进中得到具体落实。

对程序设计课程而言,这一目标需要同时回应知识教学与计算思维训练两项任务。若教学设计只强调工具使用,学生容易获得局部答案而缺少完整求解过程;若只强调价值表达,又容易脱离程序设计课程的专业特征。任务链重构的意义正在于把知识学习、思维训练和价值引导统一到同一课堂过程之中。

3.2 任务驱动式思政教学重构原则

为使任务链重构真正指向上述断点,教学设计需把握四项原则。第一,真实性原则。课堂任务应来源

于具体问题情境,使知识训练和价值引导在同一求解过程中展开,避免课程思政游离于编程任务之外。第二,过程显化原则。对问题理解、规则辨析、流程设计和结果验证等关键环节提出明确要求,使学生不能只凭最终代码完成学习。第三,工具限位原则。将人工智能定位为比较、校验和纠错工具,而不将其作为替代前置分析的答案生成器,以维护学生在任务完成中的主体责任。第四,模块适配原则。根据不同知识模块的学习难点和常见跳步点,确定任务链中的着力环节,避免整门课程采用单一化处理方式。

因此,本文提出的任务驱动式思政教学旨在以完整编程任务组织知识学习、思维训练和价值引导,使学生围绕同一问题连续经历分析、设计、实现和验证各环节。上述原则落实到课堂之中,关键在于形成一个能够促使学生持续分析、判断、验证并承担结果责任的任务过程。

3.3 总体任务链设计

结合程序设计课程教学内容与学生学习特点,课堂任务链可概括为问题理解、规则辨析、流程设计、代码实现和结果验证五个环节。问题理解强调明确任务目标、输入输出关系和求解边界,对应的是问题意识和任务意识;规则辨析强调识别数据类型、表达式约束、条件范围和循环边界,对应的是规则意识和严谨态度;流程设计强调把问题转化为顺序、选择、循环等结构化步骤,对应的是逻辑意识和整体观念;代码实现强调规范表达、注释说明和结构清晰,对应的是工程规范和责任意识;结果验证强调样例推演、边界检查和异常回溯,对应的是求真意识和审慎态度。

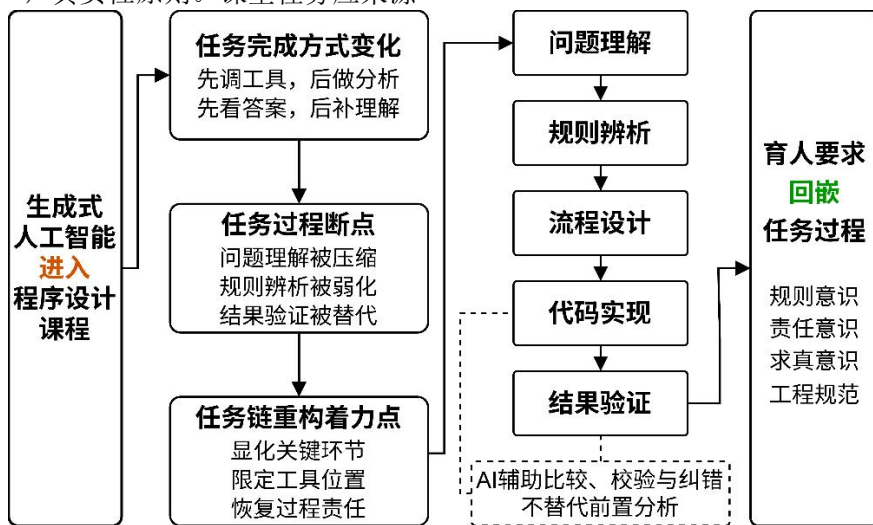


图1 AI 背景下程序设计课程育人断点与任务链重构关系图

这一任务链既保留了程序设计学习的专业逻辑，也为课程思政提供了稳定载体。人工智能可以参与其中，但只能进入学生已完成基础分析之后的比较、优化和纠错环节，而不能跳过前置思考直接替代任务完

成。图1展示了人工智能带来的育人断点与任务链重构着力点之间的对应关系。

4 面向课程模块的任务驱动式思政教学设计

4.1 课程模块与任务链着力点配置

程序设计课程以C语言为知识载体,以计算思维培养为能力主线,课程内容包括绪论、数据类型、三种算法基本结构、数组、函数和指针基本概念,并按“章节关键内容、计算思维案例库、具体计算思维训练”的方式组织教学。不同模块对应的知识性质不同,人工智能进入课堂后暴露出的学习断点也并不一致,因此任务链重构不能停留在一般性原则上,而应落实到具体模块的教学配置之中。

从课程实际看,绪论模块需要纠正学生将程序设计简单理解为语法记忆或答案调用的倾向,重点放在

问题理解环节;数据类型及运算规则模块容易出现忽视类型差异和表达规范的情况,重点放在规则辨析环节;三种算法基本结构模块既承担程序结构入门任务,又是学生最容易依赖现成代码的阶段,重点应放在流程设计与结果验证环节;数组、函数和指针等模块则分别围绕批量处理、功能分解和边界控制展开,其任务链着力点应随知识难点进行调整。只有把课程模块与任务链环节一一对应,任务驱动式思政教学才能从总体理念进入具体课堂。

4.2 课程模块与育人要求映射

围绕课程模块的学习断点、任务链着力点和育人要求,可形成程序设计课程任务驱动式思政教学的整体映射,如图2所示。

课程模块	AI背景下的主要学习断点	课程思政切入点	任务链着力点	主要育人要求
绪论	将程序学习等同于语法记忆或答案调用	程序面向真实问题 避免把学习理解为语法记忆	问题理解	科学精神、问题意识、 技术服务意识
数据类型、常量、变量、运算规则	忽视类型约束、精度差异和表达规范	类型规则、精度差异和表达式约束	规则辨析	规则意识、严谨态度
顺序、选择、循环结构	直接索取代码,弱化流程分析与结果核验	分析路径与边界条件	流程设计、结果验证	判断意识、责任意识、 过程意识
数组	将批量处理简化为模板套用	避免模板机械套用	规则辨析、代码实现	系统观念、效率意识、 数据规范意识
函数	只关注调用结果,忽视功能分解和参数传递逻辑	功能分解与参数传递逻辑	流程设计、代码实现	协作观念、整体观念
指针基本概念	机械照搬示例,忽视地址访问边界和操作风险	地址访问边界和操作风险	规则辨析、结果验证	审慎意识、边界意识、 工程规范

图2 程序设计课程模块化任务驱动式思政教学设计

图2所示映射说明,程序设计课程中的课程思政并非对每个知识点机械附加同一类价值表达,而是根据知识性质和学习断点,选择不同的任务链环节加以强化。这样处理既能保持课程思政与专业内容的内在关联,也有利于减少人工智能背景下“只见答案、不见过程”的学习偏差。

5 三种算法基本结构模块的教学实施

5.1 任务选择与教学目标

三种算法基本结构模块是程序设计课程中的关键衔接模块。学生在这一阶段第一次较系统地经历从任务分析到程序实现的完整过程,后续数组、函数等内容的学习也以此为基础。与此同时,顺序、选择、循环结构具有较为明确的代码模板特征,生成式人

工智能最容易在这一模块中以前置答案的形式介入学习过程。因此,以该模块作为教学实施示例,能够较清楚地说明任务驱动式思政教学的落点。

结合课程内容,教学中选取“学生成绩统计与等级判定”作为综合任务。任务要求输入若干名学生成绩,完成平均分计算、不及格人数统计和等级判定输出,其中既包含顺序结构中的输入输出与累加处理,也包含选择结构中的区间判断和循环结构中的重复处理。该任务的价值在于,一旦条件区间设置不当、循环边界控制失误或变量更新出现遗漏,程序输出就会直接偏离实际结果,便于学生在比较和验证中发现问题。

该任务同时具备明确的育人指向。成绩统计和等级判定涉及规则设定、结果解释和责任承担,能够

使学生在完成编程任务的过程中具体体会规则不能含混、判断必须有据、结果需要复核的基本要求。知识训练和价值引导因此能够在同一任务中展开。

5.2 教学实施过程

围绕该任务，课堂实施依次展开问题理解、规则辨析、流程设计、代码实现和结果验证五个环节。问题理解阶段，教师给出成绩统计与等级判定情境，引导学生明确输入项、输出项、统计目标和判定规则，首先解决“程序要处理什么问题”的认识问题。规则辨析阶段，要求学生说清成绩区间划分依据、关键变量含义、循环次数控制方式以及可能出现的边界情

况，避免课堂直接进入代码书写。流程设计阶段，学生以自然语言、流程图或伪代码描述处理步骤，把顺序、选择、循环三种结构落实为可以检查的求解路径。

在完成前述分析后，课堂再进入代码实现环节。学生根据既定流程独立完成程序编写，并对关键词和判断条件进行说明。人工智能工具可以在此阶段参与代码比较、语法纠错和结果核对，但不允许替代前面的任务分析和流程设计。教师关注的不是学生是否快速得到一份可运行代码，而是能否说明每一条条件判断、每一个循环边界以及每一次变量更新的依据。

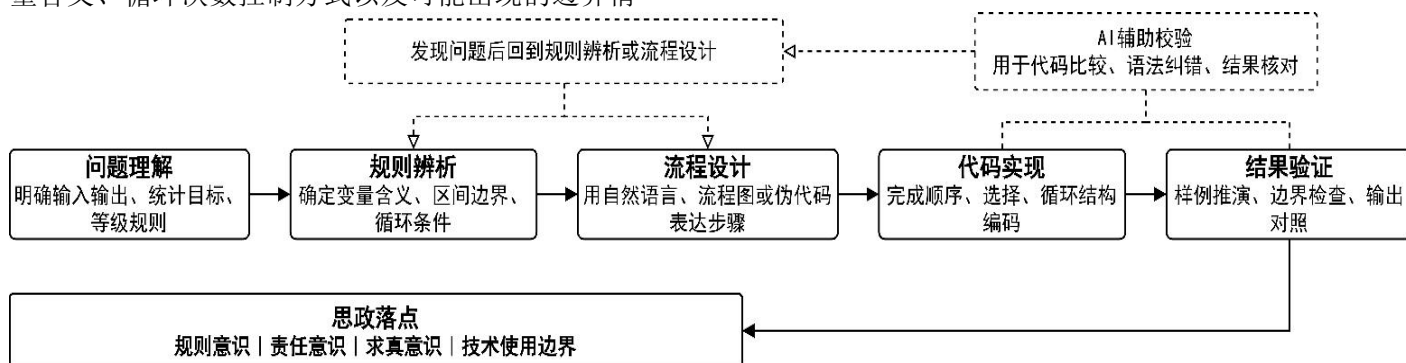


图3 顺序、选择、循环结构模块实施逻辑图

结果验证阶段，教师组织学生采用样例推演、边界检查和异常回溯等方式检验程序。学生先根据题意手工推导若干组输入下的统计结果，再与程序输出进行对照；一旦出现偏差，需要回到条件区间、循环控制或变量更新位置重新定位原因。经过这一环节，课堂评价标准由“程序能否运行”转向“结果能否解释、过程能否复核”。图3给出了该模块的实施逻辑。

5.3 育人落点分析

从实施效果看，三种算法基本结构模块中的课程思政主要落在三个层面。其一，规则意识通过条件区间划分、变量设置和边界控制得到具体呈现。学生在编程过程中会直接看到，任何模糊条件或边界遗漏都可能导致分类错误，由此形成对规则严肃性的直观认识。其二，责任意识与求真意识通过结果验证环节得到强化。成绩统计任务一旦进入真实场景，输出错误就会影响判断结论，因此程序设计者必须对结果的可靠性负责，不能以“程序能够运行”代替“结果已经正确”。其三，技术使用边界通过工具限位得以明确。人工智能可以帮助学生发现问题、比较方案和检查错误，但不能替代最基本的问题分析、流程组织和结果核验。

三种算法基本结构模块的教学实施表明，只要把前置压缩的关键环节重新纳入课堂组织，知识训练与价值引导就能够在同一任务链中实现统一。

6 教学效果分析

课程改革实施后，教学效果从学习结果、过程表现和能力迁移三个方面进行分析。与仅比较测验成绩不同，本研究更关注学生是否在人工智能辅助环境下保留必要的问题分析、规则辨析和结果验证过程。课堂观察、作业说明和阶段性测验结果显示，学生直接提交工具生成代码而不能解释关键语句的情况有所减少，能够更主动地说明变量含义、判断条件和循环边界，并通过样例推演和边界输入检查程序输出。

从学习结果看，实验班高于当次全年级均分的学生占比由第一次测验的45.57%提高到第二次的60.00%和第三次的66.25%；以第一次测验为基线，后续两次测验中相对表现改善的学生占比分别为54.43%和65.82%。这些数据说明，任务链重构并未削弱程序设计课程的知识训练要求。结合课堂过程记录和课外实践表现看，学生在综合任务中对顺序、选择、循环结构的组合使用更加稳定，能够把课堂形成的任务分析、流程表达和结果验证方法迁移到后续模块学习、程序设计类竞赛和实践任务中。相关效果见表1。

表 1 任务驱动式思政教学改革效果观察

观察维度	观察口径	主要结果	育人目标
学习成绩	每次测验高于当次全年级均分学生占比	整体呈上升趋势（45.57%→60.00%→66.25%）	学习责任、持续改进
个体改善	以第一次测验为基线的相对表现变化	第二次、第三次相对表现改善学生占比分别为54.43%、65.82%	主体意识、学习自觉
过程性学习表现	课堂提问、随堂演示、作业说明	能说明变量含义、条件边界和循环控制依据，减少只提交代码不解释过程的情况	规则意识、过程意识
任务完成质量	综合编程任务、阶段性作业	能将顺序、选择、循环结构组合用于具体情境，并通过样例推演检查结果	责任意识、求真意识
能力迁移表现	后续模块学习、竞赛与实践任务	能把任务分析、流程表达和结果验证方法迁移到数组、函数等内容及大作业实践中	工程规范、实践能力

表 1 结果表明，任务驱动式思政教学的成效既体现在学生相对学习表现的提升，也体现在完成编程任务方式的改变。学生能够在工具辅助环境下保留必要的分析、判断和验证环节，说明课程思政所强调的规则意识、责任意识和求真意识已经进入程序设计任务过程。

7 结束语

围绕人工智能进入程序设计课堂后，对学生进入任务、完成任务和理解任务的方式带来的影响，提出了任务驱动式思政教学的重构思路，并从课程模块映射和三种算法基本结构模块实施两个层面给出了具体做法。研究表明，程序设计课程思政改革不能把重点仅放在工具接入上，而应把任务过程重新组织起来，使规则意识、责任意识、求真意识和工程规范在真实编程任务中得到落实。这也是人工智能背景下程序设计课程保持育人实效的关键所在。

参 考 文 献

[1] QIAN Y. Pedagogical Applications of Generative AI in Higher Education: A Systematic Review of the Field[J]. TechTrends, 2025, 69:1105-1120.

[2] 郝祥军, 顾小清. 人机协同学习的新理解: 人与 AI 共同调节学习[J]. 中国电化教育, 2025(06):9-17+79.

[3] 毛贱, 吉清凯, 赵达. 程序设计课程思政教学与案例设计——以枚举为例[J]. 计算机技术与教育学报, 2022, 10(1):61-65.

[4] 殷建军, 陈湘骥, 梁云. 智能时代高级语言程序设计课程思政教学实践[J]. 计算机教育, 2025(06):106-112.

[5] 杨鑫, 崔志凌, 王宇新, 等. 通识教育课程中的思政教育探索与实践: 以 C 语言程序设计课程为例[J]. 计算机教育, 2023(04):163-167, 172.

[6] 姜洋, 张志远, 何志学. 程序设计课程思政系统化教学模式探索[J]. 计算机教育, 2022(10):106-109.

[7] 郝熙, 潘晟旻. 面向 GenAI 的师-机-生三元共生程序设计教育模型研究与实证[J]. 计算机技术与教育学报, 2025, 13(3):18-23.

[8] 朱俊华, 许璐瑶, 马近远. 生成式人工智能如何赋能学生学习: 基于大学生自我调节学习行为的实证研究[J]. 高等工程教育研究, 2025(02):66-72.