

Vibe Coding 赋能非计算机专业编程教学改革

石佳琳

北京科技大学计算机与通信工程学院,

北京 100083

摘要 当前面向非计算机专业的基础编程语言课程普遍存在语法导向过重、实践脱离专业、学习门槛高、学习兴趣不足等问题,难以适配 AI 时代复合型人才需求。Vibe Coding 作为以自然语言意图驱动、AI 辅助生成、快速原型迭代为核心的新型编程范式,为非计算机专业编程教育提供了破局路径。本文面向经管、文法、生化、能源等非计算机专业,构建融入 Vibe Coding 的基础编程语言课程改革体系,提出“基础语法、Vibe 范式、专业应用”三层递进培养方案。提出“看、仿、创、评”四位一体教学模式、跨学科融合师资与“过程、项目、答辩”多元化考核。经教学实践验证,可显著降低编程学习门槛、提升学习兴趣与专业应用能力,为非计算机专业计算机基础教育改革提供参考。

关键字 非计算机专业,基础编程语言,教学改革,Vibe Coding, AI 辅助编程,计算思维

"Vibe Coding" Empowers Reform in Programming Education for Non-Computing Majors

Jialin Shi

University of Science and Technology Beijing,
Beijing 100083, China;
shijl@ustb.edu.cn

Abstract—Introductory programming courses for non-computer science majors currently face issues such as an excessive focus on syntax, a disconnect between practice and the students' specific disciplines, high barriers to entry, and a lack of learner engagement; consequently, they struggle to meet the demand for interdisciplinary talent in the AI era. "Vibe Coding"—a novel programming paradigm centered on natural language intent, AI-assisted generation, and rapid prototyping—offers a breakthrough solution for programming education in non-computing disciplines. This paper proposes a curriculum reform framework integrating Vibe Coding for majors in fields such as economics and management, liberal arts and law, biosciences and chemistry, and energy. It outlines a three-tiered progressive training scheme comprising "basic syntax," the "Vibe paradigm," and "domain-specific applications." Additionally, the paper introduces an integrated "Observe-Imitate-Create-Evaluate" teaching model, an interdisciplinary faculty approach, and a diversified assessment system covering "process, projects, and presentations." Practical implementation demonstrates that this approach significantly lowers the barrier to learning programming, boosts student interest and application capabilities, and serves as a valuable reference for reforming foundational computer education for non-computer science majors.

Keywords—Non-computer science majors, introductory programming languages, teaching reform, Vibe Coding, AI-assisted programming, computational thinking

1 引言

在数字化与人工智能普及背景下,编程能力与计算思维已成为各专业人才必备素养[1]。面向非计算机专业开设的 Python、C++ 语言等基础编程课程,是培养学生数字化能力的核心载体[2, 3]。但传统教学存在明显局限:以语法讲解为中心、重理论轻应用、案例与专业脱节、考核方式单一,导致学生畏难、厌学、学不会、用不上,难以支撑专业领域的数字化创新需求[4, 5, 6]。

Vibe Coding 是近年来兴起的 AI 驱动新型编程范式[7-8]。核心是用自然语言描述需求与“感觉”,由 AI 生成、调试、优化代码[9],开发者聚焦意图表达与效果验证,大幅降低代码编写门槛,特别适合非计算机专业快速实现专业场景小工具、数据处理、可视化原型等需求。

本文针对非计算机专业学情与培养目标,将 Vibe Coding 融入基础编程语言课程,从培养方案、教学模式、师资建设、考核体系四方面实施改革见图 1,探索低门槛、强兴趣、重应用、融专业的编程教学新路径。

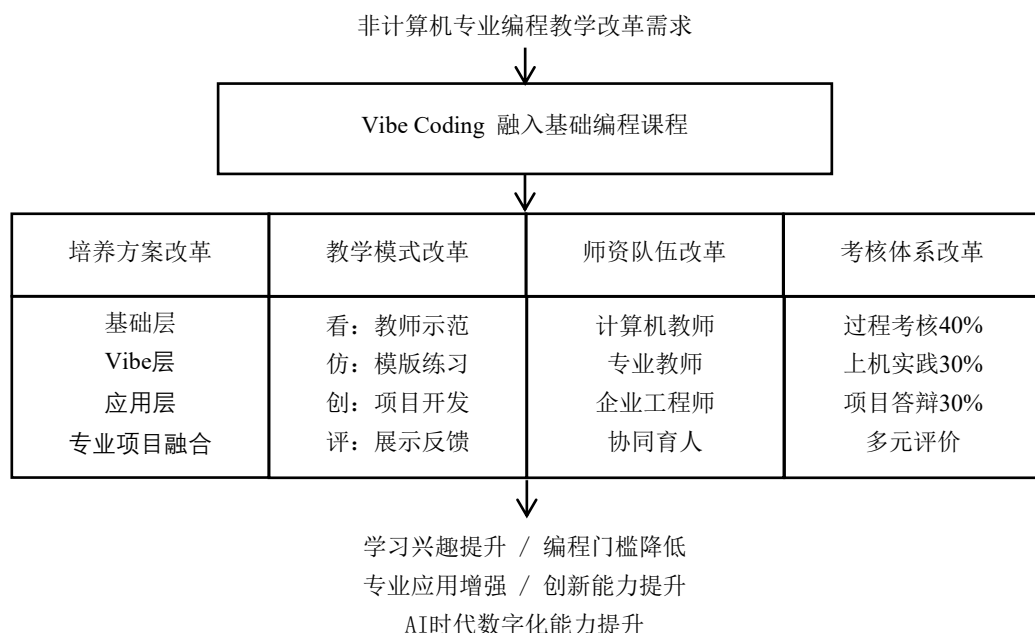


图 1 教学改革框架

2 非计算机专业基础编程教学现存问题

(1) 教学内容错位。沿用计算机专业体系，语法占比过高，抽象概念密集，非专业学生理解困难、获得感低。

(2) 实践与专业脱节。案例多为通用数学题、简单算法，与经管数据分析、文法文本处理、能源数据处理等真实需求无关，学以致用不足。

(3) 学习门槛过高。要求手写代码、逐行调试、严格规范，学生易因语法错误挫败，丧失兴趣。

(4) 考核方式单一。以期末笔试、机考为主，重记忆轻应用，难以评价创新与解决实际问题能力。

(5) 缺乏 AI 时代适配。未引入 AI 辅助工具，与当前真实开发场景脱节，学生进入岗位后难以快速适应。

3 融合思路框架设计与实施

Vibe Coding 融入非计算机专业编程教学的适配性。Vibe Coding 以意图驱动、自然语言交互、AI 生成代码、快速原型迭代为特征，与非计算机专业编程教学高度契合。

降低入门门槛，学生不必死记语法，用自然语言描述需求即可生成可用代码，快速获得成就感。聚焦问题解决，学生专注专业逻辑与需求，而非底层实现，符合“用编程解决专业问题”的教学目标。提升学习兴趣，所见即所得，快速产出可视化、可交互作品，激发创作与探索欲。衔接真实场景，可以与 Cursor、

GitHub Copilot、Replit AI 等工具一致，培养 AI 时代必备的提示工程与代码迭代能力。适配多专业应用。可快速支撑数据统计、文本分析、表单工具、交互原型、实验数据处理等跨领域需求。

4 融入 Vibe Coding 的课程教学改革实践

4.1 提出“基础—Vibe—应用”三层递进培养方案

以“会用、能用、好用”为目标，构建三层递进内容体系见表 1，平衡基础语法与新型范式。

第一为基础层，即核心语法最小集。精简传统语法，只保留变量、数据类型、分支循环、函数、文件读写、常用库等必需知识点，以够用、实用为原则，配合简单案例快速上手，为理解 AI 生成代码打下基础。

第二为进阶层，即 Vibe Coding 范式与工具。讲授 Vibe Coding 核心逻辑：意图描述→提示词工程→AI 生成→迭代优化→代码解读。训练学生把专业需求转化为清晰自然语言指令，使用 AI 编程工具生成、修正、解释代码，读懂关键代码、调整参数、适配场景。

第三为应用层，专业场景融合项目。按专业分类设计项目：经管类侧重成绩统计、财务分析、可视化报表；文法类侧重文本词频、关键词提取、简单问答工具；设计类侧重交互原型、动态图表；能源和环境类侧重实验数据处理、拟合绘图、批量处理。学生用基础语法 + Vibe Coding 完成完整项目，实现学用一体。

表 1 教学改革框架

层次	教学目标	主要内容	典型任务
基础语法层	能读懂、能修改基础代码	变量、分支、循环、函数、文件读写、常用库	读取文件、计算统计量、绘制简单图表
Vibe 范式层	能用 AI 辅助生成和迭代程序	提示词设计、代码生成、错误修复、代码解释	根据自然语言生成数据处理程序
专业应用层	能解决专业场景问题	专业案例、项目开发、结果展示	经济数据分析、文本处理、实验数据拟合

4.2 “看—仿—创—评”四位一体立体化教学模式

采用案例驱动、任务导向、AI 辅助、小组协作的立体化模式，突出非专业学生低负担、高产出。

(1) 看：教师示范 Vibe Coding 完整流程，从一个专业问题出发，示范如何拆解需求、编写提示词、调用 AI 工具生成代码、运行程序、分析报错、优化结果。例如，教师可展示“读取学生成绩表并生成成绩分析报告”的完整过程，使学生直观理解自然语言意图如何转化为程序结果。

(2) 仿：课程为学生提供提示词模板、代码框架和任务清单，学生在模板基础上替换数据、修改需求、调整输出形式。该阶段的重点是降低第一次实践的难度，使学生建立基本信心。

(3) 创：学生以 3—5 人为小组，围绕本专业真实问题自主选题，使用基础语法和 Vibe Coding 工具完成一个具有完整功能的小型项目。项目要求包括需求说明、提示词记录、代码文件、运行结果、改进过程和项目报告。

(4) 评：课程通过课堂展示、小组互评、教师点评和答辩评价等方式，对学生项目进行综合评价。评价内容不仅包括代码是否能运行，还包括问题是否真实、需求是否清晰、提示词是否有效、结果是否可信、专业解释是否合理。

4.3 跨学科融合高水平师资队伍

融入 Vibe Coding 的编程教学不仅需要计算机教师讲授语法和工具，也需要专业教师提供场景和问题。本文构建“计算机教师+专业教师+企业导师”的协同教学团队。计算机教师主要负责基础语法、AI 工具使用、代码规范和程序调试；专业教师负责提供专业应用场景、数据样例和结果解释；企业导师可结合

实际 workflows 介绍 AI 辅助开发、数据分析和自动化办公中的典型应用。

同时，课程建设专业案例库，按照专业方向划分为经管数据分析类、文法文本处理类、生化实验数据类、能源环境监测类、通用办公自动化类等模块。每个案例包括问题背景、数据文件、任务要求、参考提示词、参考代码、拓展方向和评价标准。案例库既可用于课堂教学，也可作为学生项目选题来源。

4.4 “过程、项目、答辩”多元化考核体系

为改变传统课程重期末、重语法、轻应用的问题，本文构建全过程、多维度的课程评价体系。考核内容由过程考核、上机实践和项目答辩三部分组成见表 2 所示。

(1) 过程考核 (40%)：随堂练习、课后作业、AI 工具使用、课堂互动，检验学习态度与知识掌握。

(2) 上机实践 (30%)：基于 Vibe Coding 的小任务，如数据处理、小工具、可视化，检验动手与意图表达能力。

(3) 项目答辩 (30%)：小组完成专业综合项目，提交源码、报告和演示 PPT，随机抽人答辩，确保全员参与。

表 2 多元化考核指标设计

考核模块	比例	评价重点
过程考核	40%	课堂参与、随堂练习、提示词记录、代码理解
上机实践	30%	任务完成度、程序运行结果、调试能力
项目答辩	30%	专业适配度、创新性、展示表达、团队协作

5 教学改革效果

经多专业、多轮教学实践，融入 Vibe Coding 的基础编程课程取得明显成效。首先，学生对于基础编程语言的畏难情绪下降，课堂参与度、完成率大幅提高，提升学生的学习兴趣。其次，零起点学生可快速实现可用作品，获得感与自信心增强，有效降低编程门槛。再次，项目紧密结合专业，学生能用编程解决真实问题，知识迁移能力提升，专业融合度提高。然后，团队协作、需求表达、问题解决、创新思维得到锻炼，学生的综合能力全面发展。最后，掌握提示工程、AI 辅助开发、代码迭代等实用技能，就业竞争力增强，培养学生适配 AI 时代能力。

5.1 学生学习兴趣与课堂参与度

从课堂观察和问卷调查结果见表 3 所示，融入 Vibe Coding 后，学生对编程课程的兴趣明显提高。传统教学中，部分学生由于语法错误频繁、调试困难和案例抽象，对课程存在畏难情绪。改革后，学生可以通过自然语言快速生成初始程序，并在教师指导下逐步修改和完善，学习过程由“长时间写不出代码”转变为“先看到结果，再理解和优化代码”，学习体验明显改善。

表 3 改革前后学生学习体验问卷结果

指标	改革前	改革后	提升幅度
对编程课程感兴趣的学生比例	48.6%	82.3%	+33.7%
认为编程“难以入门”的学生比例	67.5%	31.8%	-35.7%
课堂主动参与讨论的学生比例	42.1%	76.4%	+34.3%
认为课程对本专业有帮助的学生比例	51.2%	87.6%	+36.4%
愿意继续学习编程工具的学生比例	46.8%	79.5%	+32.7%

根据表 3 可以看出，改革后学生对课程价值的认可度显著提升。其中，“认为课程对本专业有帮助”的学生比例由 51.2% 提升至 87.6%，说明专业案例与 Vibe Coding 的融合增强了学生对编程应用价值的理解。

5.2 作业完成率和实践任务

在传统教学模式下，学生作业主要围绕语法题和简单算法题展开，部分学生因调试困难无法独立完成任务。融入 Vibe Coding 后，教师将作业设计为“需求描述—AI 生成—代码解读—结果验证—功能改进”的完整流程，学生不仅需要提交代码，还需要提交提示词记录和运行结果。该方式降低了初始编写难度，同时提高了学生对代码逻辑和任务结果的关注。

5.3 课程成绩与项目质量

课程组进一步比较了改革前后学生课程成绩分布和项目评分情况。结果显示，改革后学生平均成绩提高，不及格率下降，优秀率上升。尤其是在项目类任务中，学生作品的完整性、专业贴合度和展示质量均优于传统教学模式。

5.4 学生计算思维与 AI 协作能力

Vibe Coding 教学并不是弱化计算思维训练，而是将计算思维训练嵌入真实任务中。学生在完成项目时，需要经历问题抽象、任务分解、数据组织、流程

设计、结果验证和迭代优化等环节。这些环节本质上与计算思维中的分解、抽象、算法设计和自动化思想相一致。

同时，改革后学生逐渐掌握了 AI 辅助编程的基本方法，包括如何编写清晰提示词、如何要求 AI 解释代码、如何根据报错信息进行多轮追问、如何判断生成结果是否符合需求、如何修改参数和功能。学生从“被动接受代码”转变为“主动提出需求、判断结果并优化程序”，人机协同能力明显增强。

6 结语

将 Vibe Coding 新型编程范式融入非计算机专业基础编程语言教学，是顺应 AI 技术发展、贴合非专业学情、回归应用导向的重要改革。通过三层递进培养、四位一体教学、跨学科师资、多元化考核，实现从“教语法”到“育能力”、从“难学枯燥”到“易用有趣”、从“通用脱离”到“专业融合”的转变，有效提升学生计算思维与数字化创新能力。未来可进一步完善专业案例库、优化 AI 教学工具、深化跨学科协同，为培养新时代复合型人才提供更强支撑。

参考文献

[1] 韩帅, 吕继光, 谢晓芹, 等. 面向新工科的计算思维与问题求解实践教学机制探索[J]. 经济与社会发展研究, 2025(10):0241-0243.

[2] 张芯瑜. 新工科背景下面向非计算机专业的 Python 与 AI 融合教学改革[J]. 计算机应用文摘, 2024, 40(23):39-40.

[3] 李德华. 高校非计算机专业 Python 课程教学改革路径研究[J]. 河南财经学刊, 2025, 39(6):92-96.

[4] 周恒, 田春娜, 吴小俊. AI 辅助编程工具的教育应用与伦理挑战[J]. 计算机技术与发展, 2025, 35(12):123-130.

[5] 崔静, 张所娟, 陈静. AI 赋能的编程类课程教学模式构建与实践[J]. 计算机技术与发展, 2025, 35(12):165-173.

[6] 刘唤唤, 葛斌, 陈辉. 非计算机专业《计算思维导论》教学方法的探讨与实践[J]. 计算机技术与教育学报, 2021, 9(1):46-48.

[7] Ray P P. A review on vibe coding: Fundamentals, state-of-the-art, challenges and future directions[J]. Authorea Preprints, 2025.

[8] Meske C, Hermanns T, Von der Weiden E, et al. Vibe coding as a reconfiguration of intent mediation in software development: Definition, implications, and research agenda[J]. IEEE Access, 2025, 13: 213242-213259.

[9] Geng F, Shah A, Li H, et al. Exploring student-AI interactions in vibe coding[C]//Proceedings of the 28th Australasian Computing Education Conference. 2026: 45-54.