

# AI编程背景下程序设计类课程教学改革研究

徐宁, 樊郁徽

淮南师范学院计算机学院, 安徽 淮南

收稿日期: 2026年2月23日; 录用日期: 2026年3月18日; 发布日期: 2026年3月27日

## 摘要

随着大型语言模型(Large Language Model, LLM)技术的突破性发展, 人机协作、AI编程的软件开发模式对高校人才培养提出了新要求。文章在对氛围编程(Vibe Coding)的概念、特征、工作流程进行探讨的基础上, 提出将其与任务驱动教学模式相结合, 并在程序设计类课程中进行融合应用。具体而言, 在编程任务的完成过程中采用“需求描述→AI编程→测试验证→优化迭代→知识内化”的流程。教学实践表明, 该模式改变了学生的程序设计思维模式, 显著提升了学生的学习兴趣、创新能力和AI素养, 为数智时代的编程教学改革提供了理论与实践参考。

## 关键词

大语言模型, 氛围编程, 阶梯式任务, 程序设计课程

# Exploration of Teaching Reform for Programming Courses in the Context of AI Coding

Ning Xu, Yuhui Fan

School of Computer Science, Huainan Normal University, Huainan Anhui

Received: February 23, 2026; accepted: March 18, 2026; published: March 27, 2026

## Abstract

With the breakthrough development of Large Language Model (LLM), the AI programming-based software development model characterized by human-machine collaboration has raised new requirements for talent cultivation in higher education institutions. Based on an exploration of the connotation, characteristics, and workflow of Vibe Coding, this paper proposes the integration of Vibe Coding with the task-driven teaching model and its application in programming courses. The process

of completing programming tasks adheres to the workflow of “requirement description → AI code generation → testing and feedback → optimization and iteration → knowledge internalization”. Teaching practice demonstrates that this model has reshaped students’ programming thinking, significantly improved their learning interest, innovative competence, and AI literacy, and thus provides theoretical and practical references for the reform of programming teaching in the digital and intelligent era.

## Keywords

Large Language Model, Vibe Coding, Stepwise Learning Task, Programming Course

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

随着 AI 技术的飞速发展,大语言模型正在重塑软件开发流程,从传统的手写代码,到 AI 辅助编程,再到基于 LLM 编程智能体的出现,软件开发开始进入到“自然语言描述需求, AI 自动生成完整代码”的高效开发模式阶段。与传统开发模式相比, AI 编程效率高、门槛低、代码准确率高。腾讯发布的《2025 腾讯研发大数据报告》中首次系统披露 AI 已全面融入腾讯研发体系,超过 90% 的腾讯工程师使用 AI 编程助手 CodeBuddy 辅助编程,50% 的新增代码由 AI 辅助生成。Stack Overflow 发布的 2025 年度开发者调查报告也指出“84% 的受访者正在使用或计划使用 AI 工具,较去年进一步提升” [1]。AI 编程智能体能够自动完成生成代码、进行错误检测和自动化测试等任务,极大地提高了开发效率和质量。这些功能使开发者能够更专注于解决复杂问题和设计创新解决方案。

软件开发模式的改变,对从业人才提出了超越传统编程专业知识的新要求:一是构建提示词能力,即建立既能表达意图又包含约束条件的提示词[2];二是需求分析与任务拆解能力,能将复杂问题拆解为更小、定义明确的子任务,更好地引导 AI 完成工作;三是代码审查与验证能力, AI 生成代码中可能包含细微的漏洞、安全隐患以及虚构的 API 接口,开发者需要具备判断代码正确性、安全性和可维护性的能力;四是架构设计与系统思维能力,能够从宏观角度把控项目结构而非局限于具体语法实现。

围绕在数智时代如何利用 AI 提高程序设计类课程的教学效果这一问题,学者们从教学内容、教学方法、教学主体、教学模式等方面进行了变革。实践研究证明,在 AI 辅助下,学生的编程能力都得到有效提升[3]-[6]。相关调查显示,超过 84% 的学生在编程学习中已开始使用大模型辅助工具,但是在已有研究中,LLM 和 AI 编程智能体通常扮演“智能助教”的角色,解决教学资源建设及学生在代码编写中遇到的基础问题,如语法错误提示、简单算法思路和代码提示[7]。教学目标仍然将重点放在语法记忆和手工编码能力训练上,对产业人才需要的新增能力如何培养这一问题,尚未得到有效解决。本研究尝试将 Vibe Coding 与阶梯式任务驱动教学模式相融合,探寻契合智能时代编程能力培养需求的课程解决方案。

## 2. 氛围编程概述

### 2.1. Vibe Coding 的定义

OpenAI 联合创始人 Andrej Karpathy 于 2025 年初正式提出 Vibe Coding 这一概念。在中国科学院计算研究所发布的关于氛围编程的综述报告中,将其定义为一种基于大型语言模型的软件开发模式,其核

心在于建立人类开发者、软件项目和编码智能体之间的动态三元关系[8], 主要理念是开发者通过自然语言描述需求, AI 基于 LLM 的上下文理解能力, 能够理解项目结构、代码逻辑和开发者意图, 自动生成符合项目需求的完整代码。工作过程如图 1 所示:

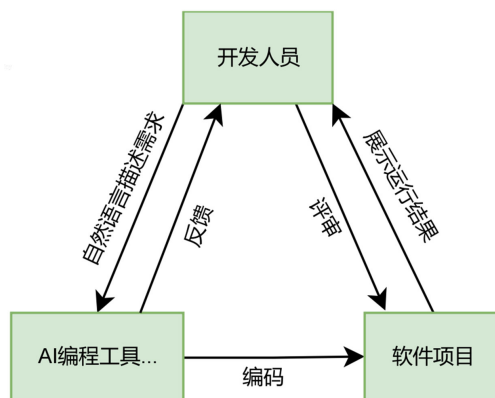


Figure 1. Vibe Coding workflow  
图 1. 氛围编程流程

由图可知, 氛围编程强调开发者通过自然语言描述需求, 只关注结果不关注代码细节, 由 AI 生成代码, 开发者通过观察执行结果而非逐行理解代码来验证方案, 形成“需求描述→AI 代码生成→验证结果→反馈优化”的快速迭代循环。

## 2.2. Vibe Coding 的应用场景

根据人类控制程度和结构化约束的不同, Vibe Coding 可分为五种开发模式[8], 各模式的特点和适用场景如表 1 所示:

Table 1. Five development models of Vibe Coding  
表 1. Vibe Coding 的五种开发模式

开发模式	特点	适用场景
无约束自动化模型(UAM)	AI 自主生成代码, 开发者验收成果	简单脚本、快速原型开发
迭代对话协作模型(ICCM)	人机通过自然语言对话, 迭代优化代码	需求模糊的探索性开发
计划驱动模型(PDM)	先制定开发计划, 再分步实现	复杂系统、模块化开发
测试驱动模型(TDM)	以测试用例为核心, 用测试结果引导生成	高可靠性软件
上下文增强模型(CEM)	丰富的项目上下文, 精准理解需求	大型代码库、企业级应用

考虑到程序设计课程教学对象一般是大学一、二年级学生, 编程经验不足甚至没有编程经历, 因此作为教学用的项目, 一般选择贴近生活、解决真实问题的案例, 以便激发学生的兴趣; 同时要求项目规模比较小, 结构太复杂不符合学生的认知规律, 也容易让学生产生挫败感。因此, 在 Vibe Coding 引入教学环节时, 可以采用无约束自动化模型或迭代对话协作模型开发模型, 通过多轮人机对话获得满足项目要求的程序代码。

## 2.3. Vibe Coding 在编程教学中的优势

传统软件开发中, 由程序员完成代码编写, 实现功能, 依赖开发者对语法规则的熟练掌握。相应地,

程序设计类课程的教学重点放在了语法知识的掌握, 教学流程遵循“语法学习→代码编写→调试运行”的线性路径, 有限的教学课时往往被用于语法讲解和练习。

在 AI 承担编码工作的背景下, 程序设计类课程的教学目标需相应调整: 学生需要掌握核心的语法知识和编程的思维, 但无需纠结语法上的细节。Vibe Coding 应用于教学过程时, 学生可以将 AI 生成的完整代码结合运行效果, 反向理解语法知识点, 减少了因调试代码带来的负担; 在这种模式下, 学习者节省了逐行书写代码的时间, 才能将精力聚焦在真实问题拆解和提示词的描述上, 通过多个项目的练习, 逐步锻炼问题抽象和提示词编写能力, 建立全新的编程思维模式, 更适合现在软件开发人员的能力需求。值得注意的是, Vibe Coding 可以显著提升学生的学习兴趣, 这是因为输入自然语言描述的功能需求, 就可以得到一个可以运行的程序。例如, 仅需描述“完成一个贪吃蛇游戏”, 即可借助 AI 生成可运行的小游戏, 这种“即时反馈”的编程体验能有效打破“编程 = 死记语法 = 亲手敲代码”的刻板印象, 让学生直观感受到解决实际问题的乐趣。

由此可见, Vibe Coding 适配编程入门阶段的教学需求, 可以实现培养知识目标、能力目标与 AI 素养的有机统一。

### 3. 基于氛围编程的阶梯式任务驱动教学模式

#### 3.1. 教学模式整体结构

任务驱动教学模式是基于构建主义理论, 以任务为主线、教师为主导、学生为主体的教学模式, 核心是将教学目标融入任务中, 引导学生在完成任务的过程中主动构建知识, 提升能力。阶梯式任务是在教学中以一个或多个任务的不断拓展和层层推进来带动课程的学习[9]。程序设计教学过程中在教师的引导下, 由学生作为主体采用 Vibe Coding 方式完成编程任务, 自主探究技能和知识的构建, 形成了基于 Vibe Coding 的阶梯式任务驱动教学模式。

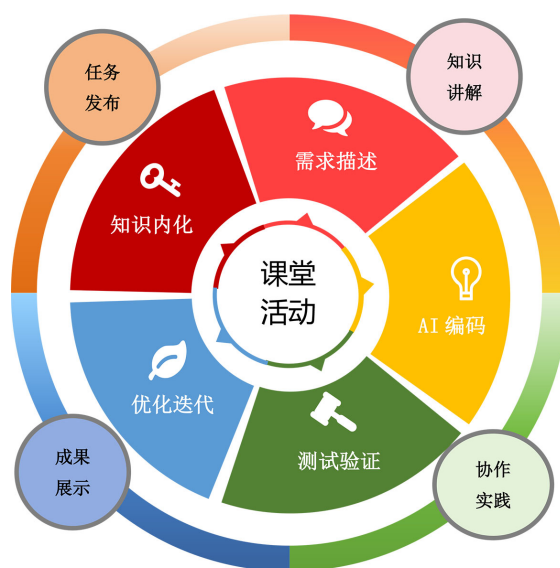


Figure 2. Classroom teaching process

图 2. 课堂教学流程

课程教学流程采用双循环结构, 如图 2 所示, 外圈表示课堂进度, 按“任务发布、知识讲解、协作实践、成果展示”四个环节进行; 内循环表示人机协作实践完成任务的步骤, 课堂中每个教学任务在 Vibe

Coding 流程的基础上, 增加了回答问题和反向分析代码这一步骤, 通过该步骤促进学生掌握本次课程任务所包含的课程知识目标。因此, 任务流程总共设计为五个阶段: “需求描述-AI 编程 - 测试验证 - 优化迭代 - 知识内化”。

### 3.2. 教学活动实施过程

在具体课程的课堂教学实施过程中, 通过教师给定的教学案例, 完成从需求分析到项目实现的过程。教师作为引导者, 以学生为主体, 使用 AI 编程工具作为编程伙伴。

在任务发布阶段, 教师明确项目任务, 引导学生进行需求分析, 模块拆解, 并通过自然语言描述需求; 然后给出提示词模板作为脚手架, 学生根据自己描述的需求, 进行提示词设计。

在知识讲解阶段, 教师以知识图谱的形式展示本次项目所牵涉的知识点, 具体内容在学生反向分析代码时自主构建; 然后教师对项目开发过程中可能用到的工具进行介绍, 并演示使用方法。

协作实践阶段是教学活动的重点环节。首先, 学生通过提示词让 AI 编程工具完成代码编写; 其次, 在 AI 生成基础代码后, 学生对代码进行审查和编写测试用例进行测试, 通过和 AI 的多轮交互, 直到程序运行效果符合项目需求; 再次, 在优化迭代阶段, 学生需自主开展思考并增添程序功能, 自行设计提示词, 通过这个环节强化与人工智能协作的能力。最后, 学生依据教师预先设定的问题, 逆向分析实现各功能的代码, 并对照知识图谱中的知识点, 理解其在程序中的功能及运用方式, 从而切实掌握本次课程所涉及的知识与技能。

在成果展示阶段, 学生展示最终运行效果, 重点阐述优化迭代环节新增功能的设计思路、AI 交互提示词、代码审查测试阶段发现的问题及解决方案。

在课后, 为巩固本次所学知识和技能, 学生需要完成一个与本次课程知识点相同的项目作为作业, 作业项目按照不同的难度级别设置成多个层次, 以适应不同能力的学生。学生在提交代码时同步提交“项目过程文档”, 文档需要包含向 AI 提出的需求描述、对 AI 生成代码的关键修改及原因、测试用例设计思路三方面内容。教师以此为依据, 可以考察学生的需求分析能力、代码审查能力和测试能力。

## 4. 基于氛围编程的阶梯式任务驱动教学模式的课程实践

本项目以 32 课时的 Java 语言程序设计课程为研究对象。鉴于 Vibe Coding 的编程思想与传统方式有很大不同, 因此在教学目标、教学内容等方面都做了相应调整。

### 4.1. 教学目标重构

依据本课程的专业基础课定位和实际需求, 将教学目标重新划分为三层:

- 1) 知识目标, 掌握 Java 核心语法与面向对象设计原则;
- 2) 能力目标, 熟练运用 AI 编程工具, 具有用 Vibe Coding 方式解决问题的能力, 其中包括需求分析、问题拆解、提示词工程、代码审查与调试;
- 3) AI 素养及思政目标, 培养 AI 协作工作能力和 AI 伦理意识, 具有工匠精神、科学态度与终身学习意识。

### 4.2. 教学内容的更新与重组

AI 驱动的编程, 对开发者来说最重要的不是程序功能如何实现, 而是如何与 AI 协同高效工作, 为此, 在原有编程语言核心语法知识点的基础上, 还要掌握 Vibe Coding 所需的工具链和知识体系。以 Java 语言程序设计课为例, 更新后的教学内容划分如表 2 所示:

**Table 2.** Teaching content  
**表 2.** 教学内容

知识能力模块	具体内容
Java 基本语法	Java 语言基础; 集合、常用 API、异常处理
面向对象知识	面向对象基础、封装、继承、多态; 接口、抽象类、内部类
数据存储	文件读写、数据库访问、多线程
编程思维和调试	Vibe Coding 编程范式; 模块化; 代码调试
开发和版本控制工具	IDE 和插件的安装使用; Git; GitHub 或 Gitee 的使用
Maven 和依赖管理	Maven 项目结构; 依赖引入和版本管理
代码测试	编写、运行单元测试用例; 简单集成测试
Prompt 工程技巧	精准提需求(明确技术栈、规范和异常处理)
AI 工具使用	使用 AI 编程工具(GitHub Copilot/Trae 等)

Trae、Git 等工具的使用, 通过提示词与 AI 工具交互的技能融入每次教学任务的完成过程中; Java 语法部分的教学内容则拆分成流程控制、数组、集合、封装等 19 个知识点, 同时将一个教学案例项目分解成 6 个阶梯式教学任务, 使 19 个知识点有机融入这 6 个教学任务中。

### 4.3. 阶梯式教学任务设计

教学过程的实施是在项目任务的驱动下进行的, 任务的设计是教学设计的核心环节。教学任务必须紧扣教学目标, 阶梯化任务的设置必须遵守由易到难、难度平滑上升的原则, 即每一阶段任务的难度增量小而可控, 要符合学生的认知规律。同时, 阶梯任务最好围绕同一主题, 这样所有任务最终形成的项目是内容相关、逻辑衔接的整体, 后一任务需在前一任务的基础上延伸、拓展, 方便学生在完成任务的过程中逐步构建系统化的知识, 而非碎片化的知识点。

基于以上原则, Java 程序设计课程可以选择一个核心项目为载体, 利用“知识点递进 + 任务迭代”的方式, 通过几个由易到难的项目任务, 完成由控制台极简版到最终版的项目迭代。在本项目的课程教学实践中, 考虑到趣味性和学生零编程基础的情况, 选择了以“背单词小游戏”作为核心项目, 分为六个教学任务, 任务内容及涵盖的教学目标如图 3 所示:



**Figure 3.** Teaching tasks and teaching objectives of the Java courses

**图 3.** Java 课程的课程教学任务及教学目标

以“背单词小游戏”项目的第二次任务为例，可以设计如下：

- 1) 任务内容：利用面向对象知识重构控制台简易版程序。
- 2) 任务目标：知识目标为通过此任务掌握面向对象编程的三大核心思想；了解对现实生活中的事物进行抽象的方法和步骤；Java 中类的定义格式；利用访问限定符对类进行信息隐藏；利用公有接口访问隐藏信息；对象的使用。能力目标包括练习使用一种绘图工具；学会用绘图工具绘制图；根据模板编写提示词。
- 3) 任务完成标准：从程序存储的单词库中，随机抽取中文单词让用户拼写英文；具有判断功能，根据判断结果给出提示；统计答题正确率；支持用户输入“exit”时手动退出。
- 4) 提示词结构和模板：提示词应包含“功能需求、类设计要求、技术约束”三部分。提示词模板如图 4 所示：

```
# 提示词模板：
请你用 Java 语言开发一个【开发目标】，严格遵循以下所有要求，生成完整、可直接运行的代码：
### 1. 核心类要求：【说明包含哪些类，给出类名】，具体设计如下：
    【逐一对类的结构进行说明】
(1)【类名】
- 私有属性：【属性名】【属性数据类型】【属性功能】↵
- 公共方法：【方法名()】：【参数及类型说明】【返回值及类型说明】【功能及类型】
- 必须提供合理的构造方法，用于初始化属性↵
### 2. 技术约束要求：
- 仅允许使用【哪些技术和知识点】
- 严格体现面向对象封装思想
- 代码必须具备良好的可读性，不允许出现冗余代码
### 3. 额外要求：
- 为每个类、每个方法添加详细注释，说明其功能、参数含义(如有**)、返回值含义(如**)
- 在 main 方法中调用，用于启动程序
### 4. 输出要求：
- 代码格式整洁，缩进规范，符合 Java 编码规范
- 为每个方法添加注释，清晰说明方法的功能
```

Figure 4. Prompt template

图 4. 提示词模版

#### 4.4. 教学评价的重构

传统的编程课程，以培养学生编程能力为主要目标，评价方式始终围绕“编写的代码是否正确运行，语法是否正确”这一主题。结果性评价占比较大。

AI 编程下，更侧重于需求拆解、AI 协作和创新拓展三个核心能力的培养，这就需要弱化结果性评价，构建“过程性评价 + 结果评价”的多元评价模式。通过每次课程任务后学生提交的作业项目过程文档的内容，来评价学生能否将复杂问题转化为清晰、结构化的自然语言指令；能否有效利用 AI 工具生成代码；能否将所学方法迁移到新场景，实现功能创新，形成过程性评价结果。

### 5. 教学效果与反思

考虑到本研究的教学模式与传统编程教学相比，教学内容、教学目的、编程方法都有较大改变，因此只选择了本校计算机科学与技术专业 24 级辅修学生进行了小范围的实验教学。教学评价仍然采用结果性评价，主要考查知识点的掌握和编程能力，与同期采用传统教学形式的班级相比，实验班的平均分为 78.51，普通班为 81.46，两者差距不大。这说明，只要在知识内化环节，教师所给问题充分，就能够引导

学生通过分析代码、主动探索掌握知识点。实验班的学生学习积极性更高, 自信心更强, 表现在小组活动中, 实验班同学的参与度高于普通班, 所提交课程设计作业的质量也优于普通班。通过对实验班学生进行半结构化访谈, 学生普遍认识到 AI 编程中 AI 是伙伴而非替代者, 对提示词的编写规则有一定了解, 能够在 AI 编程工具的帮助下完成一个简单功能的程序。

在这次教学实践中, 也发现了很多问题。首先, 教学任务完成时间估计不足, AI 编程工具的编码过程非常依赖网络环境, 导致这一阶段占用时间过长, 课堂教学活动无法全部完成; 其次, 学生对 AI 工具产生了依赖, 当环境受限(如期末卷面考试)时, 表现出明显不适应, 有些同学连基本的关键字都无法写正确; 最后, 学生纠结语法细节的思想仍然存在。针对这些问题, 可以分别通过细化教学任务、执行新的评价方案等来解决。现有六个教学任务涉及的知识目标内容较多, 需进一步细化为两到三个子任务, 使单次任务牵涉的知识点更加聚焦、任务更趋微型化, 方能在规定时间内完成整个教学流程; 在考评方式仍然侧重知识点和编程能力的情况下, 学生纠结于知识点的细节在所难免, 尽快推动执行新的评价方式, 加大过程性评价所占分值, 有助于改变学生的学习心态和习惯。

## 6. 总结

2025 年, AI 编程工具功能不断完善, 迭代加速。对软件开发人员而言, 学会利用 AI 编程已不再是一项锦上添花的附加技能, 而正演变为职场刚需。氛围编程融入阶梯式任务教学模式中, 通过设计递进式任务体系、构建基于氛围编程的学习流程, 有效解决了程序设计类课程教学中对学生 AI 能力培养不足的问题。本研究教学实践中所积累的教学内容、教学任务、教学方式、考核评价等方面的革新经验, 也为同类课程的教学改革提供了有益的参考和实践依据, 为如何培养适应未来软件开发需求的高素质人才提供了思路。

## 基金项目

本研究项目由安徽省教育厅质量工程项目“Java 语言程序设计(线上线下混合课程)”(项目号: 2023xsxx315)、淮南师范学院教学研究项目“混合教学环境下基于 Web 的协作学习模型研究”(项目号: 2022hsjyxm14)资助。

## 参考文献

- [1] 《Stack Overflow 2025 开发者报告》速览[EB/OL]. 2025-08-07.  
<https://www.oschina.net/news/364831/stack-overflow-2025-developer-survey>, 2025-12-05.
- [2] Mei, L.R., Yao, J.Y., Ge, Y.Y., *et al.* (2025) A Survey of Context Engineering for Large Language Models.  
<https://arxiv.org/abs/2507.13334>
- [3] 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.
- [4] 赵斌. 重塑课堂: 从混合式教学到 AI 辅助智慧教学[J]. 中国大学教学, 2025(11): 69-75.
- [5] 苏小红, 苗启广, 陈文字. 基于 AI 赋能和产教融合提升程序设计能力的个性教学模式[J]. 中国大学教学, 2023(6): 4-9.
- [6] 翟雪松, 张丽洁, 夏亮亮, 等. 基于 GAI 的逆向工程教学思维在人机协作中的应用研究——以编程教育为例[J]. 电化教育研究, 2024, 45(9): 61-68.
- [7] 赵璐, 蔡惠, 王烟濛, 等. 生成式人工智能赋能程序设计课程教学研究[J]. 大学教育, 2025(8): 33-39.
- [8] Ge, Y.Y., Mei, L.R., Duan, Z.H., *et al.* (2025) A Survey of Vibe Coding with Large Language Models.  
<https://arxiv.org/abs/2510.12399>
- [9] 刘瑜, 韩庆龙, 吕洁. 以培养自主学习能力和目标的阶梯任务驱动教学法在程序设计课程中的应用[J]. 计算机工程与科学, 2016, 38(S1): 178-181.