Advances in Education 教育进展, 2026, 16(4), 1392-1402
Published Online April 2026 in Hans. https://www.hanspub.org/journal/ae
https://doi.org/10.12677/ae.2026.164793

Hans汉斯

智能场景驱动的智能开发课程教学创新研究

孙博文，高梦琦*，王家辉，申　奥

上海第二工业大学计算机与信息工程学院，上海

收稿日期：2026年3月22日；录用日期：2026年4月19日；发布日期：2026年4月29日

摘　要

针对高校智能开发类课程存在的语言中心化、课赛分离及工程能力培养不足等问题，本文围绕课赛融合与人机协同驱动，提出面向系统构造能力培养的课程重构方案。首先，构建“计算表达－系统构造－智能实现”三层能力递进模型，优化课程目标结构；其次，以智能应用场景为主线，融合竞赛问题范式重组教学内容，实现课堂学习与算法训练的协同推进；同时设计贯穿学期的渐进式系统开发任务链，引入大模型辅助编程机制，形成“生成－审辨－优化－验证”的人机协同闭环。通过阶段性竞赛强化与工程化评价方式，促进基础能力、算法能力与系统实现能力的协同提升。实践表明，该体系有效增强了学生的复杂问题建模与系统构建能力。

关键词

智能开发类课程，课赛融合，人机协同编程，系统构造能力培养

Research on Teaching Innovation in Intelligent Development Courses Driven by Intelligent Scenarios

Bowen Sun, Mengqi Gao*, Jiahui Wang, Ao Shen

School of Computer and Information Engineering, Shanghai Polytechnic University, Shanghai

Received: March 22, 2026; accepted: April 19, 2026; published: April 29, 2026

Abstract

To address the issues of language-centered instruction, separation between coursework and competitions, and insufficient cultivation of engineering competence in intelligent development courses

*通讯作者。

文章引用：孙博文，高梦琦，王家辉，申奥．智能场景驱动的智能开发课程教学创新研究[J]．教育进展，2026，16(4)：1392-1402. DOI: 10.12677/ae.2026.164793

at universities, this study proposes a curriculum reconstruction framework oriented toward system construction capability, driven by course-competition integration and human-AI collaboration. First, a three-level progressive capability model—"computational expression, system construction, and intelligent implementation"—is established to optimize the structure of course objectives. Second, teaching content is reorganized around intelligent application scenarios, integrating competition-style problem paradigms to align classroom learning with algorithmic training. Meanwhile, a semester-long progressive system development task chain is designed, together with a large-model-assisted programming mechanism, forming a human-AI collaborative loop of "generation-analysis-optimization-validation". Through staged competition reinforcement and an engineering-oriented evaluation approach, the reform promotes the coordinated development of foundational skills, algorithmic competence, and system implementation ability. Practice demonstrates that the proposed framework effectively enhances students' abilities in complex problem modeling and system construction.

Keywords

Intelligent Development Courses, Course-Competition Integration, Human-AI Collaborative Programming, System Construction Competence Development

1. 引言

近年来，以 OpenAI 发布的 ChatGPT 为代表的大模型技术迅速发展，人工智能辅助编程逐步进入常态化应用阶段。大模型在代码生成、错误修复与算法提示方面表现出较强能力，对程序设计类课程的教学形态产生了深刻影响[1] [2]。与此同时，国内新工科建设持续推进，强调面向复杂工程问题的综合能力培养，而不仅仅是语法层面的知识掌握[3]。

在新工科建设背景下，程序设计课程目标逐步由“语法熟练”转向“复杂问题求解能力培养”。吴岩指出，新工科建设强调工程能力与创新能力的协同培养，要求课程目标服务于高阶能力结构的形成[4]；林健进一步提出，工程教育改革应从知识体系转向能力体系重构[5]。杨宗凯等从人工智能时代人才培养角度指出，高等教育应强化计算思维与系统思维能力建设[6]。在具体课程层面，杨雨薇基于 OBE 理念构建教学目标与评价体系，实现了目标–教学–评价的一体化闭环[7]；施晓秋等在工程教育认证背景下提出以产教融合反向设计毕业要求与课程目标，有效提升了课程目标达成度[8]。上述研究为“计算表达–系统构造–智能实现”三层能力体系的提出提供了理论基础。

传统程序设计课程多采用章节式语法递进结构，存在知识碎片化与工程脱节问题。李秀等指出，在人工智能背景下，计算机基础课程应围绕应用场景进行结构重组，以增强知识整体性[9]。陈康等提出强化综合任务与模块化设计[10]。曾新等通过构建“任务递进 + 分层实验”结构，将语法知识嵌入系统任务中，显著提高学生实践能力[11]。孙世温等在课程思政融入实践中，也强调通过问题场景串联知识模块，增强知识迁移能力[12]。这些研究表明，场景驱动结构有助于打破知识割裂状态，为系统构造能力培养奠定基础。

在教学组织方式方面，工程教育认证背景下的课程改革强调“任务链–能力点–评价指标”之间的对应关系[8]。国际研究则更加重视学习机制设计。Becker 与 Quille 回顾 CS1 五十年发展历程，指出主动学习与结构化任务设计是提升初学者成功率的关键因素[13]。Luxton-Reilly 等在系统综述中指出，持续反馈与即时评估机制显著改善学生学习效果[14]。Fangohr 等通过自动评测系统实践证明，实时反馈机制能

够增强学生代码修正与优化能力[15]。这些研究为构建“渐进式系统开发 + 闭环反馈机制”提供了理论与实证支持。

当前程序设计课程改革在目标优化、项目驱动和评价改进等方面取得了一定成效，但从整体来看，仍以局部调整为主，缺少系统性的整合设计。课程目标虽然强调能力导向，但尚未构建出清晰的层级结构，计算表达、系统构造与智能实现三者之间缺乏递进式的衔接关系。在内容组织上，虽然引入了项目和任务，但大多仍停留在模块化拼接层面，缺乏一条贯穿始终的场景驱动主线。

随着以生成式工具为代表的大模型技术广泛应用，学生获取代码结果的成本大幅降低。如果教学仍停留在语法训练层面，学生的算法分析能力和系统建模能力将面临被弱化的风险。因此，如何在智能辅助环境下保持学习的挑战性，实现“工具可用而能力不减”，成为当前程序设计课程改革亟待解决的关键问题。

在现有程序设计教学研究中，PBL (Project-Based Learning)强调通过项目驱动促进知识整合，OBE (Outcome-Based Education)强调以能力达成为导向进行课程设计。然而，在大模型广泛应用的背景下，传统 PBL 与 OBE 模式在“学习过程控制”与“能力形成路径”方面仍面临新的挑战。本文在上述理论的基础上，进一步引入人机协同视角，对教学过程进行结构化重构。

针对上述问题，本文从课程目标、内容结构和教学机制三个层面展开系统重构。首先，构建“计算表达–系统构造–智能实现”三层能力体系，形成清晰的目标层级；其次，推动教学内容从语法线性结构向场景驱动结构转变，建立以系统开发为主线的学习路径；最后，设计“渐进式系统开发 + 人机协同训练”的闭环机制，确保能力培养贯穿始终。通过目标体系、内容组织和实施路径的统一，本文为大模型背景下程序设计课程的结构化升级提供了可操作的改革方案。

与已有研究相比，本文的主要创新体现在以下两个方面：

1) 提出“计算表达–系统构造–智能实现”三层递进能力结构，将程序设计能力从代码层拓展至系统层与智能层，构建清晰的能力演进路径；

2) 构建“生成–审辨–优化–验证”的人机协同闭环机制，在大模型辅助环境下，将“代码生成”转化为“代码理解与重构”，实现对学生高阶能力的有效约束与引导。

2. 整体框架

本研究围绕课程体系结构性重构展开，从课程目标、内容组织与教学实施三个层面进行系统设计，形成“能力–结构–机制”协同推进的改革框架。整体研究框架如图 1 所示。

首先，在课程目标层面，围绕专业能力需求重构培养定位，构建“计算表达–系统构造–智能实现”三层递进能力体系。以 Python 程序表达为基础，强化变量、函数与数据结构的规范使用能力；在此之上，通过 Django 框架下的模块划分与接口设计训练系统构造能力；进一步结合算法实现与性能优化要求，引导学生形成复杂度分析与智能功能实现能力，推动能力结构由代码层向系统层、再向算法层逐步提升。

其次，在内容结构层面，改变传统按照语法知识线性展开的教学方式，转向以完整系统场景为主线的组织模式。课程围绕“智能信息管理与决策支持系统”这一工程载体，将数据建模、文件操作、数据库访问、数据分析与可视化等知识嵌入不同阶段的功能实现任务中，实现知识点在具体问题情境中的自然引出与综合应用，使内容结构由分散式讲授转变为场景驱动式构建。

最后，在教学模式层面，构建“渐进式系统开发 + 人机协同训练”的实施机制。通过贯穿学期的分阶段系统开发任务链，逐步推进数据设计、功能实现与性能优化训练，并在关键环节引入“生成–审辨–优化–验证”的人机协同闭环，要求学生对生成代码进行结构分析与复杂度评估，在规范使用辅助工具的同时强化自主判断能力，从而保障能力培养目标在实践层面的有效落实。

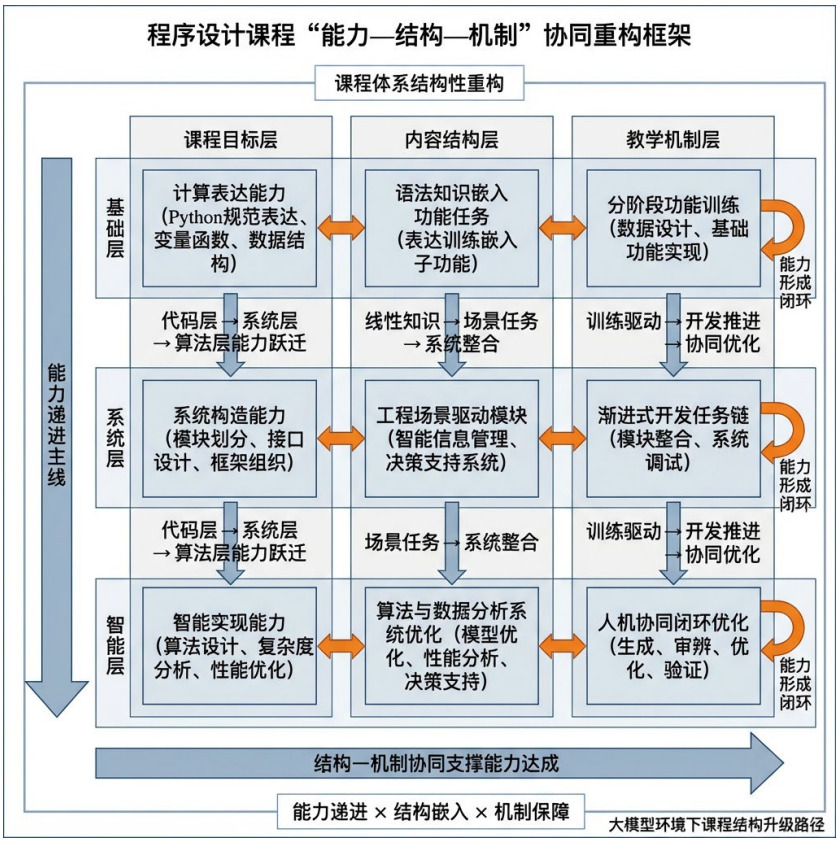

Figure 1. Framework diagram of teaching innovation research
图 1. 教学创新研究框架图

3. 教学改革创新设计方案

智能开发类课程普遍面向智能类专业大一和大二学生，既承担程序设计能力启蒙任务，又肩负支撑后续机器学习、模式识别与智能系统课程的基础功能。针对传统程序设计课程存在的“语言中心化、内容碎片化、课赛分离化”问题，本研究从能力目标、内容结构与教学模式三个层面进行系统重构，构建面向智能系统构造能力培养的课赛融合与人机协同教学体系。上述本教学改革的设计逻辑如图 2 所示。

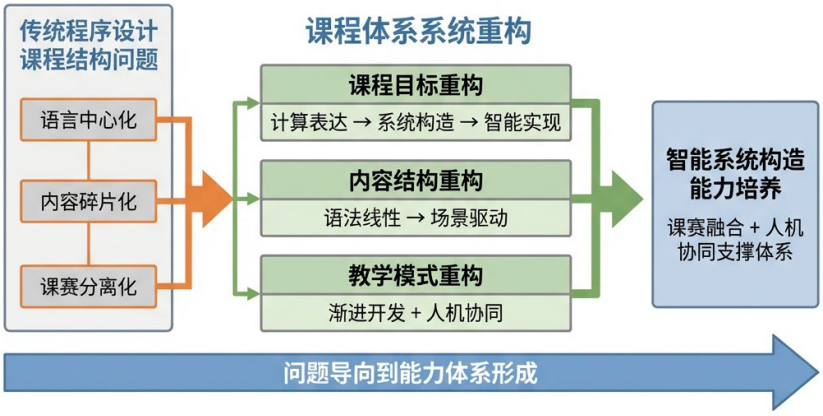

Figure 2. Design concept of the teaching reform and innovation plan
图 2. 教学改革创新方案设计思路

3.1. 课程目标重构：构建“计算表达–系统构造–智能实现”三层能力培养体系

传统开发类课程通常以语法掌握为核心，注重变量、控制结构、函数调用等知识点的线性讲解，评价方式也主要看程序运行结果是否正确。在智能类专业背景下，这种教学模式逐渐暴露出一些局限：一是能力培养比较单一，学生难以支撑起复杂系统的开发；二是缺乏从智能问题到代码实现的完整训练；三是和后续专业课程的衔接不够紧密，学生学完后不容易直接用到后面的学习中。

针对上述问题，本研究提出“计算表达–系统构造–智能实现”三层递进能力模型，每一层在教学内容与能力要求上逐级深化，具体阐述如下：

第一层为计算表达能力，旨在帮助学生完成从“能写代码”到“会想问题”的转变。传统教学往往满足于语法正确、程序能跑通，而本层强调将现实问题形式化为数据结构与算法流程的全过程。具体而言，从“语法正确”转向“表达清晰”体现在以下几个方面：一是问题抽象，引导学生从模糊的现实需求中提取关键要素，明确“输入–处理–输出”边界；二是数据建模，训练学生根据问题特征选择或设计合适的数据结构(如数组、链表、集合、映射等)来组织数据；三是算法描述，要求学生能够用自然语言、流程图或伪代码清晰表达算法思路，而非直接上手写代码。通过以上训练，使学生编写的程序不仅正确，而且逻辑清晰、结构合理、易于理解。

第二层为系统构造能力，引导学生从编写单一程序逐步过渡到模块化系统设计。随着问题规模扩大，单一文件、线性结构的程序难以满足复杂需求，本层重点训练学生拆解与整合的能力。具体内容包括：一是模块划分，学习如何根据功能职责将系统拆分为高内聚、低耦合的模块；二是接口设计，明确模块之间的交互方式，定义清晰的输入输出规范；三是数据流与控制流组织，理解数据在模块间的传递路径以及系统整体的执行时序；四是异常处理与工程规范，培养健壮性编码习惯，包括错误捕获、日志记录、代码风格统一等。通过多文件组织实践和简单架构设计(如分层架构、MVC 模式初探)，使学生切身体会“程序–模块–系统”的层级递进关系，初步具备构建可维护、可扩展的复杂系统的能力。

第三层为智能实现能力，在掌握系统构造的基础上，进一步向智能化应用延伸。本层旨在打通从数据处理到智能决策的实现路径，为后续机器学习、数据科学等专业课程搭建衔接桥梁。具体内容包括：一是规则建模，引导学生将领域知识或经验转化为可执行的规则逻辑(如 if-then 规则、决策表等)，实现简单的推荐或预警功能；二是策略算法，引入基础的搜索、排序或贪心策略，使系统能够在多选项中做出选择；三是初步数据分析，训练学生对数据进行统计汇总、趋势分析或简单分类(如基于阈值或距离的分类器实现)，让程序具备“从数据中得出结论”的能力。通过上述实践，学生能够实现具有类智能特征的功能模块，如规则推荐系统、数据统计决策模块或简单的分类逻辑，为后续深入学习智能化系统开发奠定扎实基础。

课程安排中，每一阶段均对应能力升级节点：前期侧重问题抽象与算法表达，中期强化模块化系统构建，后期融合规则与数据驱动方法，完成智能功能原型开发。能力训练由“表达”向“构造”再向“实现”逐级跃迁，形成结构化递进体系。该模型突破了传统程序设计课程的语言中心范式，为智能系统开发奠定能力结构基础。上述本节内容结构图如图 3 所示。课程内容由问题抽象与算法表达起步，逐步过渡至模块化系统构建，最终延伸至规则建模与数据驱动的智能功能实现，形成由表达向构造再向实现的结构化能力跃迁路径。

3.2. 内容结构重构：由“语法线性结构”向“场景驱动结构”转型

传统课程按语法顺序组织教学虽有利于知识系统化，但缺乏问题情境支撑，学生难以理解知识的工程价值。同时，竞赛训练往往脱离课堂，形成“课内基础–课外竞赛”的割裂状态，导致知识学习与高强

度问题求解能力培养分离。本研究以“双驱动”机制重构内容结构：

以智能开发基础课程为例。该课程主要讲述 Python 的语法知识，在该课程中，围绕“智能信息管理与分析系统”构建内容主线，将所有知识点嵌入系统构建过程。通过应用场景驱动，使知识点服务于系统目标，学生在学习过程中对相关语法用处有更具象化认识。具体 Python 知识点与智能信息管理系统功能映射表如表 1 所示。

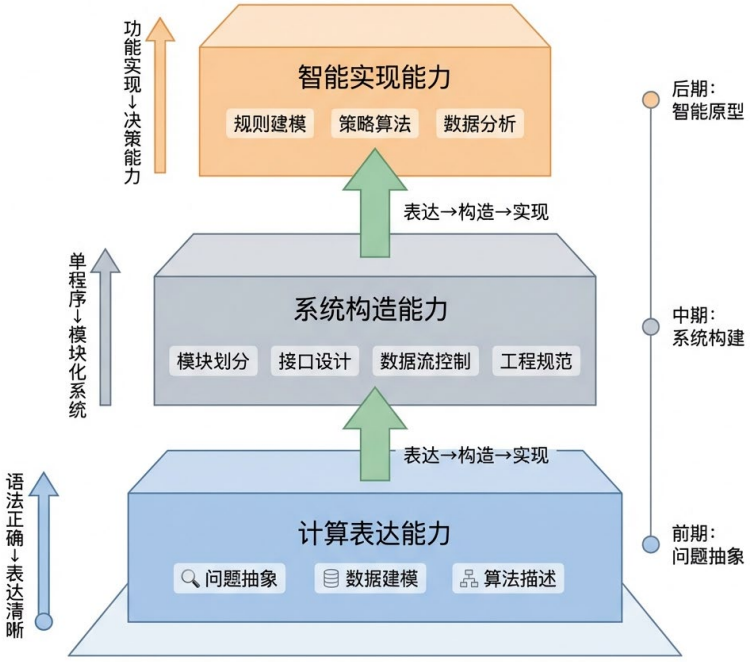

Figure 3. Three-level progressive content structure model of “computational expression-system construction-intelligent implementation”
图 3. “计算表达 - 系统构造 - 智能实现”三层递进内容结构模型

Table 1. Mapping table between Python syntax knowledge points and functions of the intelligent information management system
表 1. Python 语法知识点与智能信息管理系统功能映射表

Python 知识模块	对应具体功能	教学实现形式示例
数据类型(list, dict, tuple)	信息结构建模、数据组织、记录存储	构建学生信息表、成绩记录表、嵌套数据结构
条件语句(if-else)	数据筛选、权限控制、规则判断、异常处理	实现规则推荐逻辑、合法性校验模块
循环结构(for/while)	批量数据处理、统计计算、自动化遍历	实现成绩统计、数据批处理、频次分析
函数(function)	模块封装、接口设计、代码复用	将系统拆分为数据加载、分析计算、结果输出模块
文件操作(读写)	数据持久化、本地数据管理、日志记录	读取 CSV 数据、保存分析结果、实现数据接口
类与对象(OOP)	实体建模、组件封装	构建 DataManager 类、User 类等
简单机器学习概念(特征、模型、预测)	数据驱动决策、分析、分类等	实现线性回归预测、KNN 分类、规则推荐模块
异常处理(try-except)	错误控制、系统鲁棒性提升	数据格式错误处理、运行异常捕获
模块化结构(import)	功能解耦、系统扩展性	多文件结构设计、模块化系统开发

同样以智能开发基础课程为例，通过学科竞赛引领与创新思维融入的双重策略，本课程引导学生在期末大作业中将所学数据算法理论与数据开发技术有机结合，创新性地搭建智能数据应用系统。该过程不仅培养了学生的创新实践思维，也锻炼了其解决复杂工程问题的能力，帮助学生实现从“课程作业”到“参赛作品”、再到“毕业设计”和“求职就业”的平滑过渡，形成了“以赛促学、以赛促教、以赛促创”的良性循环。在此过程中，学生在各类学科竞赛中取得优异成绩，同时夯实了未来职业发展所需的各项基础技能。对应的学生竞赛获奖情况如表 2 所示。

Table 2. Table of the fundamentals of intelligent development course with competition award achievements
表 2. 智能开发基础对标竞赛获奖情况表

编号	项目名称	竞赛奖项
1	融合 AIGC 的病程记录与健康指导平台	全国三等奖
2	AIGC-基于 Python/Django 的动态美学创生系统	全国三等奖
3	面向机器学习课程的辅助教学平台	上海市二等奖
4	基于深度学习的葡萄酒分析系统	上海市三等奖
5	基于 FLASK 的 AI 生活管家平台	上海市三等奖

本课程构建“知识－场景－竞赛范式”三维融合机制，将 Python 基础知识嵌入智能数据系统等典型智能应用场景之中，课堂教学不再局限于语法讲授，而是在系统场景与竞赛结构的双重驱动下实现知识重组与能力跃迁，促进系统构造能力与算法优化能力的协同提升。

3.3. 教学模式重构：构建“渐进式系统开发 + 人机协同训练”机制

在教学实施层面，以智能开发基础课程为例，该课程以 Python 3.8 为核心开发语言，围绕“智能信息管理与决策支持系统”构建贯穿学期的渐进式系统开发任务链。课程要求的系统采用 Django 框架实现前后端数据交互，利用其 MVC (Model-View-Template)架构完成业务逻辑分层设计。通过 Django ORM 实现数据持久化管理，学生可在视图函数(views.py)中调用数据库接口完成数据读取与处理，从而理解 Web 系统中“模型－视图－控制逻辑”的分离机制。在数据分析模块中，引入 NumPy 进行矩阵运算，实现向量化计算。在智能分析环节，学生可以利用 Matplotlib 完成数据可视化。该课程通过真实代码实践，使数据处理、统计分析与可视化功能嵌入完整系统流程，而非独立实验操作。

课程中的系统开发按照“数据输入与结构设计－文件存储与统计分析－模块整合与接口优化－智能决策功能扩展”的顺序分阶段推进。前期要求学生完成基础数据模型设计与 Django 路由配置；中期实现数据读写与分析计算；后期整合 NumPy 运算结果与可视化输出；最终扩展简单决策模块，例如基于规则或简单分类算法实现推荐功能。教师在每一阶段均进行代码结构检查与接口测试，教师会重点审查模块划分是否清晰、函数是否具备单一职责原则，以避免代码耦合过度。

在此基础上，本研究引入人机协同编程机制，规范使用大模型辅助工具，但明确学生必须完成关键逻辑设计与算法分析。本课程要求建立“生成－审辨－优化－验证”的闭环流程：学生可使用大模型生成初始代码框架，例如 Django 视图函数或算法模板，但需自行完成数据结构设计、复杂度分析与代码重构，并通过教师指定的单元测试或边界测试。通过这种方式，实现学生由“代码生成依赖”向“算法判断主导”的能力转型。为保障教学质量，该课程也建立工程化多维评价体系。评价指标包括：模块结构是否符合 Django 分层设计、代码是否遵循函数封装规范、NumPy 运算是否实现向量化、系统接口是否清晰、测试用例是否覆盖边界情况，以及系统是否具备扩展能力。评价方式结合代码审查(code review)、阶段性功能验收与性能测试结果，避免单纯以最终运行结果作为唯一标准，实现对系统结构与开发过程的

综合评价。

具体案例比如，在“数据分类功能”实现中，引入 AI 辅助编程任务。首先由学生调用大模型生成 KNN 分类算法基础代码；随后要求学生完成以下审辨任务：

1) 分析代码中数据结构组织方式；

2) 计算时间复杂度并指出性能瓶颈；

3) 提出至少一种优化方案(如使用向量化或数据索引结构)；

4) 通过自定义测试数据验证优化效果。

该任务将“代码生成”转化为“结构分析与性能优化”，有效避免学生直接依赖生成结果。评分规则见附录(附表 1)。

3.4. 本节小结

本研究以能力结构重构为核心，以智能场景与竞赛范式为内容组织逻辑，以渐进式系统开发与人机协同编程为实施路径，构建面向智能类专业的新型智能开发课程体系。该方案不仅提升学生基础编程能力，更强化算法建模能力、系统构造能力与创新实践能力，为后续专业课程学习与高水平竞赛训练奠定坚实基础。

4. 应用效果

经过 2025 年的教学改革实践，我校智能科学与技术教研室全体教师通力合作，共同探索和优化了智能开发类课程教学与竞赛元素融合的创新策略，通过课程成绩、系统项目完成度、竞赛成果及学生反馈等多维数据进行综合评估，推广应用效果显著。

4.1. 学习成绩与系统完成质量提升

为直观反映教学改革效果，(以智能开发基础科学为例)对改革前后课程成绩结构进行对比分析。如图 4 所示，改革后课程成绩分布结构发生明显变化，整体达标水平显著提升。具体而言，改革前不及格人数为 3 人，占比 7.7%；改革后不及格人数降为 0 人，实现全员达标。改革后中等及以上人数比例由改革前的 82.0%提升至 88.6%，说明整体学习水平呈上升趋势。虽然优秀人数比例由 15.4%略降至 9.1%，但良好与中等人数显著增加，成绩结构更加稳定，整体呈现向中高分段集中趋势。

此外，系统综合项目完成质量明显改善，学生在基于 Django 的系统开发、NumPy 数据分析与 Matplotlib 可视化集成方面的实现完整度显著提高，体现出渐进式系统开发任务链与竞赛问题范式嵌入机制的综合作用。总体来看，成绩结构优化与系统实现能力提升相互印证，说明本研究构建的“能力重构–场景驱动–竞赛强化–人机协同”教学体系具有良好的实施效果。

4.2. 竞赛成果转化效果

融入竞赛问题范式后，学生的算法建模能力与项目转化能力明显增强。2025 年度，共有 15 组课程项目完成系统优化，其中 12 组主动转化为竞赛作品，占比约 80%。

在参与的省级及以上竞赛中，共获得：省级奖项 10 项，其中进入国家级赛事 6 项，获得国家级三等奖 3 项。部分优秀竞赛作品在赛后再次回归课堂，作为案例用于后续教学展示，实现课程内容与竞赛成果的双向反哺。与往年相比，课程成果转化率提升明显，体现出课赛融合机制的有效性。

4.3. 人机协同训练效果

本研究对班级 39 名学生开展问卷调查，共回收有效问卷 34 份。问卷围绕代码理解、结构分析与重构能

力三个维度设计。结果显示，83%的学生认为“生成–审辨–优化–验证”闭环机制有助于理解代码逻辑，76%的学生表示会主动分析生成代码的结构合理性，73%的学生能够在生成代码的基础上进行结构重构。

例如在 KNN 分类任务中，对学生最终提交代码与模型生成初始代码进行结构对比分析，结果显示约82%的学生对原始代码进行变量重命名、函数封装或数据结构优化，仅 12%的学生基本保持原始结构不变。同时，在独立完成的复杂度分析测试中，90%的学生能够正确给出时间复杂度表达式。

上述数据表明，人机协同机制并未削弱学生的算法分析能力，反而强化了代码审辨与结构优化能力。

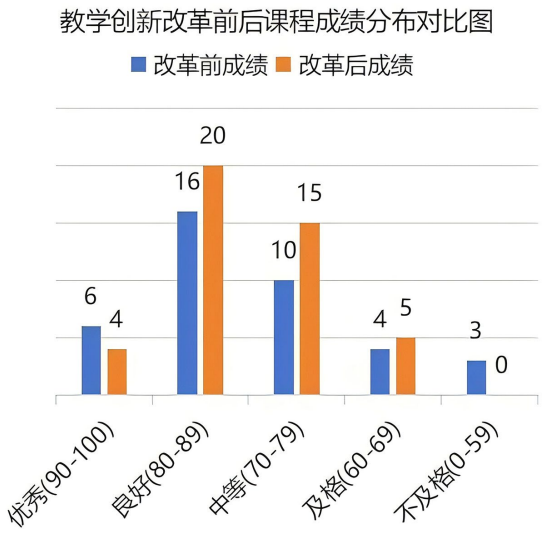

Figure 4. Comparison map of course score distributions before and after the reform
图 4. 改革前后课程成绩分布对比图

4.4. 学习兴趣与参与度变化

通过课程满意度调查(采用 5 分制量表)可以看到，教学改革实施后学生对课程学习的整体体验明显改善，尤其是在系统开发类任务方面的学习兴趣显著增强。统计结果显示，改革后班级在课程项目完成体验方面的平均满意度达到 4.56 分，明显高于上一届改革实施前的 3.90 分。与此同时，课程项目的整体完成率达到 100%，所有学生均能够按照课程进度完成系统开发任务并提交完整项目成果，说明以系统构建为主线的任务链设计在一定程度上降低了学生对复杂项目的畏难情绪，提高了学习投入度。

在具体教学环节中，阶段性竞赛训练周对课堂氛围的提升尤为明显。与以往以知识讲解为主的课堂相比，学生在竞赛问题训练环节表现出更高的参与积极性，能够围绕算法实现方案、数据结构选择以及时间复杂度优化等问题展开讨论。例如，在数据统计与搜索类任务训练中，部分学生会主动比较不同实现方式的效率差异，并尝试通过字典结构或集合结构对原有循环算法进行优化。课堂上逐渐形成了“问题讨论–代码验证–结果比较”的学习互动模式，学生不仅关注程序能否运行，更开始关注算法结构与性能表现。

整体来看，课程改革通过引入系统开发任务链与竞赛训练机制，使学生从被动完成实验任务转变为主动探索解决方案，课堂互动性与学习氛围均得到明显改善。这一变化从课程满意度评价与课堂行为表现两个方面均得到体现。

5. 总结

综合成绩结构、代码实现统计与竞赛成果转化情况可以看出，本研究的教学改革在系统开发能力培

养、算法优化意识形成以及人机协同环境下的代码判断能力提升方面取得了较为稳定的成效。课程中高分段比例提高，课程项目向竞赛作品的转化率明显增强，部分成果实现了反向融入课堂。研究实践表明，课程设计中将系统开发主线、竞赛问题范式与规范化人机协同机制相结合，有助于增强学生的工程实践能力与结构化思维，为智能类专业课程的持续改进提供了可行路径。

本研究在教学实践中取得了一定成效，但仍存在若干局限性与挑战。首先，从实施条件来看，该模式对教师提出了较高要求。教师不仅需要具备扎实的程序设计与系统开发能力，还需具备任务链设计能力及对大模型生成代码的审辨能力，这在一定程度上提高了课程实施门槛。为此，后续可通过构建教学资源库、案例库及标准化任务模板，降低教师个体经验依赖。

其次，在引入大模型辅助编程的过程中，部分学生可能产生对生成结果的依赖，弱化自主分析能力。尽管本文通过“生成–审辨–优化–验证”闭环进行约束，但在实际教学中仍存在个体差异。未来可通过强化过程性评价(如代码解释、复杂度分析报告)与限制直接生成比例等方式，进一步提升学生的独立思考能力。

再次，课程以“智能信息管理系统”为主线开展任务设计，虽然有利于形成完整的工程认知，但单一项目载体在一定程度上可能限制知识覆盖面与应用多样性。后续研究可引入多场景项目(如推荐系统、小型视觉任务等)，构建多元化任务体系，以提升知识迁移能力。

最后，当前教学效果评估主要基于课程成绩、问卷与课堂表现，缺乏长期跟踪数据。未来可通过跨课程对比与纵向跟踪分析，进一步验证该模式对后续专业课程学习效果的持续影响。

基金项目

2025 年上海高校青年教师培养资助计划项目(A01GY25F012)。

参考文献

[1] Achiam, J., *et al*. (2023) GPT-4 Technical Report. arXiv: 2303.08774.

[2] Sébastien, B., Varun, C., Ronen, E., *et al*. (2023) Sparks of Artificial General Intelligence: Early Experiments with GPT-4. arXiv: 2303.12712.

[3] 钟登华. 新工科建设的内涵与行动[J]. 高等工程教育研究, 2017(3): 1-6.

[4] 吴岩. 新工科: 高等工程教育的未来——对高等教育未来的战略思考[J]. 高等工程教育研究, 2018(6): 1-3.

[5] 林健. 面向未来的中国新工科建设[J]. 清华大学教育研究, 2017, 38(2): 26-35.

[6] 杨宗凯, 王俊, 吴砥, 等. ChatGPT/生成式人工智能对教育的影响探析及应对策略[J]. 华东师范大学学报 (教育科学版), 2023, 41(7): 26-35.

[7] 雨薇杨. 基于 OBE 的高校 Python 程序设计教学目标与评价体系改革[J]. 计算机科学技术与应用, 2025, 2(8): 12-14.

[8] 施晓秋, 徐赢颖. 工程教育认证与产教融合共同驱动的人才培养体系建设[J]. 高等工程教育研究, 2019(2): 33-39, 56.

[9] 李秀, 陆军, 牛颂杰, 等. 人工智能时代计算机基础课程建设与教育教学思考[J]. 清华大学教育研究, 2024, 45(2): 42-49, 70.

[10] 陈康, 王丹丹. 面向应用开发的计算机实践教学模式改革[J]. 科教导刊(下旬), 2016(21): 79-80.

[11] 曾新, 王梅良, 李高权, 等. Python 程序设计语言实验教学模式探讨[J]. 实验科学与技术, 2024, 22(2): 54-58.

[12] 孙世温, 魏一静, 王志欣. 程序设计类课程思政教学的探索与实践——以 C 语言课程为例[J]. 创新教育研究, 2025, 13(6): 178-184.

[13] Becker, B.A. and Quille, K. (2019). 50 Years of CS1 at SIGCSE: A Review of the Evolution of Introductory Programming Education Research. *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, Minneapolis, 27 February-2 March 2019, 338-344. https://doi.org/10.1145/3287324.3287432

[14] Luxton-Reilly, A., Simon, Albluwi, I., Becker, B.A., Giannakos, M., Kumar, A.N., *et al*. (2018) Introductory Programming:

A Systematic Literature Review. *Proceedings Companion of the* 23*rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, Larnaca, 2-4 July 2018, 55-106. https://doi.org/10.1145/3293881.3295779

[15] Fangohr, H., O'Brien, N., Prabhakar, A., *et al*. (2015) Teaching Python Programming with Automatic Assessment and Feedback Provision. arXiv: 1509.03556.

附 录

Table S1. Evaluation rubric for AI code critique ability (Example)
附表 1. AI 代码审辨能力评价量规(示例)

评价维度	指标说明	分值
逻辑理解	能准确解释代码功能与流程	0~25
结构分析	能识别数据结构与模块关系	0~25
复杂度分析	能分析时间/空间复杂度	0~20
优化能力	能提出合理优化方案并实现	0~20
验证能力	能设计测试用例验证结果	0~10

注：总分 100 分，用于过程性评价与阶段考核。