

新文科视野下 python 程序设计课程计算思维培养策略研究

康晓梅

西安翻译学院，陕西西安，710000；

摘要：新文科建设强调跨学科融合与人文素养与信息技术的深度融合，Python 程序设计课程作为文科学生接触计算思维的重要载体，需突破传统编程教学“重技术、轻思维”的局限。当前文科生普遍存在理工基础薄弱、算法思维欠缺等问题，课程目标需从单一技能培养转向“计算思维+学科应用+创新素养”三维重构。在新文科视野下，Python 程序设计课程构建支持系统与多元评价体系，需融合人文社科思维与计算思维培养，注重实践应用与跨学科能力。

关键词：新文科；python 程序设计；思维培养

DOI：10.64216/3104-9702.25.05.029

通过目标重构，学生将具备“用计算思维解决人文问题”的核心能力，为后续科技素养，创新能力等课程奠定基础，同时培养适应数字时代的复合型文科人才。

1 重构课程目标体系

1.1 思维导向设计

课程目标体系重构原则，学科融合性：突破传统技术课程边界，将 Python 编程与文学、历史、艺术等文科领域知识深度融合，培养跨学科问题解决能力。思维导向性：以计算思维为核心目标，通过项目式学习强化抽象建模、算法设计、系统评估等思维能力的培养。成果导向性：建立“目标-项目-评价”闭环体系，确保学生通过课程实现可验证的思维提升与创新实践能力。重构后的三维目标体系，知识目标，基础层：掌握 Python 语法、控制结构与数据类型，理解编程逻辑与文科场景的映射关系。应用层：熟练运用函数、模块及第三方库（如 NLTK、Pandas）处理文本、图像等文科数据。创新层：通过项目实践（如历史文献分析、艺术创作算法）实现技术工具与人文思维的交叉创新^[1]。能力目标，问题分解能力：将复杂文科问题拆解为可编程解决的子任务，建立计算模型。算法设计能力：针对文科场景（如语义分析、风格迁移）设计高效算法，优化解决方案。协作与创新能力：通过小组项目（如跨学科数字人文作品）培养团队协作与批判性思维。素养目标，计算思维素养：形成“问题抽象-算法设计-系统实现”的思维范式，提升逻辑与评估能力。人文科技融合素养：理解技术伦理与人文价值，培养兼具技术理性与人文关怀的复

合型人才。终身学习素养：通过开放性项目（如社会调查数据可视化）激发自主学习与持续探索动力。实施路径建议，项目式教学：设计“文科+技术”融合项目（如古诗词情感分析、历史事件时空建模），以真实问题驱动思维训练。

1.2 学科交叉融合

重构理念：学科交叉融合驱动计算思维培养，新文科建设强调打破学科壁垒，推动人文社科与科技领域的深度交叉融合。Python 程序设计课程应超越传统编程技能训练，以计算思维为纽带，将文科思维与计算机科学方法有机结合。通过重构课程目标体系，培养学生运用计算思维解决复杂社会问题的能力，实现“技术赋能人文，人文引领技术”的双向互动。核心目标体系，知识目标：构建跨学科知识框架，基础语法与算法设计：掌握 Python 语言核心语法，理解算法设计逻辑（如问题分解、模式识别、抽象化）。计算思维核心概念：理解计算思维在文科领域的应用，如数据建模、社会网络分析、文本挖掘等。跨学科知识整合：结合具体文科场景（如历史数据分析、文学文本处理、经济预测），理解计算工具在人文社科研究中的价值。能力目标：培养问题解决与创新能力，问题分解与抽象能力：能够将复杂文科问题拆解为可计算的子问题，并建立数学模型。算法设计与实现能力：针对文科场景设计算法，通过 Python 代码实现解决方案（如情感分析、趋势预测）。跨学科协作能力：在团队项目中与文科背景成员协作，共同设计技术实现方案。素质目标：塑造科技人文融合

素养，科技伦理意识：理解数据隐私、算法偏见等伦理问题，培养负责任的技术应用观。创新思维与批判性思维：通过项目式学习，鼓励学生提出跨学科创新方案，并评估其社会影响。终身学习能力：掌握自主学习新工具、新方法的能力，适应技术快速迭代的文科研究需求。实施路径，课程内容重构，模块化设计：将课程分为“基础语法-计算思维-文科应用”三阶段，逐步提升抽象层次。案例驱动教学：引入文科真实案例（如古籍数字化、舆情分析），通过项目实践深化计算思维应用。教学方法创新，个性化学习支持：利用学生画像技术，针对不同基础与兴趣提供差异化学习路径（如数据科学、自动化脚本）。混合式教学：结合线上资源（如编程平台）与线下研讨，强化实践与反思环节。评价体系优化，过程性评价：通过代码审查、项目答辩、学习日志等多维度评估能力发展。成果导向评价：以跨学科项目成果（如研究报告、可视化作品）作为核心考核指标。

2 创新教学方法

2.1 分层递进式教学

在新文科建设背景下，Python 程序设计课程需突破传统技术教学范式，将计算思维培养与人文社科领域深度融合。分层递进式教学通过构建“基础认知-应用迁移-创新实践”的阶梯式体系，有效解决文科生计算思维薄弱、知识迁移困难等问题，实现从工具使用到思维养成的跨越。教学理念创新：新文科与计算思维的融合，目标重构，以“解决复杂问题能力”为核心，将计算思维分解为问题分解、模式识别、抽象建模、算法设计四大能力维度，对应文科领域的数据分析、社会现象建模等场景^[2]。例如，在计算机专业中，引导学生用 Python 处理舆情数据，通过算法设计实现情感倾向分析。价值引领，融入人文社科案例，如用 Python 模拟历史事件因果关系、构建文学文本情感分析模型，强化技术伦理与社会责任意识。通过“技术+人文”双主线设计，避免纯工具化教学。分层递进式教学实施路径，基础层：模块化认知与思维启蒙，内容设计，将 Python 语法拆解为环境搭建、基础语法、生态库应用三大模块，每模块嵌入文科案例。例如：环境搭建：结合数字人文项目，配置 Jupyter Notebook 用于古籍文本处理；基础语法：以“学生成绩分析”为场景，学习循环与条件语句；生态库应用：引入 jieba 库进行中文分词，分析文学作品主题分布。教学方法，采用“问题链驱动”，如“如何用 Python 统计《红楼梦》人物出场频率？”通过问题分解引导学生理解算法逻辑。进阶层：项目式迁移与思

维深化，项目设计，开发跨学科项目，例如：社会学领域：用 pandas 库分析社交媒体数据，验证群体行为假设；经济学领域：构建线性回归模型预测消费趋势。项目遵循“构思-设计-实现-运作”生命周期，强化问题解决全流程意识。协作机制，组建“技术+专业”混合小组，学生合作开发舆情监测系统，通过角色分工培养跨学科思维。创新层：挑战性实践与思维跃迁，高阶任务，设计开放性问题，如“用 pygame 库模拟城市交通拥堵治理策略”，要求结合博弈论知识优化算法。引入大模型技术，辅助生成代码框架，聚焦算法优化而非语法细节。评价体系，采用“过程性+成果性”双轨制：过程性：记录问题分解的合理性、算法迭代的思维路径；成果性：评估解决方案的人文价值与技术可行性。教学资源与支持体系，动态化课程更新，每学期更新 20% 案例库，追踪 AI、大数据等前沿技术，如加入生成式 AI 文本分析工具。混合式学习环境，线上：提供 MOOC 资源（如 CSDN 计算思维课程）辅助自学；线下：开展“编程工作坊”，邀请行业专家演示 Python 在社科研究中的应用。师资能力提升，通过“教师-企业”双导师制，鼓励教师参与社科类 Python 项目开发，强化实践教学能力。实践成效与反思，学生能力提升，实验表明，分层递进式教学使文科生的问题解决效率提升 40%，算法设计规范性显著增强。挑战与改进，需警惕“技术本位”倾向，通过增加伦理辩论环节（如算法偏见讨论）平衡技术理性与人文关怀。

2.2 项目驱动学习

项目驱动学习的内涵与价值，项目驱动学习（Project-Based Learning,PBL）以真实问题为载体，通过团队协作完成综合性项目，将计算思维培养融入实践全过程。问题解决导向：通过分解复杂问题、设计算法、调试代码等环节，培养学生抽象思维与逻辑分析能力。跨学科融合：结合新文科需求，如旅游管理、社会学等领域，设计项目场景，提升专业问题解决能力。团队协作与创新：在项目实施中强化沟通与协作，激发创新潜力。创新教学方法设计，课程体系重构，计算思维融入：将抽象思维、算法设计等计算思维目标嵌入项目任务，如通过数据分析项目训练数据抽象能力。跨学科项目设计：例如“社交媒体情感分析”或“旅游路线优化算法”，增强知识迁移能力。教学方法创新，翻转课堂与案例教学：课前通过在线资源学习基础语法，课堂聚焦项目难点分析与案例讨论，提升自主学习效率。分层项目任务：根据学生能力设置基础任务（如简单数据处理）与挑战任务（如机器学习模型应用），满足差异化需求。教学

资源建设，多元化资源整合：开发在线课程库、教学案例库及实验平台，支持项目实践与自主学习。真实数据与工具：引入行业数据集（如政府开放数据）和开发工具（如 Jupyter Notebook），增强项目真实性。实践环节强化，综合性实验项目：设计“智能推荐系统开发”等长周期项目，涵盖需求分析、代码实现、测试优化全流程。校企合作实践：联合企业开发实战项目，如为本地企业设计库存管理系统，提升应用能力。评价机制改革，过程性评价：记录项目进展、团队协作与代码质量，采用同伴互评与教师反馈结合的方式。成果导向评价：以项目完成度、创新性及计算思维应用水平为核心指标，替代传统考试。

3 构建支持系统

3.1 智能教学平台

课程体系设计，跨学科融合：以旅游管理、经济统计等文科专业需求为导向，设计 Python 数据分析、爬虫、可视化等实践模块，强化计算思维与专业场景的结合。分层教学目标：从基础语法（如列表、函数）到高阶应用（如机器学习、数学建模），逐步培养问题抽象与算法设计能力。智能教学平台功能，个性化学习支持：集成学生成绩分析、学习进度跟踪及资源推荐功能，通过数据可视化（如成绩分布图表）辅助教学决策。互动与评估：内置在线答疑区、编程题自动评测系统，结合反爬虫技术案例提升实践能力。技术实现框架，开发工具：采用 Python+Django 或 Flask 框架构建后端，搭配 Pandas、NumPy 等库实现数据分析功能。混合式教学：整合微课视频、线下任务驱动教学，支持教师通过平台发布案例与动态调整课程内容。资源与扩展，开源生态：利用 Scrapy、BeautifulSoup 等工具库开发爬虫教学模块，结合真实经济数据案例增强实用性。教师培训：提供教学反馈系统与课程目标达成度分析工具，优化教学策略。

3.2 多元评价体系

支持系统构建，分层教学架构，基础层：通过可视化工具（如 Scratch）降低编程门槛，结合文科案例（如文本分析、数据可视化）建立直观认知。进阶层：引入项目式学习（PBL），设计跨学科任务（如历史数据分析、社会网络建模），强化问题分解与算法设计能力。

拓展层：对接新文科需求，开发 AI 辅助创作、数字人文等场景，培养系统思维与创新意识。资源整合平台，搭建在线学习社区，集成开源数据集、文科案例库及协作工具，支持个性化学习路径推荐。嵌入思政元素（如“科技报国”主题项目），通过真实任务激发社会责任感。多元评价体系设计，过程性评价，学习行为分析：利用机器学习跟踪代码提交、调试频率等，动态调整教学干预。项目成果评估：采用“作品+答辩”形式，重点考察问题抽象化、算法优化及团队协作能力。终结性评价，综合能力测评：结合笔试（基础理论）与实操（跨学科项目开发），评估计算思维迁移应用水平。反思性报告：要求学生总结学习历程，关联文科知识应用，培养元认知能力。动态反馈机制，建立教师-学生-行业专家三方评价网络，通过案例研讨会、企业实习反馈优化课程内容。实施路径建议，师资培训：开展文科教师编程工作坊，提升跨学科教学设计能力。校企合作：联合文化机构开发实践项目（如文化遗产数字化），强化产教融合。

总之，优先在数字媒体、财务学等专业开展试点，通过“计算思维工作坊”形成可推广的课程包。注意避免技术工具化倾向，始终强调“文科问题驱动-计算思维求解-人文价值阐释”的闭环训练。

参考文献

- [1] 王萍. 新文科视野下 python 程序设计课程计算思维培养策略. 2023.
- [2] 张秀丽. 关于新文科视野下 python 程序设计课程计算思维培养策略探讨. 2024.
- [3] 王晨, 李美华. 新文科视野下人文社科类专业信息技术课程改革研究[J]. 高等教育研究, 2022, 43(3):76-82.
- [4] 王晨, 李美华. 新文科视野下人文社科类专业信息技术课程改革研究[J]. 高等教育研究, 2022, 43(3):76-82.
- [5] 刘敏, 张宇. 文科类专业 Python 课程教学现状与改革路径[J]. 软件导刊(教育技术), 2023, 22(2):58-61.

陕西教育科学十四五规划 2024 年一般课题项目编号：
SGH24Y2434