

新工科背景下融合计算思维的Python程序设计教学改革实践探索

陈世峰¹, 尹艳兰², 桑海涛¹, 高秀娥²

¹岭南师范学院电子与电气工程学院, 广东 湛江

²岭南师范学院计算机与智能教育学院, 广东 湛江

收稿日期: 2024年3月3日; 录用日期: 2024年4月2日; 发布日期: 2024年4月9日

摘要

在新工科背景下, 针对Python程序设计课程的教学改革进行了深入实践探索。针对当前课程教学现状中存在的问题, 提出一些教学改革思路: 融合思政元素与编程教育, 深化课程内涵; 优选教学资源并强化师生互动, 构筑富有活力的教学新平台; 实施因材施教的教学策略, 缩小基础差距促进共同进步; 融入竞赛机制, 以赛促学, 提升学生的实践能力和创新能力; 构建多元化的评价体系, 实施全过程考核, 全面、客观地评价了学生的学习成效。教学实践表明, 学生在计算思维、自学能力、实践能力等方面得到了显著提升, 课程教学质量和学生的学习体验均得到了大幅改善。

关键词

新工科, 计算思维, Python程序设计, 教学改革

A Practical Exploration of Python Programming Teaching Reform Integrating Computational Thinking under the Background of New Engineering

Shifeng Chen¹, Yanlan Yin², Haitao Sang¹, Xiue Gao²

¹School of Electronic and Electrical Engineering, Lingnan Normal University, Zhanjiang Guangdong

²School of Computer Science and Intelligence Education, Lingnan Normal University, Zhanjiang Guangdong

Received: Mar. 3rd, 2024; accepted: Apr. 2nd, 2024; published: Apr. 9th, 2024

文章引用: 陈世峰, 尹艳兰, 桑海涛, 高秀娥. 新工科背景下融合计算思维的 Python 程序设计教学改革实践探索[J]. 教育进展, 2024, 14(4): 145-150. DOI: 10.12677/ae.2024.144490

Abstract

In the context of new engineering, an in-depth practical exploration was conducted on the teaching reform of Python programming courses. In view of the problems existing in the current course teaching status, some teaching reform ideas are put forward: integrating ideological and political elements and programming education to deepen the content of the course; optimizing teaching resources and strengthening teacher-student interaction to build a dynamic new teaching platform; implementing teaching strategies that teach students in accordance with their aptitude; narrow the basic gap and promote common progress; integrate into the competition mechanism, promote learning through competition, and enhance students' practical and innovative abilities; build a diversified evaluation system, implement full-process assessment, and comprehensively and objectively evaluate students' learning effectiveness. Teaching practice shows that students have been significantly improved in terms of computational thinking, self-learning ability, and practical ability, and the quality of course teaching and students' learning experience have been greatly improved.

Keywords

New Engineering, Computational Thinking, Python Programming, Teaching Reform

Copyright © 2024 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

随着科技日新月异的发展以及数字时代浪潮的到来,新工科应运而生。自2016年“新工科”概念提出后,形成了“复旦共识”“天大行动”“北京指南”,发布了《教育部高等教育司关于开展新工科研究与实践的通知》[1]《教育部办公厅关于公布首批“新工科”研究与实践项目的通知》[2],明确提出对新工科人才培养的需求,成为高等教育领域中的热门话题。新工科不仅涵盖了传统工程学科的内容,还融合了计算机科学、数据科学、人工智能等新兴技术,致力于培养具备创新精神、实践能力和跨界融合能力的新型人才。

在新工科的背景下,计算思维作为一种重要的思维方式,已逐渐成为大学生不可或缺的核心素养。计算思维,即运用计算机科学的基础概念来求解问题、设计系统并理解人类行为的思维方式,自周以真[3]教授给出其定义以来,便受到了广泛的关注。它不仅能够助力学生深刻剖析复杂的工程问题,更能锻炼学生的逻辑思维、创新思维,并提升其解决现实生活中挑战的能力,从而全面提升学生的综合素质和竞争力。

Python语言,以其简洁、易读、功能强大和跨平台等特性,在新工科教育中占据一席之地,成为培养学生计算思维能力的关键工具。当前,关于Python程序设计课程改革的研究层出不穷。贾萍等人[4]通过引入项目教学法、案例教学法和任务驱动教学法,努力实现从知识灌输到能力培养的转变;王雷等人[5]构建了包含编程实践能力、计算思维能力和自主学习能力的课程能力模型,用以指导教学改革;周旭[6]通过分层次设计教学内容,采用情景教学法等方式进行教学,并取得了良好效果。然而,现有研究对计算思维的培养仍显薄弱,未能全面适应新工科人才的多维需求。因此,本文在汲取前人智慧的基础

上,力求探索计算思维培养与 Python 语言学习的有机融合之道,构建一套系统的教学体系,以契合新工科背景下对程序设计人才的全面要求。

2. 课程教学现状

目前,Python 程序设计课程已成为高校计算机教育的重要组成部分,但在实际的教学过程中,我们不难发现一些与新工科人才培养目标不相适应的问题。

首先,教学内容过于偏重理论,实践环节相对不足。传统的 Python 教学往往将重点放在语法规则、数据结构等基础理论知识的传授上,而对于算法设计、编程项目实践等应用技能的培养则显得相对薄弱。这种理论与实践脱节的教学方式,不仅难以激发学生的学习兴趣,也无法有效训练他们的编程思维和工程实践能力。因此,学生在面对实际问题时,往往缺乏将理论知识应用于实践的能力,无法有效解决实际问题。

其次,教学资源同质化,缺乏针对不同基础和兴趣学生的个性化教学。目前,市面上的 Python 教材和在线课程大多遵循统一的教学结构和内容,缺乏针对不同基础和兴趣学生的个性化教学资源。这种情况导致部分学生无法找到适合自己的学习路径和发展方向,学习积极性和兴趣受到严重影响。同时,教师也面临着如何满足学生个性化需求的挑战,因为他们需要在有限的课堂时间里照顾到大多数学生的学习进度和兴趣,这无疑增加了教学的难度。

再者,教学方式单一,缺乏有效的课堂互动和学生参与。传统的 Python 教学往往采用以教师为中心的讲授方式,学生处于被动接受的状态,课堂互动和参与严重不足。这种教学方式不仅难以调动学生的学习积极性,也无法有效培养他们的自主学习能力和问题解决能力。同时,由于缺乏有效的课堂互动和学生参与,教师也无法及时了解学生的学习情况和问题,无法及时调整教学策略,导致教学效果不佳。

最后,考核方式单一,难以全面评价学生的编程能力和计算思维。目前,Python 教学的考核方式多以笔试或简单的编程题为主,这种方式虽然能够考察学生的基础知识和编程技能,但无法全面反映他们的计算思维、创新思维以及问题解决能力。单一的考核方式也容易导致学生为应试而学习,忽视了对自身能力和素质的全面提升。同时,这种考核方式也无法准确反映学生的实际编程能力和工程实践能力,导致评价结果不够准确和客观。

3. 教学改革思路

3.1. 思政编程双提升,深化课程新内涵

在新工科教育的广阔天地中,Python 程序设计课程承载着更为深远的意义——它不仅是编程技能的传授,更是社会责任感和职业道德的熏陶。有鉴于此,我们致力于在 Python 程序设计的教学中巧妙融入思政元素,以期实现课程思政的深远目标。

Python 语言本身蕴含着丰富的思政资源,这些资源如同宝藏般等待我们去发掘。Python 社区的开源精神,正是团队协作、知识共享与无私奉献的生动体现,这与社会主义核心价值观中的和谐、友善等理念不谋而合。同时,优秀程序员的职业道德案例,如同明灯一般,照亮学生前行的道路,引导他们在编程的道路上坚守诚信、勇于担当。

为了进一步将思政教育与编程实践紧密结合,我们精心设计了系列具有思政意义的编程任务。这些任务紧密围绕社会公益项目和社会热点问题展开,如开发助老助残的应用程序、设计环保主题的互动游戏等。在这样的编程实践中,学生不仅能够提升编程技能,更能在潜移默化中体会和践行社会主义核心价值观,深刻领悟职业操守和社会责任感的重要性。

3.2. 优选资源强互动，构筑教学新平台

针对当前 Python 程序设计教学中资源分散、内容同质化的问题，我们致力于整合线上线下优质教学资源，打造一个稳固且多元的教学平台。经过深入筛选与整合，我们选择了学习通和头歌实践教学平台作为本课程的线上教学平台，它们不仅提供了丰富的基础理论知识，还汇聚了大量实践案例、项目库及在线编程练习，为学生提供从理论到实践的全方位学习体验。

同时，注重教学平台的交互性和智能性建设，通过在超星学习通和头歌平台上设置在线讨论区、智能答疑系统等功能，实现了师生之间、学生之间的高效交流与协作。这种即时互动的学习方式不仅激发了学生的学习热情，还帮助他们在交流中深化对 Python 编程的理解。此外，我们还与企业合作，引入行业最新技术和项目案例，确保教学内容与行业需求紧密对接，为学生搭建起通往实际工程应用的桥梁。这样，学生在学习过程中就能接触到真实的工程应用场景，为未来的职业发展奠定坚实基础。

3.3. 因材施教补基础，缩小差距共进步

在 Python 程序设计的教学过程中，学生基础参差不齐是一个不可忽视的挑战。为了有效应对这一问题，我们积极倡导并实践因材施教的教学理念。课前通过精心设计的测试和问卷调查，深入了解每位学生的编程基础、兴趣偏好和学习风格。这些宝贵的数据为我们后续的个性化教学提供了有力的支撑。

基于这些信息，我们精心将学生划分为不同的学习层次，并为每个层次量身定制了贴切的教学目标和内容。对于基础相对薄弱的学生，我们着重夯实他们的基础，通过生动有趣的入门案例和实操练习，循序渐进地引导他们掌握 Python 编程的核心概念和基本语法。而对于基础较好的学生，我们则通过引入更高级的编程技术和复杂算法，不断挑战他们的思维边界，助力他们的编程能力和计算思维更上一层楼。

同时，注重小组合作学习的运用。在教学过程中，积极鼓励并组织学生开展小组合作学习。通过精心组建的异质小组，学生们在相互合作中相互学习、相互启发，不仅实现了知识的互补和共享，更在无形中培养了他们的团队协作精神和沟通能力。这种个性化的教学方式，不仅有效缩小了学生之间的基础差距，更让每位学生都能在 Python 程序设计的学习旅程中找到属于自己的进步之路。

3.4. 融入竞赛激兴趣，以赛促学提能力

在传统的 Python 程序设计课程，学生往往难以感受到学习兴趣和动力。为了改变这一现状，我们提出引入竞赛机制，旨在通过竞技的激情点燃学生对 Python 编程的热爱，进而提升他们的学习积极性和实践能力。

竞赛机制为学生搭建了一个展示编程才华的舞台。在这个平台上，学生有机会将课堂所学的 Python 知识应用于实际问题的挑战中，从而深刻体会到编程带来的成就感和喜悦。这种即时的正向反馈，像是一剂强心针，进一步激发了学生的探索欲望和学习动力，引领他们更深入地徜徉在 Python 编程的世界。

同时，竞赛机制的引入也对教学内容和方法的创新提出了更高要求。为了让学生在竞赛中脱颖而出，教师需要紧跟技术潮流，不断更新教学内容，将最新的编程技术和工具引入课堂。在教学方法上，教师也需要更加注重学生的主体地位，采用灵活多样的教学方式，如项目驱动、团队协作等，以更好地激发学生的学习兴趣和创造力。这种以赛促教、以教助赛的良性循环，将有力推动 Python 程序设计课程的持续创新和发展。

最后，为了让竞赛机制更好地融入教学体系中，我们还将定期组织小组展示和评比活动。在这些活动中，表现优异的小组将有机会获得成绩的附加分，并代表学校参加更高级别的编程竞赛，如“太极杯”软件编程大赛、中国大学生程序设计竞赛以及 ACM 国际大学生编程竞赛等。这些竞赛不仅是对学生技能的进一步锤炼，更是他们拓宽视野、增进交流的重要平台。

3.5. 多元评价全过程，全面考核促发展

为了更加公正、客观地评价学生的学习成果，构建多元评价体系，并实施过程性考核。在这一评价体系中，不仅引入了传统的教师评价，还鼓励学生进行自我评价和同伴评价，以形成多角度、全方位的评价机制。自我评价将帮助学生更深入地反思自己的学习状况，激发他们的自主学习意识和自我提升动力。同伴评价则能促进学生间的互动与合作，让他们在相互的反馈中发现不足，从而调整学习策略，提高学习效果。

在评价方式方面，采用笔试、机试、项目评价等多种形式，以全面、深入地评估学生的 Python 编程能力和实践应用水平。其中，笔试主要考察学生对 Python 基础知识的掌握情况，机试则着重检验学生的编程能力和实践应用水平。而项目评价则是一种更加综合的评价方式，它要求学生完成一个完整的 Python 编程项目，并从项目的需求分析、设计、实现、测试等多个方面进行评价。这种评价方式不仅可以全面反映学生的编程能力，还可以培养他们的团队协作精神和问题解决能力。

学生的最终成绩将由期末成绩和过程性考核成绩两部分组成，各占 50% 的权重。期末成绩采用闭卷机试的方式，涵盖选择题、填空题、简答题、编程题等多种题型，全面考核学生对 Python 知识的理解和掌握程度。过程性考核则更加注重对学生的学习和表现，主要由平时表现(5%)、随堂测验(15%)、平台学习情况(20%)、实验报告(30%)、综合项目报告(30%)以及附加分六种考核形式组成。平时表现考查学生课堂表现和学习态度，如出勤率，课堂互动和积极性等；随堂测验侧重考查学生对基本理论知识的理解掌握情况，以检验他们的学习效果；平台学习情况侧重考查学生在教学平台(如学习通、头歌教学平台等)上的自主学习情况，如课前预习测试，课后作业完成度等；实验报告侧重考查学生的理论知识应用能力和编程实践能力，要求他们能够将所学知识灵活运用 to 实际编程中；而综合项目报告则更加注重学生的综合运用能力、团队协作精神和创新思维能力，要求他们能够独立完成一个具有实际应用价值的 Python 编程项目，并能够清晰地阐述项目的思路、过程和成果。

值得一提的是，在过程性考核中引入了附加分机制。学生可以通过参与 Python 相关的竞赛活动来争取附加分。根据获奖情况，学生将获得相应的加分，但最后的总分不能超过过程性考核总分。

4. 教学改革成效

在 2020 级物联网工程专业的 Python 程序设计课程中，我们实施了一系列的教学改革措施，为了客观评估这些改革的实际效果，我们从成绩分析和问卷调查两种方式进行深入探究。

从成绩分析来看，学生在超星学习通平台上的平时表现和随堂测验平均成绩为 83.62，显示出他们在课程学习中的扎实基础和良好表现。在头歌实践教学平台上，学生通过完成了 8 次实践项目测验，不仅锻炼了自己的实践能力，也加深了对理论知识的理解，平均分达到了 86.23 分。而在课程设计的分组项目报告中，学生们更是通过团队合作，成功完成了 Python 实训项目，并进行了项目答辩，各组平均成绩达到了 81.36 分。然而，期末闭卷考试的成绩分析也暴露出一些问题。全体参考学生 47 人，60 分以下者 8 人、90 分以上者 1 人，平均成绩 67.52 分，卷面成绩及格率达 82.98%。通过试卷分析我们发现，失分主要集中在考查学生程序设计能力的程序填空类型的试题方面，以及在字典、列表等基本概念的理解与运用等方面。这表明部分学生在编程实践和基础概念掌握上仍需加强。此外，期末测试还反映出有 10% 的学生平时未能够按要求自主完成头歌实践教学平台和超星学习通平台上相关的学习任务，这也提醒我们在未来的教学中需要进一步加强学生的学习监督和引导。

在问卷调查方面，我们设计了涵盖教学内容、教学方法、教学资源、师生互动、学习成效等多个维度的问卷，并收集了学生们的匿名反馈。结果显示，绝大多数学生对 Python 程序设计课程的教学内容表示满意，认为课程内容丰富、实用，能够很好地满足他们的学习需求。同时，学生们对教学方法也给予

了高度评价，他们认为教师采用的案例教学、项目驱动等教学方法生动有趣，能够激发他们的学习兴趣和动力。此外，学生们还对教学资源的丰富性和便捷性表示赞赏，他们认为超星学习通平台和头歌实践教学平台为他们提供了丰富的学习资源和便捷的学习途径。最后，他们也表示在分组项目报告中，通过与小组成员的交流和合作，不仅提升了自己的团队协作能力，也加深了对课程内容的理解。

5. 结语

本文对新工科背景下融合计算思维的 Python 程序设计教学进行了改革实践探索，通过深入挖掘思政元素与编程教育的契合点，优化了教学资源配置，实施了因材施教的教学策略，同时巧妙地融入了竞赛机制，构建了多元化的考核体系，达到了提升教学质量的目的，锤炼了学生的计算思维、自主学习能力以及动手实践能力。本文在借鉴前人智慧的基础上，提出了自己的一点拙见和尝试，期望能为新时代的 Python 程序设计课程教学改革添砖加瓦。

基金项目

岭南师范学院 2021 年度教育教学改革项目“新工科背景下以计算思维为导向的 Python 程序设计教学改革研究”。

参考文献

- [1] 教育部高等教育司. 教育部高等教育司关于开展新工科研究与实践的通知[EB/OL]. http://www.moe.gov.cn/s78/A08/tongzhi/201702/t20170223_297158.html, 2024-02-23.
- [2] 教育部办公厅. 教育部办公厅关于公布首批“新工科”研究与实践项目的通知[EB/OL]. http://www.moe.gov.cn/srcsite/A08/s7056/201803/t20180329_331767.html, 2024-02-23.
- [3] Wing, J.M. (2006) Computational Thinking. *Communications of ACM*, **49**, 33-35. <https://doi.org/10.1145/1118178.1118215>
- [4] 贾萍, 柳欣. 面向创新实践能力培养的计算机基础课程教学改革与实践研究[J]. 电脑知识与技术, 2023, 19(21): 131-134.
- [5] 王雷, 秦琳琳, 王嵩. 面向新工科人才能力培养的计算机程序设计课程改革[J]. 高教学刊, 2023, 9(24): 161-164.
- [6] 周旭. 混合式教学模式下的“Python 程序设计基础”课程教学改革探索[J]. 教育教学论坛, 2023(18): 49-52.