

AI大模型技术在编译原理课程教学中的应用

王焱林, 毛明志, 黄炎贤, 王乐球, 郑子彬

(中山大学软件工程学院, 广东 珠海 519082)

摘要: 编译原理课程具有基础概念抽象复杂、对实践能力要求高等特点, 给学生学习带来一定的挑战。因此, 针对课程教学中存在的各种困难, 开展AI大模型技术与编译原理课程设计相结合的实践创新及探索。借助AI大模型技术将编译原理理论课程中的抽象概念形象化、复杂算法简单化、涉及范围广泛且深奥的课程内容具体化, 并将实践课程中实验代码的理解过程方便化、生成过程高效化、错误查改过程高质量化。通过应用大模型技术, 丰富了编译原理课程教学方式, 探索了课程实践新方向, 并显著提高了课程学习效果。

关键词: 编译原理; AI大模型技术; 课程设计; 课程实践

DOI: 10.11907/rjdk.241696

中图分类号: G434

文献标识码: A

开放科学(资源服务)标识码(OSID):

文章编号: 1672-7800(2025)008-0201-06



Application of AI Large Language Model Technology in Compiler Principles Course Teaching

WANG Yanlin, MAO Mingzhi, HUANG Yanxian, WANG Leqiu, ZHENG Zibin

(School of Software Engineering, Sun Yat-Sen University, Zhuhai 519082, China)

Abstract: The basic concept of the compiler principles course is abstract and complex, and the practical ability is highly required, which brings some difficulties and challenges for students to learn this course. To address these teaching difficulties existing in this course, the idea of integrating AI large model technology to practice innovation and exploration of the compiler principles course is put forward. By leveraging AI large model technology, abstract foundational concepts in the theoretical portion of the course can be concretized, complex core algorithms simplified, and the broad and profound course content made more accessible; In the practical portion, the understanding of experimental code is facilitated, the process of code generation is made more efficient, and the quality of code error detection and correction is enhanced. By combining the large model technology, the teaching method of the compilation principle course is enriched, and the learning effect of the course is significantly improved.

Key Words: compiler principles; AI large language model technology; course design; curriculum practice

0 引言

随着AI大模型技术的迅猛发展, 模型规模不断增长, 性能持续提升, 应用场景也日益丰富。已有的大模型如ChatGPT和GPT-4在自然语言处理、软件工程、计算机视觉等领域展现了强大的能力, 推动了人工智能技术的发展与应用^[1]。与此同时, 编译原理课程作为计算机科学领域

的重要基础课程, 关注程序设计语言的理论与实践, 涵盖了词法分析、语法分析、语义分析等关键内容。已有工作对编译原理课程的设计与实践进行了一定的探索和改革, 但还没有工作将先进的AI大模型技术引入其中^[2-9]。将大模型技术尤其是代码大模型与编译原理课程相结合, 能够提供更智能、个性化的学习体验, 例如通过大模型生成器提供编程语言规范、利用自然语言处理技术解释复杂概念, 以及开发智能编译器辅助工具进行实时指导、反馈

收稿日期: 2024-08-11

扫描二维码阅读全文:



基金项目: 全国高等学校计算机教育研究会项目(CERACU2023R02); 广东省自然科学基金项目(2024A1515010255)

作者简介: 王焱林(1993-), 女, 博士, CCF高级会员, 中山大学软件工程学院助理教授, 研究方向为大模型、软件工程、编程语言及其交叉领域; 毛明志(1966-), 男, 博士, CCF杰出会员, 中山大学软件工程学院副教授, 研究方向为软件工程、人工智能、计算机教育等。本文通讯作者: 毛明志。

等^[10,11]。这种结合不仅能提高编译原理课程教学效果,而且能推动 AI 技术在教育领域的应用。本文对 AI 大模型技术与编译原理课程的结合进行了初步探索,结合 AI 大模型技术对编译原理课程设计进行改革,将大模型技术作为强大的教学工具应用于教学设计与实践中,以丰富编译原理课程教学方式,提高课程学习效果。

1 编译原理课程研究现状

编译原理课程是计算机科学领域的重要课程之一,旨在教授计算机程序设计语言的翻译和编译技术,涵盖了从源代码到可执行代码的整个编译过程,内容包括词法分析、语法分析、语义分析、中间代码生成、代码优化以及目标代码生成等。该课程不仅要求学生能够理解、掌握理论知识,而且要求学生具备编写、调试简单编译器的实践能力。通过学习该课程,学生应达到以下目标:

(1)理解编译过程。通过课程学习,学生能够理解编译器将高级编程语言转换为机器语言的整个过程,包括词法分析器和语法分析器设计与实现、语义动作执行、中间代码生成、代码优化以及目标代码生成等基本技术。

(2)分析和设计编译器。学生能够分析编程语言的语法和语义,并设计相应编译器对源代码进行编译,生成最终的目标代码。

(3)理解编译器优化。学生能够理解编译器优化的原理和方法,包括常量传播、死代码消除、循环优化等技术,并应用这些技术提高程序性能。

(4)提高实践能力。通过理论课程学习以及实验项目实践,学生在加深对编译原理理论知识理解的同时,能够掌握编译器开发中常用的工具、技术,编写简单的编译器前端或后端代码。

目前编译原理课程设计与教学方法受到国内外高校的广泛关注,课程设计普遍注重理论与实践相结合,并采用项目驱动的教学方法,引导学生通过项目实践深入理解课程内容^[4-5]。同时,国内许多高校都十分关注编译原理课程教学模式的改革和创新,如武汉理工大学将在线教学模式引入编译原理课程设计中,提出编译原理在线云课堂的设计思路,建立 SPOC 课程资源+云课堂+线上实验的三位一体编译原理课程在线教学模型^[6];辽宁师范大学将翻转课堂教学模式引入计算机专业课程教学中,提出“智慧树成长模式图”,构建了基于“A+课堂派”的编译原理翻转课堂教学新模式^[7]。随着人工智能、大数据等技术的发展,编译原理与大模型技术相结合已是大势所趋,但相关研究工作还比较欠缺。

2 编译原理课程教学难点

编译原理作为计算机科学及相关专业的核心课程,对

于学生深入理解计算机系统具有重要价值。课程的学习过程涉及计算机系统各个层面,包括硬件、软件及其交互、程序执行过程等。通过编译原理课程的学习,学生能够深入理解程序设计语言的本质、程序执行过程以及编程工具实现原理,培养计算机科学素养和解决实际问题的能力。

编译原理作为计算机专业核心课程,其自身特点也给课程学习带来一定的困难和挑战,具体表现如下^[12]:

(1)理论知识复杂抽象。编译原理课程涉及许多抽象概念和复杂算法,如正则表达式、有限自动机、上下文无关文法、语法树等。这些概念对于初学者来说可能较为晦涩,需要耗费一定时间、精力才能理解并掌握。

(2)对编程技能有所要求。编译原理课程通常会涉及编程实践,例如使用词法分析器生成器与语法分析器生成器构建编译器前端或者实现中间代码生成、优化等,学生需要具备一定的编程技能才能完成相关实践项目。

(3)涉及知识范围广泛。编译原理课程涉及数据结构与算法、计算机体系结构、操作系统等众多计算机相关学科知识。此外,还需要掌握一定的数学知识,如集合论、离散数学、图论等。因此,在学习编译原理课程的过程中,学生需要掌握跨学科知识以理解并应用编译器设计与实现原理,给不少学生带来一定的挑战。

随着人工智能等领域的发展以及 AI 大模型技术的出现,解决上述编译原理课程教学难点有了新的探索方向和可行方案。借助大模型技术,学生能够:①通过生动具体的示例更好地理解复杂抽象的概念、理论;②在编译原理课程实践过程中,根据大模型提供的实际编程指导来增强自身的编程能力,提高实验效率与学习效果;③根据自己的学习需求和进度得到个性化的学习指导,从而更有针对性地学习、掌握相关知识。因此,本文提出结合 AI 大模型技术的编译原理课程设计方案,将大模型作为高效的教学工具引入课堂,借助其出色的上下文理解与对话能力,丰富了编译原理课程设计方式,为大模型技术与计算机相关课程教学设计结合提供了参考。

3 编译原理课程设计

3.1 AI 大模型技术与编译原理课程教学设计

AI 大模型技术在各个领域都展现出了巨大潜力,借助 AI 大模型出色的自然语言理解和生成能力、代码理解和生成能力以及优越的上下文理解和对话能力,设计编译原理课程,将 AI 大模型作为强大的教学工具应用于编译原理课程各个环节的教学设计中,见图 1。

3.1.1 理论课程

(1)基础概念理解。编译原理课程中的基础概念具有抽象难懂的特点,学生理解起来较为困难。因此,可以借助 AI 大模型生成技术,让大模型联系实际生活生成生动的示例,将抽象概念转换成具象实例,帮助学生更好

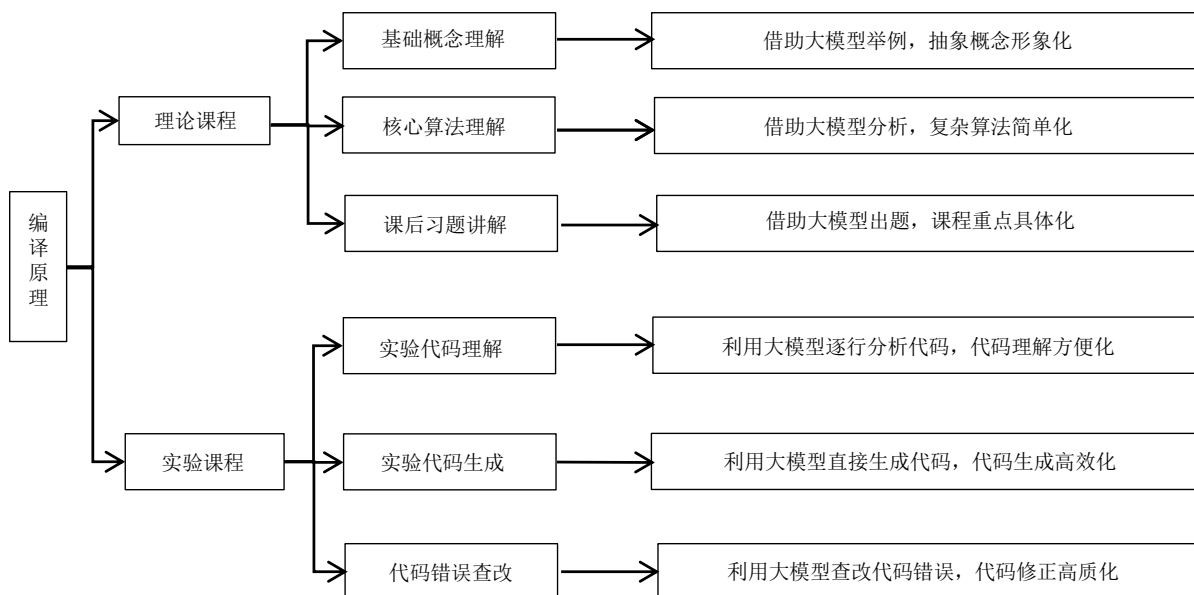


Fig. 1 Course design diagram of compiler principles combined with AI language model technology

图1 结合AI大模型技术的编译原理课程设计图

地理解和掌握基础概念。

(2)核心算法理解。编译原理课程中涉及多个复杂的核心算法,尽管伪代码形式已经一定程度上降低了理解难度,但是对于初次学习编译原理课程的学生来说还是存在挑战。因此,可以利用AI大模型出色的自然语言处理技术,让大模型对复杂的核心算法进行解释,并给出例子说明算法的具体处理过程,借助具体、直观的实例帮助学生更快地理解和消化知识。

(3)课后习题讲解。课后习题是检验课堂知识掌握情况的有效途径之一,传统的编译原理课程教学中通常仅利用教材习题对课堂知识进行检验和巩固。该方式不但针对性较弱,且教材中针对习题的讲解并不详细,往往需要教师再根据学生的反馈情况进行习题辅导,而利用大模型技术生成课后习题能够很好弥补上述缺陷。具体地,利用AI大模型出色的自然语言理解和生成能力,可针对编译原理理论课程中的重点内容生成多个课后习题并给出详细解析过程,以帮助学生更好地检验知识掌握情况,从而进一步巩固薄弱知识。此外,学生还可通过该方式生成有针对性、个性化的课后习题进行自我检测和自我巩固。

3.1.2 实验课程

(1)实验代码理解。在编译原理实验的实现过程中,首先需要理解所给代码的框架,并在此基础上实现所需的代码。得益于大模型出色的代码理解能力,学生可以在代码理解过程中利用大模型对实验代码进行逐行解释,以更好地理解实验代码。该方式不仅简化了代码理解过程,而且降低了代码理解成本。

(2)实验代码生成。得益于大模型出色的代码生成能力,在掌握了编译原理相关理论知识并理解了所给实验代码框架的基础上,学生可以将所要实现的代码功能告诉大模型,利用大模型生成代码,并对大模型生成的代码进行

一定调整以实现实验目标,从而大大提高了代码生成效率。

(3)代码错误查改。在代码调试过程中遇到错误时,学生可以借助大模型进行错误查改。利用大模型进行语法检查或者给出错误提示与修改建议等,并基于大模型所给的回答进行代码修改,从而提高了代码调试效率和生成代码质量。

3.2 AI大模型技术与编译原理课程教学实践

3.2.1 基础概念理解

学习编译原理中的子集构造法会涉及到NFA、DFA、 ϵ -closure(s)、 ϵ -closure(T)等抽象概念,借助大模型可以类比现实生活并给出具体例子,将实际生活与这些抽象概念联系起来,从而帮助理解概念含义以及如何通过各个操作进行状态转换,最终掌握子集构造法的算法过程和核心原理。具体地,将“请联系实际,举一个具体的例子将编译原理课程中的子集构造法与实际生活联系起来,帮助理解相关抽象概念”作为提示输入ChatGPT,图2是利用ChatGPT得到的类比例子。将舞会类比子集构造法中的各个概念和对象,并基于此对子集构造法进行说明,从而生动、形象地介绍了该算法的具体操作过程。

3.2.2 核心算法理解

除了通过联系实际,用实际生活场景类比编译原理相关知识来理解抽象概念,从而掌握相关理论知识以外,还能够借助大模型对核心算法作出简单易懂的解释并进一步给出具体示例。通过具体示例演示算法中各个变量的转变过程,加深对核心算法的理解,也为后续实验过程中实现算法奠定良好的理论基础。例如,将提示“请解释编译原理中的预测分析算法并进一步给出具体示例,通过具体示例演示算法运算过程中各个变量的转变过程”作为ChatGPT的输入,图3、图4给出了学习预测分析算法时,利

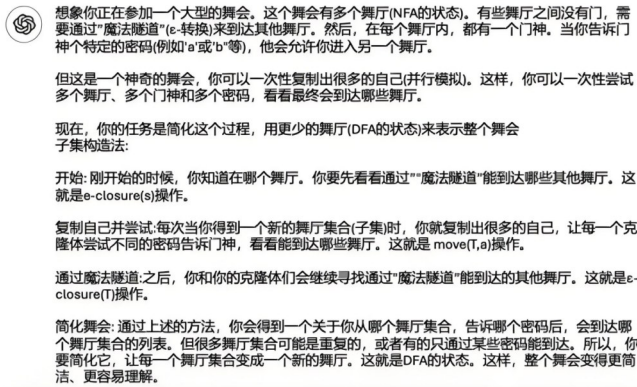


Fig. 2 Screenshot of the example from ChatGPT to illustrate the subset construction method

图 2 ChatGPT 类比实例说明子集构造法截图

用大模型给出实例说明该算法过程的部分截图。

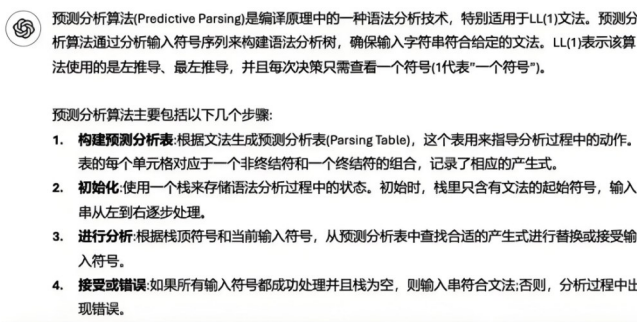


Fig. 3 Screenshot 1 of ChatGPT illustrating the predictive analytics algorithm

图 3 ChatGPT 说明预测分析算法截图 1

示例

假设我们有一个简单的LL(1)文法,如下所示:



我们的任务是分析输入串 ab, 是否符合这个文法。

1. 构建预测分析表

根据上述文法, 我们构建预测分析表如下:

	a	b
S	S → AB	
A	A → a	

Fig. 4 Screenshot 2 of ChatGPT illustrating the predictive analytics algorithm

图 4 ChatGPT 说明预测分析算法截图 2

3.2.3 课后习题讲解

将提示“请生成编译原理中关于子集构造法的习题, 并给出详细的解题过程”作为 ChatGPT 的输入, 图 5、图 6 给出了利用大模型生成子集构造法相关习题的例子, 大模型生成了习题并给出了详细解题过程。基于此学生能够通过对比自己的解题过程和参考题解进行自我检验, 存在疑问时也能够通过参考题解进一步学习、巩固相关知识。

子集构造法习题

题目:

给定一个确定的 NFA (非确定有限状态自动机), 请使用子集构造法将其转换为等价的 DFA (确定有限状态自动机)。

设定如下 NFA:

- 状态集: {q0, q1, q2, q3}
- 字母表: {a, b}
- 初态: q0
- 终态集: {q3}
- 转移函数:
 - $\delta(q0, a) = \{q0, q1\}$
 - $\delta(q0, b) = \{q0\}$
 - $\delta(q1, a) = \{q2\}$
 - $\delta(q2, b) = \{q3\}$

Fig. 5 Example 1 of the exercises of subset construction method generated by ChatGPT

图 5 ChatGPT 生成子集构造法习题示例 1

解题步骤:

1. 初始状态:

我们从 NFA 的初态 q0 开始, 构造 DFA 的初态。由于 q0 本身就是初始状态, 同时 q0 没有 e-闭包的扩展, 因此 DFA 的初态是 {q0}。

2. 构造 DFA 状态:

在子集构造法中, DFA 的状态是 NFA 状态的集合。我们对每个状态集合进行扩展:

a. 从初始状态 {q0} 出发

• 处理字符 a:

- $\delta(q0, a) = \{q0, q1\}$, 所以从 {q0} 读取 a 后, 转移到 DFA 状态 {q0, q1}。

• 处理字符 b:

- $\delta(q0, b) = \{q0\}$, 所以从 {q0} 读取 b 后, 仍然回到 DFA 状态 {q0}。

b. 从状态 {q0, q1} 出发

• 处理字符 a:

Fig. 6 Example 2 of the exercises of subset construction method generated by ChatGPT

图 6 ChatGPT 生成子集构造法习题示例 2

3.2.4 实验代码理解

下文是从编译原理实验二语义分析中选取的某个函数实现算法:

```

Entry findSymbolAll(char* name) {
    unsigned int hash = hash_pjw(name);
    Entry tmp = symbolTable[hash];
    while (tmp != NULL) {
        if (strcmp(tmp->name, name) == 0 && tmp->type != NULL
        && tmp->type->kind != ENUM_FUNC) {
            break;
        }
        tmp = tmp->hashNext;
    }
    return tmp;
}
  
```

将提示“请对下面代码逐行进行分析解释: {待解释代码}”作为 ChatGPT 的输入, 利用大模型对以上代码进行解释。图 7 是大模型给出的回答, 逐行对代码进行了解释, 并概括了该函数所实现的功能。



Fig. 7 Example of code explanation from ChatGPT

图7 ChatGPT进行代码解释示例

3.2.5 实验代码生成

在编译原理实验二语义分析实验的实现过程中,需要实现和维护一个符号表,其中一个实现方式是通过散列表来构造符号表,向大模型发送请求“请用散列表实现一个符号表,用于实现编译原理实验中的语义分析实验”。图8是大模型给出的回答,实现了符号表构造以及相应的初始化符号表、在表中插入符号、在表中查找符号等函数。

代码实现



Fig. 8 Example of ChatGPT generating code

图8 ChatGPT生成代码示例

3.2.6 代码错误查改

给出错误代码以及对应的报错信息,大模型能够针对错误代码分析报错原因,给出修改意见,并基于修改意见进一步给出修改后的代码。图9、图10是借助大模型进行错误查改的一个例子。其中,图9给出输入大模型的具体提示,图10给出大模型生成的回答。

4 结语

教学是高等学校的中心工作,课程建设是提高教学质量最重要的环节^[13-15]。将AI大模型技术与编译原理课程教学设计与实践相结合,对传统的编译原理课程教学进行了创新,显著提高了课程学习效果。实施新修订的2022

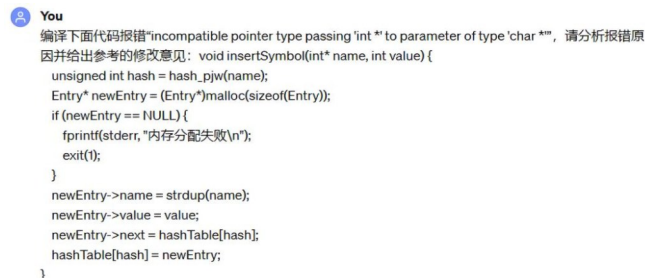


Fig. 9 Example 1 of ChatGPT detecting and correcting coding errors

图9 ChatGPT进行代码错误查改示例1

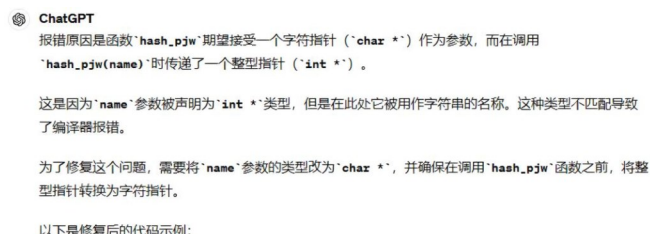


Fig. 10 Example 2 of ChatGPT detecting and correcting coding errors

图10 ChatGPT进行代码错误查改示例2

课程教学方案后,与中山大学2021级和2020级软件工程专业学生的学习效果进行比较,结果如下:①2022级学生学习兴趣提升,课堂参与度高,参与课堂互动140余次,缺勤仅两人次,缺勤率下降了95.1%;②2022级学生期末测试全员合格,理论课程期末测试平均分86.97,提高了1.48%;实验课程平均分96.41,提高了3.12%。

将AI大模型技术与编译原理课程设计相结合,可以提供更智能化、个性化的学习体验。通过AI大模型,学生可以获得更直观的交互式学习环境(包括自动生成的丰富实例、定制化的解释指导以及大模型作为辅助工具所给的实时反馈),从而加深对编译原理核心概念的理解。将先进的AI大模型技术与传统编译原理课程设计相结合,扩展了编译原理课程的教学设计手段,丰富了教师教学与学生学习的方式。未来,教学团队将针对编译原理课程的其他教学难点,结合大模型技术探索更加巧妙的解决之道。本文编译原理创新方案的提出为计算机类相关专业课程改革创新和高质量的软件工程课程建设提供了参考。

参考文献:

- [1] YANG J F, JIN H Y, TANG R X, et al. Harnessing the power of LLMs in practice: a survey on ChatGPT and beyond [J]. ACM Transactions on Knowledge Discovery from Data, 2024, 18(6): 1-32.
- [2] SHU Z M, LI W J, ZHOU X C. Practice and experience of teaching reform for compiler principle [J]. Acta Scientiarum Naturalium Universitatis Sunyatseni, 2007(S2): 101-104.
舒志梅,李文军,周晓聪. 编译原理教学改革实践初探(英文)[J]. 中山大学学报(自然科学版), 2007(S2): 101-104.
- [3] ZHANG D M, FANG X W. Teaching practice and reform of principles of

- compilers course design[J]. Research and Exploration in Laboratory, 2012, 31(11):134-137, 153.
- 张冬莱, 方习文. 编译原理课程设计的教学实践与改革[J]. 实验室研究与探索, 2012, 31(11):134-137, 153.
- [4] GAO X Y, ZHANG C X. Teaching reform of compiler principle course combining case and project[J]. Journal of Higher Education, 2024, 10(3): 64-67.
- 高雪瑶, 张春祥. 案例与项目相结合的编译原理教学改革[J]. 高教学刊, 2024, 10(3):64-67.
- [5] ZHANG Y J, LIU H B, ZHENG Q L, et al. Research on the practice of compiling principle teaching based on the ability training of engineering applied talents[J]. Software Guide, 2014, 13(1):171-173.
- 张亚娟, 刘寒冰, 郑全录, 等. 基于工程应用型人才培养的编译原理教学实践研究[J]. 软件导刊, 2014, 13(1):171-173.
- [6] RAO W B, WANG Y H, LIN H, et al. On-line teaching of compiler principle course based on cloud classroom[J]. Software Guide, 2020, 19(12): 81-84.
- 饶文碧, 王云华, 林泓, 等. 基于云课堂的编译原理课程在线教学探究[J]. 软件导刊, 2020, 19(12):81-84.
- [7] MIAO J Y. E-learning in the compilation principle flipped classroom application model——taking the course of compilation principle as an example[J]. Software Guide, 2020, 19(12): 281-284.
- 苗婧怡. 移动学习在翻转课堂中的应用模型研究——以编译原理课程应用为例[J]. 软件导刊, 2020, 19(12): 281-284.
- [8] NIE W M, SONG X X. Mixed teaching reform based on the compilation principle of PAD class[J]. Computer Education, 2023(10):107-111.
- 聂文梅, 宋晓霞. 基于对分课堂的编译原理混合教学改革[J]. 计算机教育, 2023(10):107-111.
- [9] ZHAI J L. Research on diversified teaching strategies of compiler principle courses based on the combination of online and offline[C]//Proceedings of the Third International Conference on Innovative Talent Training and Sustainable Development in 2023, 2023:298-299.
- 翟建丽. 基于线上线下结合的编译原理课程多样化教学策略探究[C]//2023年第三届创新人才培养与可持续发展国际学术会议论文集, 2023:298-299.
- [10] ZHANG Z Y, CHEN C Y, LIU B C, et al. Unifying the perspectives of NLP and software engineering: a survey on language models for code[DB/OL]. <https://arxiv.org/abs/2311.07989>.
- [11] XU K, JIANG L Y, HUANG H P, et al. Practice method for compiler principles course for international students based on ChatGPT[J]. Software Guide, 2023, 22(9):227-231.
- 徐康, 蒋凌云, 黄海平, 等. 基于ChatGPT的留学生编译原理课程实践方法[J]. 软件导刊, 2023, 22(9):227-231.
- [12] FU Z Y, ZHANG Y. Discussion on the reform of compiler principle teaching[J]. China Education Innovation Herald, 2011(31):41-42.
- 傅朝阳, 张妍. 编译原理教学改革探讨[J]. 中国科教创新导刊, 2011(31):41-42.
- [13] MAO M Z, LIU Z. Discussion on ideological and political teaching of IT project management course under new engineering background[J]. Computer Education, 2022(4):21-23, 28.
- 毛明志, 刘钊. 新工科背景下IT项目管理课程思政教学探讨[J]. 计算机教育, 2022(4):21-23, 28.
- [14] XU W, LIU S, ZHANG Y. Practical teaching reform of compilation principle based on LLVM driver[J]. Software Guide, 2023, 22(6):191-195.
- 徐伟, 刘硕, 张昱. 基于LLVM驱动程序的编译原理实践教学改革[J]. 软件导刊, 2023, 22(6):191-195.
- [15] WANG L Q, CHEN J G, MAO M Z, et al. Ideological and political research of software engineering practical training course based on constructivism theory[J]. Computer Education, 2024(4):101-104.
- 王乐球, 陈建国, 毛明志, 等. 基于建构主义理论的软件工程实训课程思政探究[J]. 计算机教育, 2024(4):101-104.

(责任编辑:黄健 刘晨玥)