

生成式 AI 赋能面向对象程序设计课程教学改革与实践*

杨雨倩 陈嘉敏** 周富肯 冯思源 钟伦 沈瑞琳

广东东软学院人工智能学院, 佛山 528000

摘要 生成式人工智能技术正在改变很多行业, 教育领域也开始受到影响。面向对象程序设计是计算机专业的核心课程, 这门课程教学生用类和对象来编写程序。但在实际教学中, 我们发现学生学习起来有困难, 教师教学也面临挑战。本文分析了目前面向对象程序设计课程教学中存在的主要问题, 比如学生难以理解抽象概念、编程实践能力不足、教学模式单一等。我们提出了一个生成式 AI 赋能的教学改革方案。这个方案把生成式 AI 工具引入到课程教学中, 通过具体的案例来帮助学生理解面向对象的核心思想。实践结果表明, 这种教学方式能够提高学生的学习兴趣, 帮助学生更好地掌握面向对象编程的方法, 同时也培养了学生的计算思维能力。本文为人工智能时代计算机专业课程的教学改革提供了新的思路。

关键字 生成式 AI, 面向对象程序设计, 案例教学, 编程教学, 教学改革

Generative AI Empowers the Teaching Reform and Practice of Object-Oriented Programming Courses

Yang Yuqian Chen Jiamin Zhou Fukun Feng Siyuan Zhong Lun Shen Ruilin

College of Artificial Intelligence
Guangdong Neusoft University,
Foshan 528000, China;

Abstract—Generative artificial intelligence technology is transforming many industries, and the field of education has also begun to feel its impact. Object-Oriented Programming is a core course for computer science majors, teaching students to write programs using classes and objects. However, in actual teaching practice, we have found that students experience learning difficulties and teachers face instructional challenges. This paper analyzes the main problems currently existing in the teaching of Object-Oriented Programming courses, such as students' difficulty in understanding abstract concepts, insufficient practical programming ability, and a monolithic teaching model. We propose a teaching reform plan empowered by generative AI. This plan introduces generative AI tools into the course instruction, using specific examples to help students understand the core ideas of object-orientation. Practical results show that this teaching method can enhance students' learning interest, help them better master object-oriented programming methods, and also cultivate their computational thinking skills. This paper provides new ideas for the teaching reform of computer science courses in the era of artificial intelligence.

Keywords—Generative AI, Object-Oriented Programming, Case-based Teaching, Programming Education, Teaching Reform

1 引言

这几年, 生成式人工智能技术发展得很快。以 DeepSeek、ChatGPT 为代表的 AI 工具能够理解人类的语言, 也能生成代码和解释代码。这些技术已经开始改变我们的学习和工作方式。中共中央、国务院印发的《教育强国建设规划纲要(2024—2035 年)》^[1]中也提到, 要用人工智能来推动教育的改革和创新(中共中央国务院, 2025)。

面向对象程序设计是计算机专业非常重要的一门课。这门课教学生怎么用类和对象来组织代码, 怎么用继承、封装、多态这些思想来解决问题。Java、C++、Python 这些主流的编程语言都支持面向对象的编程方式。学好这门课, 对学生以后的软件开发能力培养很重要。但是, 我们在教学实践中发现, 面向对象程序设计课程存在一些问题。很多学生觉得类和对象这些概念太抽象, 不好理解。学生写代码的时候经常出错, 调试起来也很费时间。教师一个人要教几十个学生, 很难照顾到每个人的学习进度和困难。传统的教学方式大多是老师讲、学生听, 学生动手的机会不够多。

*基金资助: 本文得到中国民办教育协会 2024 年度规划课题(学校发展类)(项目号: CANFZG24122)资助。

**通信作者: 陈嘉敏 756304078@qq.com

国内已经有不少老师对编程课程的教学改革做了研究. 在生成式 AI 与大模型赋能教学方面^[2-6], 余红宴(2025)探索了生成式 AI 在 Python 编程数据分析教学中的应用, 通过 AI 辅助数据分析任务的设计与实现, 提升了学生解决实际问题的能力。宋晓玲和刘磊(2025)构建了基于大模型的软件工程课程 AI 智能助教系统, 实现了智能答疑、代码辅助等功能。

在教学模式创新与能力培养方面^[7-11], 罗雪梅等(2025)开展了程序设计课程混合式教学改革与实践, 将线上资源与线下课堂有机结合。嵩天和黄天羽(2017)提出了 Python 语言程序设计教学案例新思维, 强调以问题为导向设计教学案例。

在课程体系与思政建设方面^[12-15], 王哲等(2025)提出了融合人工智能的高校计算机专业课程体系化改革方案。杨娟等(2025)研究了数智化时代“大模型+”赋能的高校计算机类专业教学模式。李国和等(2025)开展了面向智能搜索的程序课程设计与教学实践。

这些研究对我们很有启发。所以, 我们想尝试把生成式 AI 用到这门课的教学, 看看能不能帮助学生更好地学习。

2 面向对象程序设计课程教学面临的问题

我们在教学过程中发现, 面向对象程序设计课程主要存在三个方面的问题。

第一, 理论概念太抽象, 学生学习有困难。类和对象是面向对象编程的基础, 但是很多学生一开始很难理解这两个概念。什么是类? 什么是对象? 类和对象之间是什么关系? 这些问题看起来很基础, 但是要真正理解并不容易。还有继承、多态、接口这些概念, 学生经常搞不清楚它们的作用和用法。有些学生能把代码写对, 但是问他为什么要这么写, 他就说不出来了。这说明学生只是记住了语法, 没有真正理解面向对象的思想。

第二, 学生实践能力不足, 遇到问题不知道怎么办。编程是一门需要动手的课程。学生在课堂上听懂了, 但是自己去写代码的时候还是经常出错。编译错误、运行时错误、逻辑错误, 各种问题都会出现。学生调试错误的能力比较弱, 有时候一个小问题要花很长时间才能解决。有些学生遇到困难就放弃了, 慢慢就对编程失去了信心。

第三, 教学模式比较单一, 学生参与度不高。现在的教学还是以老师讲课为主。老师在讲台上讲 PPT, 学生在下面听。讲完理论之后, 给学生布置一些练习题。这种模式很难激发学生的学习兴趣。有些学生上

课听讲, 下课就不碰代码了。等到期末考试的时候, 发现很多知识都忘记了。这样的教学效果不是很理想。

3 生成式 AI 赋能的教学改革方案

针对上面这些问题, 我们提出了一个生成式 AI 赋能的教学改革方案。这个方案把生成式 AI 工具(比如 DeepSeek、文心一言、通义千问等)引入到面向对象程序设计的教学中。我们用具体的案例来帮助学生理解抽象的概念, 用 AI 来辅助学生写代码和调试代码。

3.1 用 AI 解释面向对象的基本概念

在讲类和对象的时候, 我们会给学生一些生活中的例子。比如, 可以把“狗”看作是一个类, 把家里养的那只具体的狗看作是一个对象。但是有些学生还是不太好理解。这时候, 我们可以让学生用 AI 来帮忙。

【案例 1】理解类和对象的概念

教师给出提示词:“请用通俗的语言解释一下面向对象编程中类和对象的区别。请用生活中的例子来说明。”

AI 的回答是这样的(根据豆包的回答整理):

“类就像一个设计图纸。图纸上画了房子应该是什么样的, 有什么房间, 门和窗在什么位置。但是图纸本身不是房子。对象就是按照这个图纸盖出来的具体的房子。你可以根据同一个图纸盖出很多个不同的房子, 它们结构相似, 但是位置不同、颜色不同、里面住的人也不同。”

“在编程里也是这样。类是一个模板, 它定义了属性(比如名字、年龄)和方法(比如吃饭、睡觉)。对象是根据这个模板创建出来的具体的东西。每个对象都有自己的属性值, 但是它们都有相同的行为。”

学生看了这个解释之后, 对类和对象的关系就更清楚了。他们还可以继续问 AI 问题, 比如“怎么用 Java 写一个类和对象?” AI 会给出代码示例和解释。

3.2 用 AI 辅助理解继承和多态

继承和多态是面向对象编程中比较难理解的两个概念。我们设计了一个动物类的案例来帮助学生理解。

【案例 2】理解继承和多态

教师给出提示词:“我想让学生理解继承和多态。请设计一个动物类的例子, 有一个父类 Animal, 有两个子类 Dog 和 Cat。父类有一个方法叫 makeSound(), 子类重写这个方法。请用 Java 代码实现, 并解释代码的含义。”

AI 会生成这样的代码(根据 deepseek 的回答整理):

```
class Animal {
    String name;
    public Animal(String name) {
        this.name = name;
    }
    public void makeSound() {
        System.out.println("动物发出声音");
    }
}
class Dog extends Animal {
    public Dog(String name) {
        super(name);
    }
    @Override
    public void makeSound() {
        System.out.println(name + "说: 汪汪汪");
    }
}
class Cat extends Animal {
    public Cat(String name) {
        super(name);
    }
    @Override
    public void makeSound() {
        System.out.println(name + "说: 喵喵喵");
    }
}
public class Test {
    public static void main(String[] args) {
        Animal myDog = new Dog("小黑");
        Animal myCat = new Cat("小白");
        myDog.makeSound();
        myCat.makeSound();
    }
}
```

AI 接着解释: “在这个例子中, Animal 是父类, Dog 和 Cat 继承自 Animal。它们继承了父类的 name 属性, 同时重写了 makeSound() 方法。在测试代码中, 我们把子类对象赋值给父类类型的变量。运行时, Java 会根据实际的对象类型来调用对应的方法。这就是多态。”

学生可以运行这段代码, 观察输出结果。他们也可以修改代码, 添加新的动物类, 或者增加新的方法。通过动手实践, 学生对继承和多态的理解就更深入了。

3.3 用 AI 辅助调试代码

学生在写代码的时候经常会遇到错误。以前他们可能要去问同学, 或者等老师来解答。现在可以先用 AI 来帮忙。

【案例 3】调试代码错误

学生写了一段代码, 但是运行的时候报错了。他把错误信息复制给 AI:

“我有一个 Java 程序, 编译的时候报错 ‘NullPointerException’。这是我的代码:

```
public class Test {
    String name;
    public void printName() {
        System.out.println(name.length());
    }
    public static void main(String[] args) {
        Test t = new Test();
        t.printName();
    }
}
```

AI 分析后回答 (根据文心一言的回答整理):

“你的代码出现了 NullPointerException, 也就是空指针异常。这个错误的原因是, 你在 Test 类中定义了一个 String 类型的变量 name, 但是没有给它赋值。在 Java 中, 没有赋值的对象变量默认是 null。当你在 printName() 方法中调用 name.length() 时, name 是 null, 所以程序就报错了。”

“要解决这个问题, 你可以在创建对象之后给 name 赋值, 比如: Test t = new Test(); t.name = “Hello”; 或者你在类中给 name 一个默认值, 比如: String name = “”;

学生看了 AI 的解释, 很快就找到了问题所在, 也明白了空指针异常产生的原因。这样他不仅解决了当前的问题, 也学到了相关的知识。

3.4 用 AI 生成练习题和项目案例

教师备课的时候, 可以用 AI 来帮助生成练习题和项目案例。这样可以节省时间, 也可以得到更多样化的教学资源。

【案例 4】生成练习题

教师给出提示词: “请给我出 5 道关于 Java 继承的选择题, 每道题要有 4 个选项, 并给出正确答案和解析。”

AI 会生成这样的题目 (根据通义千问的回答整理):

“1. 在 Java 中, 关键字什么用于继承一个类?

A. implements B. extends C. inherit D. super

正确答案: B

解析: 在 Java 中, 使用 extends 关键字来实现类的继承。implements 用于实现接口。”

AI 还会继续给出其他 4 道题目和解析。教师可以选择合适的题目用在课堂练习或者课后作业中。

4 教学改革的实践效果

我们把这个教学改革方案应用到了 2024-2025 学年第一学期的《Java 面向对象程序设计》课程中。上课的学生是计算机科学与技术专业的大二学生, 一共 117 人。我们在课程中引入了生成式 AI 工具, 鼓励学生在学习过程中使用 AI 来帮助理解概念、调试代码、完成作业。

学期结束之后，我们做了一个问卷调查，了解学生对这种教学方式的想法。问卷通过问卷星发放，回收有效问卷 112 份。

调查结果如图所示。从图 1 中可以看出，83.5% 的学生对教师的授课方式表示满意。从图 2 中可以看出 90.4% 的学生对课程表示满意。

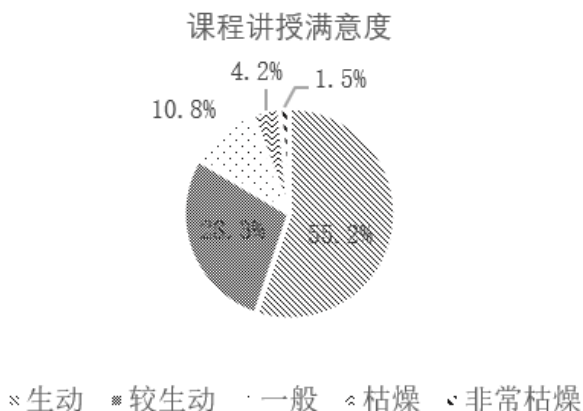


图 1 课程讲授满意度调查结果

学生也在问卷中写了一些自己的感受。有学生说：“以前学继承和多态的时候总是搞不清楚，这次用 AI 帮我解释，还生成了很多例子，我终于明白了。”还有学生说：“写代码遇到错误的时候，以前要等老师解答，现在问 AI 很快就能得到答案，学习效率提高了。”

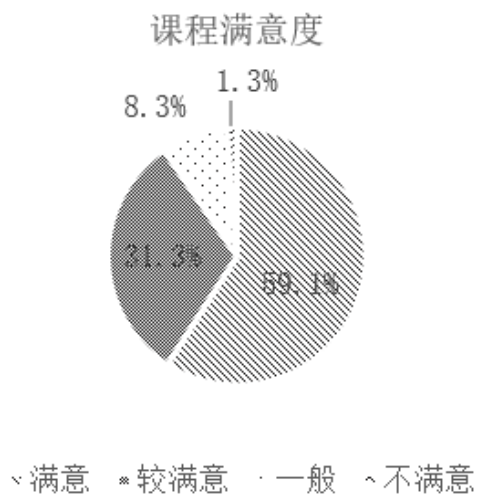


图 2 课程满意度调查结果

教师的感受也很好。以前答疑的时候，很多学生问的是相同的基础问题，老师要反复回答。现在学生先问 AI，解决不了的问题再来问老师，老师的答疑负担减轻了不少。老师可以把更多的时间用来准备教学内容，和学生进行更深入的交流。

5 讨论

从实践效果来看，生成式 AI 赋能面向对象程序设计教学有几个明显的优势。

第一，帮助学生更好地理解抽象概念。AI 可以用通俗的语言和生动的例子来解释复杂的概念。学生可以随时提问，得到即时回答。这种互动式的学习方式比单纯听老师讲课效果要好。

第二，提高学生调试代码的效率。学生遇到错误的时候，可以先用 AI 来诊断问题。AI 能够快速定位错误，并给出修改建议。学生不用花很多时间去查资料或者等老师解答，学习的连续性得到了保证。

第三，减轻教师的重复性工作负担。AI 承担了一部分答疑和出题的工作，教师可以更专注于教学设计和个性化指导。教师的角色从知识的传授者变成了学习的引导者。

第四，激发学生的学习兴趣。学生发现 AI 可以帮自己解决问题，学习起来更有信心。他们愿意花更多时间去写代码、去探索，形成了良性循环。当然，这种教学方式也面临一些挑战。

第一个挑战是 AI 可能给出错误答案。生成式 AI 有时候会产生“幻觉”，给出一些看似正确但实际有问题的代码或解释。如果学生不加辨别地接受，可能会学到错误的知识。我们在教学中反复提醒学生，AI 的回答只能作为参考，不能完全相信。要学会验证和判断。

第二个挑战是学生可能过度依赖 AI。有些学生遇到问题不自己思考，直接问 AI 要答案。这样下去，他们解决问题的能力反而会下降。我们在课堂上强调，要先自己思考，实在想不出来了再用 AI 辅助。AI 是工具，不能替代思考。

第三个挑战是技术本身还在快速发展。现在的 AI 工具功能很强大，但是过一段时间可能又有新的工具出现。教师需要不断学习和适应，才能用好这些工具。

6 结束语

本文探讨了生成式 AI 在面向对象程序设计课程教学中的应用。我们提出了一个 AI 赋能的教学改革方案，通过具体的案例展示了如何用 AI 帮助学生理解面向对象的核心概念、调试代码、完成练习。教学实践表明，这种教学方式能够提高学生的学习兴趣和学习效果，同时也能减轻教师的重复性工作负担。

未来的教学中，我们还会继续探索怎么更好地利用 AI。比如，可以尝试用 AI 来评估学生的代码质量，给出个性化的学习建议。也可以研究怎么把 AI 和项目式学习结合起来，让学生完成更复杂的编程项目。我们相信，随着技术的不断进步，AI 会在教育中发挥越来越重要的作用，帮助培养更多优秀的计算机人才。

参考文献

- [1] 中共中央国务院. (2025). 教育强国建设规划纲要 (2024—2035 年). 北京: 人民出版社.
- [2] 宋晓玲, 刘磊. (2025). 基于大模型的软件工程课程AI智能助教系统构建与实践[J]. 计算机技术与教育学报, 13(7), 51-56.
- [3] 康雁, 张璇, 杨云, 朱锐. (2025). 大模型驱动的软件工程实践课程改革与实践[J]. 计算机技术与教育学报, 13(7), 57-62.
- [4] 余红宴. (2025). 生成式AI赋能Python编程数据分析教学改革与实践[J]. 计算机技术与教育学报, 13(7), 78-86.
- [5] 陈倩, 陈婷婷. (2025). 基于国产大模型线上线下辅助程序设计基础课程探索[J]. 计算机技术与教育学报, 13(6), 83-87.
- [6] 高竹, 马文波, 彭程, 刘玮, 孙平平. (2025). 数智赋能《大数据与云计算》课程教学创新与实践[J]. 计算机技术与教育学报, 13(6), 40-48.
- [7] 王建, 周红, 赵慧玲, 聂芳. (2025). 基于大模型和知识图谱的个性化教学新模式[J]. 计算机技术与教育学报, 13(7), 93-99.
- [8] 罗雪梅, 万波, 王苗等. (2025). 程序设计课程混合式教学改革与实践[J]. 计算机技术与教育学报, 13(2), 158-162.
- [9] 嵩天, 黄天羽. (2017). Python语言程序设计教学案例新思维[J]. 计算机教育, (12), 11-14+19.
- [10] 杨志卓. (2025). 基于OBE的程序设计教学及计算思维能力培养[J]. 计算机技术与教育学报, 13(2), 169-173.
- [11] 刘海波, 秦秋滢, 程雁, 张伟. (2025). 基于AIGC与BOPPPS模式的中职《C语言》课程教学路径探索[J]. 计算机技术与教育学报, 13(6), 93-102.
- [12] 王哲, 梁新宇, 张桂芬, 葛丽娜. (2025). 融合人工智能的高校计算机专业课程体系化改革[J]. 计算机技术与教育学报, 13(6), 68-72.
- [13] 杨娟, 贺禹鑫, 韩光川. (2025). 数智化时代“大模型+”赋能的高校计算机类专业教学模式研究[J]. 计算机技术与教育学报, 13(6), 78-82.
- [14] 李国和, 张扬武, 董丹丹. (2025). 面向智能搜索的程序课程设计建设与教学实践[J]. 计算机技术与教育学报, 13(7), 47-50.
- [15] 乔文豹, 于欣宜. (2025). 突破AI代写: 计算机项目实践课程教学改革与思政建设[J]. 计算机技术与教育学报, 13(6), 9-17.