

# 基于OBE理念结合思政的Java程序设计教学模式探讨

乔丽红\*, 纪良浩, 舒禹程, 唐晓军, 聂永萍

重庆邮电大学计算机科学与技术学院, 重庆

收稿日期: 2023年10月6日; 录用日期: 2023年11月7日; 发布日期: 2023年11月14日

## 摘要

在基于OBE教育理念的背景下, 积极推进课程思政建设, 构建适应新时代本科“三全育人”教育模式。本文综合运用OBE教育理念, 并以课程思政的观点审视Java程序设计课程中的最新要素, 深入探讨创新的课程思政融入方式。首先, 详述了OBE理念, 随后梳理并精炼了课程思政的关键要点。接下来, 通过深度分析Java程序设计中的抽象性以及其与思政相关的要点, 提供了全面的视角。最后, 阐述了在课程思政教学实践中采用的教学方法。

## 关键词

OBE教育理念, Java, 思政, 模式探讨

# Discussion on the Teaching Mode of Java Programming Based on OBE Concept Combined with Ideology and Politics

Lihong Qiao\*, Lianghao Ji, Yucheng Shu, Xiaojun Tang, Yongping Nie

College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing

Received: Oct. 6<sup>th</sup>, 2023; accepted: Nov. 7<sup>th</sup>, 2023; published: Nov. 14<sup>th</sup>, 2023

## Abstract

In the context of OBE education concept, we actively promote the construction of course ideology

\*第一作者。

文章引用: 乔丽红, 纪良浩, 舒禹程, 唐晓军, 聂永萍. 基于OBE理念结合思政的Java程序设计教学模式探讨[J]. 教育进展, 2023, 13(11): 8653-8661. DOI: 10.12677/ae.2023.13111336

and politics, and build an education model that adapts to a new era of undergraduate “three-pronged education”. This paper comprehensively applies the OBE education concept and examines the latest elements in the Java programming course from the viewpoint of curriculum ideology, and discusses innovative ways of integrating curriculum ideology. Firstly, the concept of OBE is described in detail, and then the key points of curriculum ideology are sorted out and refined. Next, a comprehensive perspective is provided through an in-depth analysis of the abstract nature of Java programming and the key points related to course ideology and politics. Finally, the pedagogical approach adopted in the teaching practice of course ideology and politics is elaborated.

## Keywords

OBE Education Concept, Java, Ideology and Politics, Model Discussion

Copyright © 2023 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/>



## 1. 研究背景

Java 语言一直焕发着新生命力，它被誉为程序设计领域的不朽之树。自 1995 年首次发布以来，Java 经历了多次版本迭代和功能扩展，一直保持着强大的生命力。它广泛应用于几乎所有的程序设计领域，并始终稳居 IEEE Spectrum 编程语言榜单的前列[1]。因此，Java 已成为几乎所有计算机专业学生的必修课程，Java 程序设计学习者的数量也不断增加。在这个背景下，已经存在大量丰富多样的 Java 程序设计课程资源，涵盖了从基础知识点、题库到项目实战等各个方面的内容。可以说，仅仅从 Java 语言本身的知识点出发，在面向本科学生的教学中，如果不采用不同的教学方法和角度，很难有新的教学思路。在 2016 年的全国高校思想政治工作会议上，习总书记明确指出，“要坚持把立德树人作为中心环节，把思想政治工作贯穿教育教学全过程，实现全程育人”。课程思政工作是落实总书记该讲话精神的重要抓手[2] [3]。在 2020 年 5 月，教育部下发了《高等学校课程思政建设指导纲要》，明确指出了各学科课程思政建设的目标要求和内容重点，提出要科学设计课程思政学体系，结合专业特点分类推进课程思政建设。更进一步地明确了课程思政在当前高等学校教育教学体系中的重要性，并提供了指导性的纲要。在这样的契机下，众多教育工作者从课程思政的角度，针对 Java 相关课程的教学工作展开了教改研究[2] [3] [4] [5] [6]。OBE 教育理念应用于课程教学，颠覆了“教为主、学为辅”的传统教学模式，倡导教育形式由“内容驱动”向“成果驱动”转型，以学生完成该课程时应取得的学习成果为起点和出发点，采用反向设计原则，根据毕业要求，开设对应课程(形成任务)，确定课程目标、组织教学内容、实施教学过程、评估学习成效，并持续改进课程教学质量，实现“学生为中心”的教学模式[4]。

《Java 程序设计》是计算机类专业的重要必须基础课程。通过课程的学习，使学生能够掌握面向对象程序设计的基本理论与方法，掌握实用的面向对象编程语言的技能，为学生应用面向对象程序设计理论解决实际问题奠定了良好基础。学生如何学好这门课，教师如何教好这门课，如何将先进的思政理念和 OBE 理念与课程进行有机结合，实现知识传授与思想教育的紧密结合，实现教师所教与实际应用密切相关是任课教师亟须思考的问题。

目前，针对《JAVA 程序设计》课程存在以下问题：学生学习兴趣不强烈，缺乏明确学习目标：目前许多学生容易受到电子游戏等诱惑，导致他们对学习的兴趣不高，同时缺乏明确的学习目标和积极的学习动力。在大学学习的过程中，学生需要逐渐培养正确的人生观、世界观和价值观。在这一点上，教

师在教学中扮演了引导的重要角色，帮助学生认识到学习的重要性和明确的学习目标[5] [6]。

教学模式和教学手段不适应社会需求：传统的教学模式通常以内容为导向，偏重于传授知识点，而较少关注学生的实际学习效果。然而，现代社会对于学生的需求更强调综合能力的培养，包括解决问题的能力 and 团队协作能力等。因此，教师需要采用更为先进的教学理念和方法，以使课程更加贴近社会的实际需求。对知识点的掌握优先于预期学习成果：传统的教学往往过于强调学生对知识点的掌握，而忽视了学生应该实现的学习成果。现代教育更加注重学生的综合素质和能力培养，因此，教学应该更加关注学生是否能够运用所学知识来解决实际问题。

学生缺乏正确的学习引导：许多学生通常不清楚他们的学习应该具体达到什么样的目标，而教师也未能提供足够的引导。这导致学生对《JAVA 程序设计》课程的工程性、应用性和实践性特点理解不够深入，因此学习兴趣不高。在这方面，教师需要更加积极地引导学生，使他们能够更主动地参与学习过程。

教学以教师为中心：传统的教学模式通常以教师为中心，教学内容、进度、形式和方法都是由教师决定的，这未能充分突出学生的主体地位和个体特点。然而，现代教育理念更加强调以学生为中心，以学生的学习过程为导向，因此，教师需要更加注重学生的需求，采用分层次教学和因材施教的方法。

## 2. Java 程序设计教学中思政要点的挖掘

思政要点是课程思政工作具体实施的出发点，其源于课程的相关专业知识点但又高于知识点，是“课程”和“思政”这两个要素的交汇之处。在之前的工作中，我们将信息安全导论的课程思政要点分为 5 类，分别是道德法律类、哲学思维类、技术发展类、国家政策类以及民族自信类[7]。在将此经验向 Java 程序设计课程拓展推广后，共计提炼出 26 个思政要点，见表 1。

**Table 1.** Summary of ideological points of Java programming course

**表 1.** Java 程序设计课程思想要点汇总

序号	思政要点	道德法律	哲学思维	技术发展	国家政策	民族自信	所在章节	序号	思政要点	道德法律	哲学思维	技术发展	国家政策	民族自信	所在章节
1	编程语言排行榜			1	1		1.1	14	大学的学习方法		1				4.1
2	人与机器交互过程分层抽象		1				1.2	15	参数传递 数据输入输出抽象		1	1			4.4
3	跨平台和运行速度之间的平衡中庸		1			1	1.3	16	现实生活中继承的概念和 Java 中继承		1	1			5.1
4	信息的存储方式我国的快速发展			1	1		2.1	17	父类和子类 无：无限的可能		1			1	5.3
5	变量和内存内容物和外形式		1				2.1	18	接口：面向对象的抽象到面向过程的抽象		1	1			6.1
6	基础类型类框架和实现的中庸		1				2.1	19	正则表达式 化繁为简		1				7.1
7	简单变量外部画面内在和外		1				2.2	20	从外部类到内部类 从有名类到匿名类		1	1			7.2
8	把大象放进冰箱里顺序流程		1				3.1	21	异常处理和断言 允许出错		1				7.3

## Continued

9	分支流程 机器智能人工智能	1	1	3.1	22	用户接口 内在和外在	1	1	8.1
10	循环中流程 愚公移山精神	1		1	3.1	23	层层抽象的输入输出 分层和抽象	1	9.2
11	循环中的跳出标签 框架和实现的中庸	1			3.2	24	流式读写和随机读写 OOP 和 OPP 的中庸	1	9.3
12	循环和分支的启示 天才 = 汗水 + 灵感	1			3.3	25	线程工作的抽象 深耦合和浅耦合	1	10.2
13	类和对象 抽象具体共性个性	1			4.1	26	TCP 的套接字编程 数据收发抽象	1	11.2

### 3. Java 思政点新思考

对表 1 进行初步分析后发现, 所提炼出来的 26 个思政要点基本上都属于哲学思维类。之所以出现这种情况, 是由于本文凝练课程思政点的方向是向内探究的, 而 Java 语言本身就是以抽象思维、逻辑智慧为出发点编写的, 当抛开一些表面的技术细节(如各种语法规则、后台实现)之后, 其所呈现的必然是哲学性的。就表 1 的内容, 又可以从抽象和中庸两个观点出发进行进一步分析。

#### 3.1. Java 编程中的抽象思想和举例

在 Java 课程中融入思想政治教育的案例可以通过强调一些价值观念、社会责任感以及伦理原则来实现。下面是一个详细的 Java 课程思政案例:

以下是一个简单的 Java 编程示例, 说明了如何使用接口来解决上述提到的问题, 以避免刻板地按照“父生子、子生孙”的思想从 Object 类依次派生导致的抽象冗余。这个示例模拟了生活中的马和牛拉车的情景, 强调了方法的抽象和接口的使用。

首先, 我们定义一个名为 Vehicle 的接口, 该接口包含一个名为 pull 的抽象方法, 表示车辆可以拉的行为:

```
java
Copy code
interface Vehicle {
    void pull();
}
```

然后, 我们创建两个类, Horse(白马)和 Cow(牛), 它们都实现了 Vehicle 接口, 并提供了 pull 方法的具体实现:

```
java
Copy code
class Horse implements Vehicle {
    @Override
    public void pull() {
        System.out.println("白马正在拉车");
    }
}
```

```
}  
class Cow implements Vehicle {  
    @Override  
    public void pull() {  
        System.out.println("牛正在拉车");  
    }  
}
```

接下来，我们创建一个名为 `Cart` 的类，该类表示车辆，但不关心具体是马还是牛在拉车。`Cart` 类接受任何实现了 `Vehicle` 接口的对象，并调用它们的 `pull` 方法：

```
java  
Copy code  
class Cart {  
    public void attachVehicle(Vehicle vehicle) {  
        vehicle.pull();  
    }  
}
```

最后，我们可以在主程序中创建 `Horse`、`Cow` 和 `Cart` 对象，并演示它们的使用：

```
java  
Copy code  
public class Main {  
    public static void main(String[] args) {  
        Horse whiteHorse = new Horse();  
        Cow cow = new Cow();  
        Cart cart = new Cart();  
        cart.attachVehicle(whiteHorse);  
        cart.attachVehicle(cow);  
    }  
}
```

这个示例中，我们使用接口 `Vehicle` 来表示车辆的拉行为，然后 `Horse` 和 `Cow` 类都实现了这个接口，表明它们都可以拉车。`Cart` 类则接受任何实现了 `Vehicle` 接口的对象，这样我们可以轻松地将不同的车辆连接到车上，而不关心具体是什么动物在拉车。这种方式避免了刻板的继承关系，更加关注对象的行为和方法，符合面向对象的设计原则。

抽象是人类进行观察、研究以及实践的重要工具，是计算机程序设计的核心思想之一。面对纷繁的外部世界，只有提炼出研究对象的核心特征，把握事物的主要矛盾，才能更加准确、直接地实现自身的应用需求。Java 语言在发布时，为了便于快速应用，不光提供了基本的语法框架，更是提供了一系列功能强大的应用库，这些库的层级结构设计，是 Java 语言对不同类型应用需求的高度总结和抽象，对其中某些案例的学习，

不光是对 Java 语言编程应用技术的学习，更是自身方法论和世界观的构建过程。

老子的思想与 Java 中的类层次结构和对象方法确实有一定的相似之处。在 Java 中，所有的类都直接

或间接地继承自 `java.lang.Object` 类，这就符合了老子所说的“道生一，一生二，二生三，三生万物”的思想，即所有的类都源于 `Object` 类，是对外部世界中事物的一个统一抽象。

同时，老子提到“有名，万物之母”，这可以与 Java 中的 `toString` 和 `hashCode` 方法联系起来。在 Java 中，`toString` 方法用于返回一个对象的字符串表示，以便描述对象本身。而 `hashCode` 方法用于返回一个唯一的标识符，以便标识对象。这两个方法可以被认为是对象的“名字”，它们有助于将对象与外部世界关联起来。

此外，老子提到事物之间的相互对比，这可以与 Java 中的 `equals` 方法联系起来。在 Java 中，`equals` 方法用于比较两个对象是否相等，这是在编程中常常需要的一个操作。这也符合老子所说的事物之间的相互对比的概念。

总之，虽然老子的思想是哲学性的，而 Java 是一种编程语言，但可以看到在 Java 的设计中，一些概念和思想与古代哲学有一定的共通性，这些思想有助于理解和解释 Java 编程中的一些概念和原则。

当谈到不同的方法时，需要了解每种方法中所蕴含的哲学思想以及它们的背后原则。以下是一些常见的编程方法和相关的哲学思想：

**面向对象编程(OOP):**

哲学思想：OOP 基于物体和其相互作用的思想。它将问题分解为对象，每个对象都有自己的属性和方法。OOP 强调了封装、继承和多态的概念，以模拟现实世界的关系和行为。

原则：封装性(Encapsulation)、继承性(Inheritance)、多态性(Polymorphism)。

**函数式编程(FP):**

哲学思想：FP 基于数学中的  $\lambda$  演算，强调将计算视为函数应用的过程。它避免了可变状态，强调不可变数据和无副作用的函数。

原则：不可变性(Immutability)、高阶函数(Higher-order functions)、纯函数(Pure functions)。

**面向切面编程(AOP):**

哲学思想：AOP 强调了横切关注点(cross-cutting concerns)的分离，这些关注点通常横跨应用的不同部分，如日志记录、事务管理等。

原则：分离关注点(Separation of Concerns)、切面(Aspects)、连接点(Join points)。

**敏捷开发(Agile Development):**

哲学思想：敏捷开发强调合作、响应变化、快速迭代和交付可工作的软件。它鼓励开发团队和客户之间的沟通和协作。

原则：个体和互动胜过流程和工具、可工作的软件胜过详尽的文档、客户合作胜过合同谈判、响应变化胜过遵循计划。

**测试驱动开发(TDD):**

哲学思想：TDD 强调在编写实际代码之前编写测试用例。这鼓励开发者思考如何测试代码，并确保每一行代码都经过了测试。

原则：先写测试(Write tests first)、仅编写足够使测试通过的代码(Write just enough code to make the tests pass)、持续重构(Continuous refactoring)。

**领域驱动设计(DDD):**

哲学思想：DDD 强调将业务领域的知识和模型融入软件设计中。它将复杂的业务问题分解成领域对象和领域服务。

原则：领域模型(Domain model)、界限上下文(Bounded context)、领域驱动设计语言(DDD language)。

极限编程(XP):

哲学思想: XP 强调小团队、快速反馈、简化和自组织。它鼓励程序员与客户紧密合作,以实现高质量的软件。

### 3.2. 平衡和谐的 Java

Java 语言在设计上追求平衡和谐的中庸之道,旨在平衡各种因素,以满足不同的编程需求。Java 源自于 C++, 最初的目标是在保留 C++的强大功能的基础上,消除一些可能引发安全问题的机制,比如指针。这样,Java 成为了一个更加轻便实用的开发平台。从一开始,Java 注重功能和安全性、效率和稳定性之间的平衡,力求实现所有需求目标的平衡和谐的中庸之道。

平衡功能和安全、效率和稳定: Java 语言的设计追求中庸之道,力图在保持强大功能的同时确保安全性、效率和稳定性。例如,Java 引入了垃圾回收机制以提高程序的稳定性,同时通过字节码和 JVM 的混合式编译运行方式提高了执行效率。这个平衡在以下示例中得以体现:

```
java
Copy code
public class Main {
    public static void main(String[] args) {
        // Java 的垃圾回收机制确保程序的稳定性
        // 同时,通过 JIT 编译提高执行效率
        System.out.println("Hello, World!");
    }
}
```

面向对象的抽象和面向过程的灵活兼容: Java 语言采用了面向对象的思想,但也保留了一些面向过程的特性,以保持灵活性和兼容性。例如,Java 中的基本数据类型是面向对象的,但也提供了关键字(例如 byte)以保留面向过程编程的特性。以下示例展示了 Java 中的面向对象和面向过程的兼容性:

```
java
Copy code
// 面向对象: 整数的封装
Integer num = new Integer(42);
// 面向过程: 使用基本数据类型
int x = 10;
```

抽象和实际实现之间的平衡: Java 在基本数据类型的抽象上取得了平衡。它引入了面向对象的思想,但也保留了基本数据类型以提高执行效率。以下示例说明了 Java 在这方面的中庸之道:

```
java
Copy code
// 面向对象: 整型有对应的 Integer 类
Integer num = new Integer(42);
// 面向过程: 使用基本数据类型
int x = 10;
```

在实际功能实现中的平衡: Java 提供了丰富的库和框架,以满足不同类型的编程需求。它在不同领域的功能实现中寻求平衡,例如文件读写、网络通信和语法框架设计。以下示例说明了 Java 在文件读写

方面的中庸之道:

```
java
Copy code
// 文件读写: 提供了丰富的流式读写模型
FileInputStream fileInput = new FileInputStream("input.txt");
FileOutputStream fileOutput = new FileOutputStream("output.txt");
// 同时保留了基本的面向过程风格
byte[] buffer = new byte[1024];
int bytesRead;
while ((bytesRead = fileInput.read(buffer)) != -1) {
    fileOutput.write(buffer, 0, bytesRead);
}
```

总的来说, Java 编程语言的设计哲学强调平衡和中庸, 以满足不同需求和编程风格。这种平衡体现在语言的设计、功能实现和库的使用中, 使 Java 成为一种广泛应用于多个领域的强大编程语言。

以上表明, Java 作为一门杰出的编程语言, 其最为核心的吸引力在于对世界的卓越抽象描述, 以及对编写简洁高效代码的不懈追求。现代人才培养的要求强调全面发展, 包括道德、智力、体质、美感和劳动等各个方面。Java 作为一种具体的编程工具语言, 代表了智力的体现。然而, 当我们深入研究 Java 并将其扩展到不同领域时, 我们可以发现一系列思想和价值观, 这些可以成为课程思政的内容。这是道德内化的一部分。此外, 当我们欣赏 Java 中清晰的层次结构、简洁的语法逻辑以及其中蕴含的人类智慧时, 我们也在感受美的享受。因此, Java 不仅是一门程序设计语言, 还是一种涵盖多个领域的思想和价值的表现。

#### 4. 结语

计算机程序设计语言是人类逻辑思维的高度概括, 代表了形式化哲学智慧的核心, 而 Java 语言则在众多编程语言中脱颖而出。为了更好地开展 Java 程序设计课程的教学, 本论文从哲学思维的角度深入挖掘了 Java 语言中所蕴含的课程思政要点, 强调了其中的抽象性和中庸性。经过连续两轮的教学实践验证, 我们得出以下结论: 经过连续两轮的教学实践验证, 我们得出以下结论: 首先, Java 语言的抽象性是其最显著的特点之一。它以对象为中心, 通过类和接口的层次结构, 将世界的复杂性进行了高度抽象。这种抽象性在教学中为学生提供了深入理解计算机科学和编程思维的机会。学生通过学习 Java, 不仅仅是学习一门编程语言, 更是培养了抽象思维的能力, 能够将现实世界的问题抽象成计算机程序, 这是一种非常宝贵的思维方式。其次, Java 语言的平衡性表现在它在功能和安全、效率和稳定之间取得了平衡。这种平衡使得 Java 语言在广泛的应用领域都有出色的表现, 既能够处理高性能的计算任务, 也能够编写稳定可靠的企业级应用。这种中庸性在教学中也具有重要意义, 因为它教导学生在编程过程中不仅要追求功能的完善, 还要考虑代码的可维护性和可扩展性, 以及程序的安全性。综上所述, Java 语言在计算机程序设计课程中不仅仅是一门工具, 更是一门哲学, 一门思政的媒介。通过深入理解其抽象性和中庸性, 学生可以培养出全面发展的能力, 不仅在编程领域表现出色, 还能够在人生的各个方面取得成功。因此, 我们应该充分挖掘和传播 Java 语言中的哲学思想, 为学生的综合素质教育提供更多的可能性。

#### 基金项目

重庆邮电大学教育教学改革项目 XJG21218, 重庆邮电大学课程思政项目 XKCSZ230; 重庆邮电大学教育教学改革项目 XJG19230。

---

## 参考文献

- [1] 罗如为. Java Web 开发技术与项目实战[M]. 北京: 水利水电出版社, 2019.
- [2] 姜大志, 熊智, 杜支强. 计算机类专业课程思政实施方略研究[J]. 计算机教育, 2021(3): 85-89, 94.
- [3] 刘毅. Java 高级编程课程思政案例教学初探[J]. 计算机时代, 2020(12): 85-87, 90.
- [4] 郭世仁, 王俊红, 连剑波. 教学矩阵驱动的面向对象程序设计课程思政[J]. 计算机教育, 2020(9): 64-67.
- [5] 王中华. 程序设计“课程思政”建设的要点探析[J]. 教育现代化, 2020, 7(14): 137-139.
- [6] 张静, 倪琳妍. 高职专业课程融入思想政治教育的路径探索——以《Java 程序设计基础》课程为例[J]. 课程教育研究, 2020(5): 61+63.
- [7] 李振江, 王亚婷, 徐世鹏, 等. 信息安全导论课程思政教学探讨[J]. 高教研究, 2021, 37(2): 64-68, 78.