

贯通式教学驱动的面向对象课程 跨平台能力培养研究^{*}

杨扬 汪花梅 谢佳 房敏

哈尔滨工业大学（深圳）实验与创新实践教育中心（分析测试中心），深圳 518055

摘要 随着新工科建设的持续推进，跨平台开发能力已成为软件工程人才的核心素养。针对高校实践教学缺乏系统化跨平台开发训练的现状，本文提出了基于贯通式教学的课程改革方案。导论课程通过面向对象原则与模块化设计夯实编程基础，实践课程通过分层重构、适配开发与功能拓展实现跨平台迭代，形成覆盖“基础模块构建-系统集成-跨平台迭代”的递进式培养路径。教学实践表明，该模式通过模块化设计、系统集成与跨平台迁移的工程化训练，显著提升了学生的系统设计能力与工程素养，为工程教育中的系统能力培养提供了可复制的教学范式。

关键字 贯通式教学；实验教学；面向对象；项目式教学；跨平台开发

Integrated Teaching-Driven Approach to Cross-Platform Skill Development in Object-Oriented Programming Courses

Yang Yang Wang Huamei Xie Jia Fang Min

Education Center of Experiments and Innovations Analysis and Testing Center
Harbin Institute of Technology, Shenzhen
Shenzhen 518055

Abstract—With the continuous advancement of the New Engineering Education initiative, cross-platform development skills have become a core competency for software engineering professionals. In response to the current lack of systematic cross-platform development training in higher education practice teaching, this paper proposes a curriculum reform plan based on an integrated teaching approach. The introductory course strengthens programming foundations through object-oriented principles and modular design, while the practical course achieves cross-platform iteration through layered refactoring, adaptive development, and functional expansion, forming a progressive training path that covers "basic module construction—system integration—cross-platform iteration." Teaching practice demonstrates that this model, through engineering training in modular design, system integration, and cross-platform migration, significantly enhances students' system design capabilities and engineering literacy, providing a replicable pedagogical paradigm for cultivating system competencies in engineering education.

Keywords—Integrated teaching; Experimental teaching; Object-oriented; Project-based learning; Cross-platform development

1 引言

随着新工科建设的持续推进，多学科交叉融合与工程创新能力培养已成为高等教育改革的核心目标。新一代信息技术的发展催生了跨平台开发技术的广泛应用，从移动终端到物联网设备，从云端协同到边缘计算，跨平台开发能力日益成为工程师应对复杂系统设计与异构环境适配的必备素养。然而，当前高校计算机专业课程体系面临“技能供给侧”与“产业需求侧”的结构性矛盾，一方面，产业界对具备跨平台架构设计与多端协同开发能力的复合型人才需求激增，另一方面，传统课程仍固守单一语言语法教学模式，导致人才培养与产业需求出现结构性错配。

教育部新工科项目已经通过多轮试点验证了课程体系贯通的必要性^[1]，在此背景下，本研究提出基于贯通式教学面向对象课程改革方案。通过构建“基础模块构建-系统集成-跨平台迭代”的培养路径，以跨平台代码迁移与多端功能拓展为实践主线，探索如何将分散的知识点转化为系统性工程能力，为破解高校跨平台开发实践教学困境提供切实可行的范式参考。

2 面向对象课程现状分析

目前国内高校在开设面向对象课程时通常以JAVA语言或C++语言为载体，教学内容普遍聚焦于基础语法特性的讲解。在设计实验内容时，多采用Windows操作系统作为实验平台，学生通过完成体量较小、相互独立的实验任务掌握知识点，导致学生缺少综合性软件项目开发体验。同时，这种单一的实验

* * 通讯作者：杨扬 yangyang1986@hit.edu.cn

环境限制了学生对跨平台开发共性问题的系统性认识,不利于帮助学生构建从“模块设计”到“系统集成”再到“多端迭代”的能力体系^[2],难以应对工业界对跨平台架构设计、多场景适配及工程化迭代的复合型需求。

部分高校已经意识到这些问题,尝试引入移动开发(如Android、iOS、HarmonyOS)或跨平台框架(如React Native、Flutter)相关课程。然而,这些改革往往陷入“工具主义”的误区,教学设计多局限于特定技术工具的应用,缺乏对跨平台开发方法论的深度总结与提炼。例如,学生能够使用HarmonyOS的API,但缺乏设计跨平台适配层的能力。这种教学方式导致课程内容与企业实际技术栈脱节,能力培养与系统化工程思维脱节。

综上所述,当前面向对象课程在教学内容、实践体系和改革尝试三个维度均存在局限性,这些问题严重制约了学生跨平台开发能力和系统工程思维的培养,急需通过系统化的课程重构予以解决。需要重新设计教学内容、创新教学方法、重构实践体系,从语法导向转向工程思维培养,从单一平台实验转向多平台验证,从工具使用训练转向方法论提炼。

3 面向对象系列课程的贯通式教学设计

3.1 方法论

跨平台开发作为当前软件工程实践中的重要方向,要求开发者能够在不同平台之间实现功能一致性、性能优化与用户体验统一。现有工程领域普遍应用的最佳实践集合,主要包括统一抽象与平台解耦、最大化代码复用以及渐进式开发等原则。其中,统一抽象通过引入适配层隔离平台差异,提升系统的可移植性;最大化复用强调共享核心逻辑,降低重复开发成本;渐进式开发遵循“共性优先、差异延后”的策略,确保系统在多平台间平滑演进。

基于上述工程方法论,本研究将贯通式教学理念融入面向对象系列课程设计,以新工科“系统能力培养”为导向^[3-4],围绕“基础模块构建-系统集成-跨平台迭代”三大环节,构建递进式能力培养路径。在教学过程整合最佳实践,培养学生具备完整的软件构造与迁移能力。

3.2 课程体系构建

系列课程由两门紧密衔接的课程组成,分别对应不同阶段的能力目标:

(1) 面向对象的软件构造导论(导论)

本课程有40学时的理论教学和16学时的课内实

验,聚焦“基础模块构建-系统集成”阶段。在此之前,学生更多接触的是体量较小的、相对独立的编程实验,缺乏系统性设计的意识。针对这一现状,课程在初期系统讲授Java语言基础,涵盖基本语法、类与对象、继承与多态、接口、异常处理等内容,帮助学生建立起面向对象程序设计的基本能力。接下来,课程引入常用的软件设计原则,指导学生理解模块划分、接口定义、代码复用与扩展性优化的基本方法,为后续的模块化设计与系统集成奠定理论基础。在学生掌握基础的评判标准后,课程系统讲解六种经典的设计模式,帮助学生理解每种模式的应用场景和结构特点。

课程实践以Windows平台下的“飞机大战”单机游戏开发为载体,要求学生实现游戏的基本功能模块(敌机生成、道具掉落、弹道切换等),同时通过需求变更,促使学生进行代码重构和架构优化,让学生体会设计模式的必要性和优越性^[5]。

(2) 面向对象的软件构造实践(实践)

承接导论课程,聚焦“系统集成-跨平台迭代”阶段。课程以项目式任务驱动,通过真实场景的复杂需求,将离散知识点整合为系统性能力^[6-7]。课程要求学生在24学时内将导论课程中的Windows平台代码迁移至Android平台,并通过新增网络通信、移动端UI适配等任务,直面跨平台开发中的共性挑战。

课程通过“目标-内容-任务”的协同设计,以迁移目标驱动内容组织,以平台差异点细化教学模块,以递进式迁移任务锻炼应用能力,帮助学生构建从桌面端到移动端、从单机到联网的完整能力链条。

3.3 教学目标

本系列课程以系统能力培养为导向,明确设定以下教学目标:

- ① 模块化建模能力:掌握面向对象设计原则与模块化实现方法,能够独立完成中小型软件系统的模块划分与接口设计;
- ② 系统集成与演进能力:理解软件系统从单模块到多模块集成、从静态设计到动态演化的全过程,具备根据需求变化进行系统优化与重构的能力;
- ③ 跨平台适配与迁移能力:熟悉主要平台(Windows、Android)在资源管理、界面布局与异步通信机制的差异,掌握基本的跨平台迁移与功能拓展方法;
- ④ 工程化实践素养:在项目开发过程中应用版本控制、模块测试、性能优化等工程技术,具备初步的软件工程项目意识。

4 教学实施过程

表 1 从教学目标、教学内容与实践任务三个维度。系统呈现两门课程的协同设计与实施路径。导论课程

表 1 导论和实践的教学目标、教学内容与实践任务

| 课程 | 教学目标 | 教学内容 | 实践任务 |
|----|--------------------|--|---|
| 导论 | 模块化建模能力 系统集成能力 | Java语言基础 面向对象设计原则 设计模式 | 单例模式: 英雄机 工厂模式: 敌机/道具 策略模式: 弹道切换 观察者模式: 炸弹 数据访问对象模式: 排行榜 模板模式: 游戏难度 Swing界面开发 |
| 实践 | 跨平台开发能力 工程化实践素养 | Android平台架构 应用组件与生命周期 资源管理 UI布局设计 音频播放 数据存储 异步通信 网络编程 | 游戏画面渲染迁移 文件存储适配 排行榜重构 异步通信重构 音频播放重构 网络功能开发 |

4.1 差异化对比构建教学内容

实践课程的首要任务是帮助学生理解 Android 平台的核心架构与技术特性，针对性解析 Android 与 Windows 平台的关键差异点，为后续跨平台迁移奠定基础。因此课程内容将围绕以下几部分展开。

首先，了解 Android 平台架构对理解 Android 应用的运行机制至关重要。与 Windows 桌面应用只需关注窗体的加载和关闭事件不同，Android 中的各类组件（如 Activity, Service 等）都拥有独立和复杂的生命周期，管理不当可能导致内存泄露或应用卡顿，因此课程专门讲解 Android 应用组件的生命周期管理，旨在提升学生的资源管理能力。UI 布局方面，Windows 常用固定像素布局，而 Android 需应对多种屏幕尺寸和密度，课程引入了响应式设计，帮助学生高效完成多屏适配。异步处理与网络编程方面，课程针对 Android 单线程模型的限制，系统介绍了 Handler 等异步通信方案，以及 HTTP 和 Socket 两种网络通信技术，帮助学生解决 Android 平台中的 UI 线程限制和网络请求问题，提高应用的响应性和稳定性。在资源管理、音频播放、数据存储等内容模块中，课程也通过与 Windows 技术栈的对比，强调了 Android 在多分辨率资源适配、本地持久化策略及高效音频处理等方面的优势。

已在前期研究中系统阐述（见参考文献[5]），本文聚焦实践课程的创新设计，重点解析如何通过 Android 平台特性解析、分层重构策略，将导论课程形成的设计能力转化为跨平台工程实践素养。

表 1 导论和实践的教学目标、教学内容与实践任务

4.2 模块化分层优化实践任务

在实践课程中，学生需以导论课程完成的 Windows 平台代码为基础，开展系统化分层重构与迁移实践，并完成网络功能的开发与拓展。课程采用开放式的任务设计模式^[8]，打破了传统实验中固定流程与单一解决方案的局限性，强调学生主动探索与创新。具体可分为以下三个阶段：

(1) 核心层抽取

指导学生识别原 Windows 平台代码中与平台强耦合的模块（如 Swing 界面渲染、Windows 文件存储 API 等），提取与平台无关的核心业务逻辑（如敌机生成、碰撞检测、道具效果），以提高代码复用性。重构过程中，要求学生严格遵循单一职责原则（SRP）和接口隔离原则（ISP），为后续多平台扩展奠定基础。教学中安排代码走查环节，保证核心模块抽取的完整性与合理性。

(2) 适配层开发

课程设计五个递进式实验任务，每个任务聚焦一个关键适配挑战，逐步强化学生的平台适配能力与系统整合能力，具体的实验设计见表 2。

为每个实验任务提供对比代码示例（Windows vs. Android），帮助学生理解平台差异。

表 2 实验任务设计

| 实验任务 | 技术挑战 | 适配方案 |
|---------|--------------------------------|--------------------------------------|
| 画面渲染迁移 | Swing绘制机制 vs SurfaceView绘制机制 | 引入双缓冲机制优化性能, 基于Canvas重写绘图逻辑 |
| 文件存储适配 | Windows文件系统 vs Android沙箱存储 | 使用Android Context API重构文件读写 |
| 排行榜页面重构 | Swing列表组件 vs Android Adapter模式 | 采用 ListView+Adapter, 实现异步数据绑定与高效列表更新 |
| 音频播放重构 | Windows统一音频 vs Android分离式管理 | 整合MediaPlayer与 SoundPool, 在主线程中合理调度 |
| 异步通信重构 | Java原生线程模型 vs Android单线程模型 | 替换原线程模型为 Handler消息队列, 确保UI线程安全 |

为每项迁移任务设置明确的目标与检查点, 确保迁移过程规范高效。引导学生使用 Git 进行版本控制, 每完成一个任务, 需提交一次代码并撰写迁移报告, 详细记录迁移策略、遇到的问题与优化方案, 培养规范的工程文档编写习惯。同时, 设置小组互评环节, 要求学生分析他人代码的适配策略, 提出优化建议, 进一步强化工程规范意识与团队协作能力^[9]。

(3) 应用层整合

在完成适配层开发后, 应用已经能正确运行, 应用层整合的目标是进一步完善系统整体运行流程、提升用户体验、扩展新功能。此阶段要求学生完成以下任务:

- ① 页面导航设计: 适配阶段已实现基础的页面跳转, 本阶段进一步要求设计合理、清晰的页面流程, 包括基于 Activity 栈管理的跳转控制、统一的返回逻辑, 以及异常场景处理 (如用户按 Back 键时的逻辑), 确保导航体验自然流畅。
- ② 用户输入处理: 在完成基本触摸响应适配的基础上, 进一步优化 Touch 事件分发机制优化, 支持多点触控识别, 提升用户交互体验。
- ③ 网络功能扩展实践: 在基本迁移任务完成后, 要求学生拓展网络功能, 如联网排行榜, 联机对战等, 设计具备良好扩展性的客户端-服务器通信架构。

为保障整合质量, 课程设置集体演示环节, 要求各小组展示应用整体流程、功能拓展亮点及系统架构

设计思路, 并接受同伴评议与教师点评。设立“功能拓展积分制”, 鼓励学生自主设计附加功能 (如道具商城、成就系统), 每项创新功能经评审后可获得额外加分。此外, 引入“用户体验评分”机制, 通过学生互评与教师评分, 综合评价界面友好性、操作流畅度等非功能性指标, 全面提升开发质量与用户体验意识。

5 贯通式教学实践效果及应用价值

从 2022 年春首次开课至今, 覆盖 3 个教学周期, 累计逾 1200 名学生参与导论和实践课程。在导论课程中, 95%以上的学生成功实现了项目任务书中的基本功能, 约有 25%的学生在此基础上有所创新。在实践课程中, 所有学生均顺利完成了单机代码的迁移, 80%以上的学生成功实现了基本网络功能的开发。其中, 约三分之一的学生实现了创新性的网络功能, 包括登录注册、道具背包、积分商城等, 实现了更加丰富的游戏效果, 见图 1。课程满意度调查显示, 85%以上的学生成为“课程内容实用性和趣味性强, 显著提升了工程实践能力和跨平台开发能力”, 这一数据与项目完成率相互印证, 充分体现了学生对课程体系的认可。

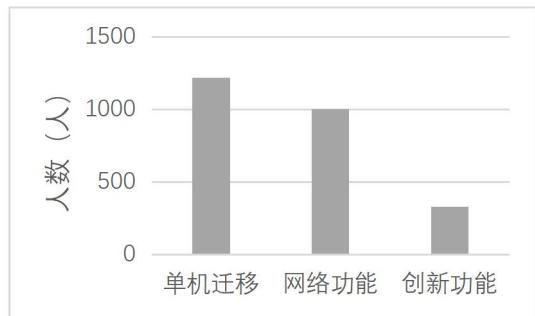


图 1 实践课程分阶段任务完成情况

目前系列课程仍存在一些有待克服的问题。首先, 缺乏标准化和细节化的实验步骤分解, 导致部分学生在实际操作中上手较慢, 甚至出现了无效劳动的情况。其次, 课程目前对学生开发过程中的实际工作量检测不足, 难以准确评估不同学生的学习投入和技能掌握情况。这些问题的存在提示我们需要进一步优化教学设计和评价体系, 以提高实验教学的有效性和公平性。

本系列课程通过贯通式的教学设计, 将导论与实践课程紧密结合, 培养了学生的跨平台开发能力。这种教学设计不仅适用于 Android 与 Windows 之间的迁移实践, 还为学生将来从事其他平台 (如 iOS、Web 或其他嵌入式系统) 开发提供了思维框架和技术基础。通过平台间的差异分析与适配任务, 学生在面对其他平台时能够迅速识别相似的技术挑战, 运用迁移经验完成开发任务, 具备了在多平台间进行系统集成和优化的能力。

课程思政方面,本课程还注重培养学生的工匠精神、创新意识和家国情怀^[10]。通过跨平台开发实践,引导学生深入理解核心技术自主可控的重要性,激发学生投身国产操作系统和软件开发平台的使命感。同时在项目任务中融入团队合作、责任担当等价值观,引导学生不仅关注技术实现,更要思考技术的社会价值。

参 考 文 献

- [1] 言十. 计算机系统能力培养的回顾与前瞻[J]. 计算机教育, 2024(4): 1-6.
- [2] 彭鑫. 以软件构造为中心的软件工程实践化教学探索[J]. 软件导刊, 2021, 20 (06): 1-4.
- [3] 叶佩青, 张辉, 朱陶元敏, 等. 设计贯通式教学模式 践行个性化人才培养理念[J]. 中国大学教学, 2015(07): 43-46.
- [4] 王开军, 阮淑玲, 方莹. 编程语言的核心问题贯通式教学法[J]. 计算机教育, 2022(09): 162-165.
- [5] 谢佳, 房敏等. 面向对象的软件构造导论课程实验教学方法探索[J]. 计算机教育, 2023(3):23-27.
- [6] 代双璠, 王泳微, 肖广德. 大学专业课程实施项目化学习的整体考量[J]. 教育教学论坛, 2024(08): 165-168.
- [7] 刘宏伟, 张宏莉, 王亚东, 郭茂祖, 李治军. 项目驱动的计算机专业实践教学体系初探[J]. 计算机教育, 2015(09): 2-4.
- [8] 张海鑫, 钟将, 汪成亮, 肖春华, 冯永. 面向系统能力培养的开放实验体系建设[J]. 计算机教育, 2023(12): 288-292+297.
- [9] 朱洪军, 伍祥, 朱伟杰. 项目化学习过程性评价模型的设计与实践[J]. 计算机教育, 2021(06): 159-162.
- [10] 唐广花, 朱显红. 课程思政背景下《移动平台开发技术》教学方案研究[J]. 计算机技术与教育学报, 2023, 11(4): 100-104.