基于 JavaParser 的 Java 源代码 HTML 化浏览工具设计与实现

陈 剑[1] 田睿晨[1]
CHEN Jian TIAN Ruichen

摘 要 在大型 Java 项目的维护、代码审查和教学过程中，开发人员往往需要在多个源代码之间查找类、方法、变量以及标准库对象的定义位置。集成开发环境虽然具备较完善的代码导航功能，但对运行环境和项目配置有一定要求，生成结果也不便于直接保存和分发。普通代码高亮工具能够改善代码显示效果，但通常缺少对类、方法和变量之间引用关系的处理。针对这一问题，文章设计并实现了一种基于 JavaParser 的 Java 源代码 HTML 化浏览工具。首先对 Java 源文件进行静态解析，构建抽象语法树，并提取类、接口、方法、变量及其在源代码中的位置信息。在此基础上，对相关符号进行整理和索引，并在 HTML 生成过程中加入关键字高亮、本地文件跳转和 Java API 外部链接。生成的页面可以直接在浏览器中打开，用户能够在不同源文件之间进行基本跳转，也可以查看部分标准库对象对应的 API 文档。与仅用于生成 API 说明文档或进行代码高亮的工具相比，文章所提工具将源码解析、符号信息整理和 HTML 页面生成结合起来，并同时处理项目内部代码链接和 Java 标准库外部链接。在不依赖完整集成开发环境的情况下，生成结果便于离线查看、保存和共享。测试结果表明，系统能够完成 Java 源文件解析、HTML 页面生成、关键字高亮以及主要符号跳转等功能，可用于代码审阅、课程教学、项目交接和一般性源码浏览等场景。

关键词 JavaParser；源代码浏览；静态分析；符号索引；HTML 生成；程序理解；软件维护

doi：10.3969/j.issn.1672-9528.2026.08.009

0 引言

随着软件系统规模的扩大，代码量也随之增长，导致软件在开发、维护、测试和二次开发过程中的源代码阅读变得越来越困难 [1-2]。同时为了理解类之间的继承关系、方法调用关系以及变量引用位置，开发人员在第一次接触 Java 项目时必须在多个文件之间来回切换 [3]。项目规模越大，源文件数量就越多，类层次结构和调用链复杂度越高，代码的理解变得越困难 [4]。

目前所用的源代码阅读工具往往只偏重某一方面功能，JavaDoc 依靠注释为基础来生成 API 文档，能够方便类和方法的阅读，但对实现细节显示不够完整；传统 Java 代码转 HTML 工具能够将语法高亮显示，但对类、方法和变量引用关系不能有效进行结构化处理；IDE 能实现代码导航、补全和调试功能，但对运行环境和项目配置要求较高，安装也较为麻烦，对离线使用、教学演示和审阅来说极为不便 [5-6]。

针对上述问题，本文设计了一种介于 IDE 与普通静态 HTML 高亮工具之间的轻量化方案，该方案以 JavaParser 的抽象语法树解析能力为基础，通过自动提取源代码中的结构化符号信息并将这些信息转化为可浏览、可跳转、可分享的 HTML 页面。该方案既保留源代码原貌，又为源代码阅读增加符号级导航能力，极大地降低了项目熟悉和代码阅读难度。

综上所述，本文设计的系统工具具有如下特点：

一是设计了一种面向 Java 源代码浏览的轻量级静态符号索引方法，该方法能将类、接口、方法、变量和位置数据统一组织为后续生成链接提供基础；

二是设计了一种本地项目符号链接与 Java API 外部链接结合的混合导航机制，该机制能使生成的页面同时实现项目内部和标准库文档的定位；

三是实现从源码输入、AST 解析、符号索引、HTML 渲染到结果输出的模块化流程，该模块化流程能为后续多路径源码、第三方依赖和更复杂的页面导航功能的扩展提供有力支撑。

1 系统需求分析

该工具能够为 Java 开发人员、代码审阅人员和课程教师提供一种轻量、直观、易分发的源代码浏览方式，有效提高

1. 湖南国防工业职业技术学院 湖南湘潭 411207

[基金项目]2025 年湖南省教育厅科学研究优秀青年项目（25B1028）；2025 年湘潭市指导性科技计划项目（CG-ZDJH20241039）

工作效率。用户只需输入源码路径、依赖路径和输出路径，该工具将自动完成源代码分析、符号索引构建和 HTML 文件的生成。用户可直接打开 HTML 文件进行相关浏览而不需要特定的 IDE，可以方便地进行离线阅读、文档归档和项目交接。如表 1 所示。

从功能层面看，系统应具备 Java 文件批量识别、抽象语法树解析、类与接口识别、方法与变量识别、符号位置记录、HTML 页面生成、Java 关键字高亮、本地符号跳转和标准库 API 跳转等能力。从非功能层面看，系统应尽量保持源代码原始缩进和行结构，降低输出页面对外部运行环境的依赖，并通过模块化设计为后续功能扩展保留接口。

2 系统总体设计

本工具采用模块化设计将系统分为输入层、文件处理层、语法解析层、符号索引层、HTML 生成层和输出层。输入层负责读取用户输入的源码路径、依赖路径和输出路径；文件处理层根据输入层传入的源码路径遍历源码目录，筛选 Java 源文件进行分析处理；语法解析层负责调用 JavaParser 对源代码进行静态解析；符号索引层负责将语法解析层解析得到的符号名称、类型、位置和文件关系写入符号索引；HTML 生成层根据索引结果生成带有语法高亮和超链接的 HTML 文件集合；最后输出层负责按要求输出结果。处理流程如图 1 所示。

在系统架构设计上，本工具系统采用分层架构设计，各层只负责处理本层功能实现，如语法解析层负责从 Java 源程序文件中提取结构化信息，符号索引层负责将符号名称、类型、位置和文件关系写入符号索引，HTML 生成层则负责生成符合 HTML 语法规则的页面，并将语法高亮显示和将依赖关系形成超链接。该设计能够有效降低各功能模块之间的耦合关系，在后续新增搜索框、侧边目录、第三方库链接或更多语言解析器时能够只改动相应功能模块，不必对整个系统进行重构。如表 2 所示。

表 1 系统需求与创新对应关系

需求类别	具体需求	实现方式	创新价值
输入与运行	命令行输入源码路径、依赖路径和输出路径	参数校验后触发批量处理流程	降低环境依赖，便于脚本化集成
源码解析	识别 Java 文件中的类、接口、方法、变量和调用节点	基于 JavaParser 构建 AST 并访问关键节点	将普通文本高亮升级为结构化静态分析
符号索引	保存符号名称、类型、文件位置、行列范围和跳转目标	构建轻量级符号映射表	为 HTML 页面提供符号级导航基础
HTML 生成	输出保留缩进和代码结构的网页文件	HTML 转义、样式渲染和逐行输出	兼顾源码原貌与浏览器可读性
链接跳转	支持本地类、方法、变量及 Java API 跳转	本地索引优先，标准库链接补充	形成项目内部与外部文档的混合导航
可扩展性	后续支持多路径源码、第三方库和搜索导航	模块化划分解析、索引、渲染和输出层	便于从原型工具演进为项目浏览平台

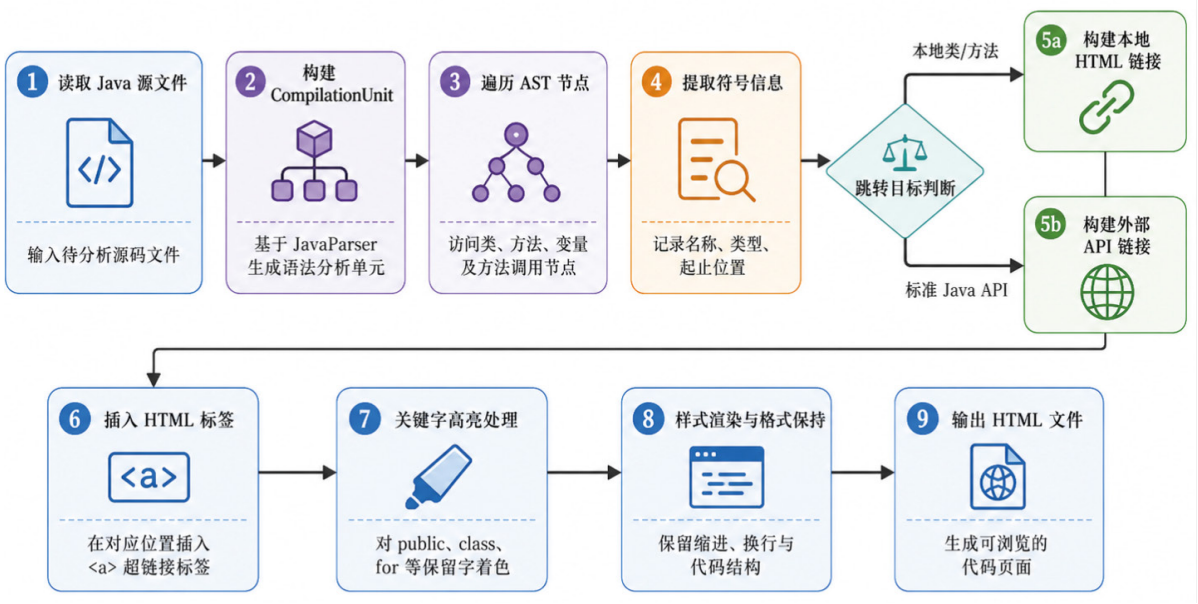


图 1 处理流程图

表 2 系统主要模块及功能说明

模块名称	输入内容	核心处理过程	输出结果	作用
命令行参数模块	源码路径、依赖路径、输出路径	读取并校验用户输入参数	可调用的路径配置	提供工具运行入口
文件遍历模块	Java 项目目录	递归遍历并筛选 Java 文件	Java 文件列表	为解析模块提供文件基础
源码解析模块	Java 源文件	调用 JavaParser 构建 AST	语法节点集合	获取结构化源代码信息
符号索引模块	语法节点和位置数据	记录符号名称、类型、行列范围和目标文件	统一符号映射表	支撑跨文件链接生成
链接解析模块	符号索引与标准库类名	优先匹配本地符号，无法匹配时构建 API 链接	本地或外部跳转地址	形成混合导航机制
HTML 生成模块	原始源码和符号映射	转义字符、插入锚点、添加样式和链接	HTML 页面	实现源码网页化展示
输出管理模块	生成的 HTML 文件	按源文件关系组织输出目录	HTML 文件集合	方便用户浏览和分发

3 关键技术与实现

3.1 基于 JavaParser 的源代码解析

JavaParser 能够将 Java 源文件转换为抽象语法树，并提供较为丰富的节点访问接口。系统在读取源文件后生成 CompilationUnit 对象，再通过访问 ClassOrInterfaceDeclaration、MethodDeclaration、VariableDeclarator、FieldDeclaration 和 MethodCallExpr 等节点，提取类名、接口名、方法名、变量名、调用信息以及节点所在的行列位置 [7]。

与直接使用正则表达式进行文本匹配相比，基于 AST 的解析方式能够结合代码结构识别不同类型的语法元素，对声明、方法调用、关键字和普通标识符的区分更加明确，也能减少注释、字符串内容和复杂语句结构带来的误判。本文工具正是在这一解析结果的基础上获取符号及其位置信息，并进一步生成相应的跳转链接 [8]。

3.2 轻量级符号索引构建

该工具系统设计了一种轻量级符号索引结构将分散在多个源程序文件中的符号进行关联。索引中的每个符号记录包括符号名称、符号类型、所属文件、起始行列、结束行列和目标锚点等。对于类和接口，索引主要用来实现文件跳转；对于方法和变量，索引主要用于定位其声明位置并在生成的页面中锚定位置。

该工具系统采用的符号索引不追求完整编译器级语义分析能力，而是紧紧围绕在“浏览”场景下有效信息显示的需求。该策略有效降低了系统的实现难度和工具依赖，能够在没有 IDE 工具的情况下完成大部分源代码导航任务，有效提高工具的使用便捷性。

3.3 符号位置映射与超链接生成

HTML 化浏览的关键技术在于如何实现将源代码中的符号与其目标定义位置建立链接关系。为实现上述功能，系统在对每个 Java 文件进行分析时记录节点位置，并根据行列坐标判断当前位置是否对应某个符号。如果当前位置为符号起点则在生成 HTML 页面时插入超链接开始标签，在符号终点位置则在生成 HTML 页面时插入超链接结束标签 [9]。该方式下系统能够在保留原始代码格式情况下为符号添加超链接。

对于类和方法，如果是本地项目，则系统将跳转目标指向对应的 HTML 文件及页面锚点；如果是 Java 标准库的类和方法，因其无法在本地索引中解析，系统通过构建 Java API 外部链接，并将跳转目标指向该链接 [10]。

3.4 关键字高亮与格式保持

本工具系统为提升页面可读性将对 public、private、class、interface、for、while、if、else、return、int 和 void 等 Java 关键字进行高亮显示。系统先对源代码进行 HTML 转义，转义完成后再对关键字、符号链接和样式标签等进行处理，能够有效避免源代码中的尖括号、引号和缩进字符破坏页面结构。

为保持源代码中的格式，系统按照源文件原始行结构逐行输出，输出过程中尽量保留空格、制表符和换行等字符，该输出方法能够有效减少标签插入时对代码缩进和代码阅读的影响。

3.5 命令行运行与输出管理

系统以命令行作为人机交互入口，接收 Java 源代码路径、依赖库路径、HTML 输出路径三类参数后，自动完成文件遍历、源码解析、索引构建和页面生成。相较于图形化程序，命令行运行方案更易与批处理脚本、教学资源批量生成、集成开发平台联动，具备较强工程复用性，并能有效实现后续拓展项目文档自动化功能。

4 系统测试与结果分析

为验证系统功能，本文从运行入口、源码读取、语法解析、符号索引、HTML 生成、超链接跳转、关键字高亮和格式保持等方面进行了测试。测试重点不仅关注文件是否成功生成，还关注生成结果是否能够支撑源代码浏览任务，如表 3 所示。

测试表明，该系统不仅能够完成对 Java 源代码的读取与解析并生成可浏览的 HTML 页面，而且还能够对常见的类、方法和变量进行识别并建立链接，实现源代码浏览和跨文件查看。但该系统也存在一定的不足，在复杂方法调用、局部变量作用域以及代码格式保持等一些极端情况下，仍可能发生符号定位不准或页面显示偏差的情况。在后续的研究中可通过改进符号解析和位置匹配方法，提高系统处理复杂代码时的稳定性。

5 与相关工具的比较

为明确本系统工具的功能和其他相关工具的区别，选取 JavaDoc、Java2HTML 类工具和集成开发环境进行比较。JavaDoc 主要用于生成较为规范的 API 文档，便于查看类、方法及参数说明，但对具体实现过程的展示有限。Java2HTML 类工具能够将源代码转换为网页形式，并提供基本的语法高亮，但通常缺少类、方法等代码元素之间的关联跳转。集成开发环境具备比较完善的代码导航、补全和调试功能，但对运行环境和项目配置要求较高。相比之下，本文工具更侧重于轻量化使用，即便是在不完整开发环境的情况下，依然为用户提供源代码展示、关键字高亮和基本链接跳转功能。如表 4 所示。

表 3 系统功能测试结果

测试编号	测试内容	测试方法	预期结果	实际结果	结论
T01	命令行运行	输入源码、依赖和输出路径	系统正常启动并读取参数	能够执行转换任务	通过
T02	Java 文件读取	遍历指定源码目录	识别并读取 Java 文件	能够获取目标文件列表	通过
T03	类与接口识别	解析含类和接口声明的文件	提取类名与接口名	能够提取主要类和接口	通过
T04	方法声明识别	解析包含多个方法的文件	识别方法名称与位置	能够获取方法声明信息	通过
T05	变量声明识别	解析成员变量和局部变量	识别变量名称、类型和位置	能够识别主要变量声明	通过
T06	符号索引构建	汇总多个文件的解析结果	形成统一符号映射	能够支持后续链接生成	通过
T07	HTML 文件生成	转换 Java 源文件	输出可打开 HTML 页面	能够生成页面文件	通过
T08	本地类跳转	点击页面中的本地类名	跳转至对应类页面	主要本地类可完成跳转	通过
T09	方法跳转	点击页面中的方法名	跳转至方法声明位置	部分方法可实现跳转	基本通过
T10	Java API 链接	点击标准库相关类名	跳转至官方 API 文档	部分标准库类可跳转	基本通过
T11	关键字高亮	查看常见 Java 保留字	关键字显示设定样式	主要关键字能够高亮	通过
T12	格式保持	检查缩进、换行和代码结构	基本保持原始排版	复杂结构存在少量偏差	部分通过

表 4 本工具与相关工具功能对比

对比项目	JavaDoc	Java2HTML 类工具	集成开发环境	本文工具
主要用途	生成 API 文档	源代码 HTML 展示	开发、调试和维护	源代码 HTML 化浏览与符号跳转
完整源代码展示	否	是	是	是
语法高亮	一般不强调	支持	支持	支持
类跳转	文档层面支持	通常较弱	支持	支持
方法跳转	文档层面支持	通常较弱	支持	部分支持
变量跳转	不支持	通常不支持	支持	部分支持
符号索引	依赖注释和声明	一般缺少	工程模型完整	轻量级静态索引
Java API 外链	支持文档链接	一般缺少	可配置	部分支持
环境依赖	较低	较低	较高	较低
离线分发	较方便	较方便	一般	较方便
创新侧重点	注释文档化	代码可视化	开发全流程支持	解析、索引与 HTML 浏览融合

6 不足与展望

该系统工具已经能够实现 Java 源代码解析、HTML 页面生成、关键字高亮和基本链接跳转，但在面对复杂工程项目时仍存在不能有效处理的情况，现阶段系统更适合处理源码相对集中的 Java 项目。具体表现如下：

一是面对目录层级较多、源文件分散的工程项目，其文件读取和索引管理会变得更加复杂。如 Maven、Gradle 项目通常包含多个源码目录，该系统工具在处理这类项目中存在一定局限，在后续的研究中可通过补充多路径读取和源码根目录识别功能，并统一管理不同目录中的源文件和符号信息；

二是面对第三方依赖也未实现有效处理。本系统工具能够为部分 Java 标准库内容建立 API 链接，但对于外部类库只依靠本地源码往往无法确定其具体来源，后续研究中可以考虑从 pom.xml、build.gradle 等配置文件中读取依赖信息，再根据依赖名称和版本关联相应的 Javadoc 文档、源码仓库或项目主页；

三是在符号解析方面，该系统工具在一般代码中能够正常工作，但遇到方法重载、同名变量、内部类、泛型和多层作用域时，其当前索引较多依赖符号名称和位置信息，因对类型和上下文考虑不足仍存在匹配错误问题发生。后续研究中可以引入 JavaParser Symbol Solver，结合参数、类型和作用域信息进行判断，对于结构更加复杂的项目也可借助编译器语义信息提高方法调用和变量引用的识别准确性 [11-12]；

四是目前生成页面的功能也相对简单，当项目规模增大后，仅依靠普通 HTML 页面查找代码并不方便。后续研究中可以加入类列表、方法目录、搜索、侧边导航和代码折叠等功能；

五是在测试方面仍保留较多人工检查，容易漏掉特殊情况。后续研究中应增加 JUnit 测试和链接有效性检查，对源码解析、HTML 生成和页面跳转等功能进行重复验证，如表 5 所示。

7 结论

本文设计并实现了一种基于 JavaParser 的 Java 源代码 HTML 浏览工具。该系统工具首先通过静态解析获取源代码中的类、接口、方法和变量等信息并记录其在代码中的位置；然后在此基础上建立符号索引并将 Java 源文件转换为带有语法高亮、页面定位和超链接跳转功能的 HTML 文件。该系统工具生成结果可直接在浏览器中打开，对于本地代码用户通过点击相关符号跳转到对应的本地代码位置；对于部分 Java 标准库内容则可进一步链接到相关 API 文档位置。

现有工具 JavaDoc 侧重类和方法说明，普通代码转 HTML 工具主要用于代码展示和语法高亮，集成开发环境功能较多，但需要安装和配置相应环境。与现有工具相比，本文工具主要用于较为轻量的源代码浏览，能够在显示完整源代码的基础上增加了基本跳转功能，生成的 HTML 文件也便于保存和分享。

从使用情况看，该系统工具能够方便用户了解项目源代码的主要结构，在代码审阅、教学演示和项目交接等场境中有较强的应用价值。但是，该系统工具面对复杂目录、第三方依赖和部分符号关系的处理还不够完善，在后续研究中可继续补充多路径源码读取、第三方库链接、符号识别、页面导航和自动化测试等功能。

表 5 系统存在问题与改进方向

存在问题	具体表现	可能原因	改进方向
多路径源码支持不足	对分散在不同目录下的 Java 文件处理能力有限	当前文件读取逻辑主要面向单一目录	增加多路径输入和统一索引机制
第三方库链接不完善	无法准确跳转第三方库类和方法	缺少 Maven、Gradle 依赖解析	读取依赖配置并构建外部文档链接
复杂符号定位存在偏差	重载方法、同名变量或复杂作用域识别不稳定	轻量索引缺少完整语义分析	引入 Symbol Solver 或编译器级语义信息
代码格式保持仍需优化	复杂缩进或标签插入可能影响展示	HTML 标签与源文本位置映射存在冲突	优化转义、分词和逐字符渲染流程
自动化测试不足	当前测试更多依赖人工检查	缺少系统化单元测试和回归测试	引入 JUnit、快照测试和链接有效性检测
页面交互功能基础	缺少搜索、目录和折叠功能	当前重点在核心转换流程	增加侧边导航、全文搜索和代码结构树
多语言支持缺失	只能处理 Java 源代码	核心解析依赖 JavaParser	接入 Python、C++、JavaScript 等语言解析器

（下转第 53 页）

[2026-07-29]. https://github.com/ultralytics/ultralytics.
[13]Ultralytics. Model training with ultralytics YOLO: train mode [EB/OL].[2026-07-29]. https://docs.ultralytics.com/modes/train/.
[14]Ultralytics. Ultralytics YOLO configuration[EB/OL]. [2026-03-03]. https://docs.ultralytics.com/usage/cfg/.
[15] Lin T-Y, Maire M, Belongie S, et al. Microsoft COCO: common objects in context[C]//Computer Vision-ECCV 2014. Berlin: Springer,2014: 740-755.
[16] Lin T-Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017: 2117-2125.
[17] He Kaiming, Gkioxari G, Dollár P, et al. Mask R-CNN[C]//Proceedings of the IEEE International Conference on Computer Vision. Piscataway:IEEE,2017: 2961-2969.
[18] Bolya D, Zhou Chong, Xiao Fanyi, et al. YOLACT: real-time instance segmentation[C]//Proceedings of the IEEE International Conference on Computer Vision.Piscataway:IEEE, 2019: 9157-9166.
[19] Bochkovskiy A, Wang C-Y, Mark Liao H-Y. YOLOv4: optimal speed and accuracy of object detection[PP/OL]. (2020-04-23)[2026-07-29].https://doi.org/10.48550/arXiv.2004.10934.
[20] Zhang Minghua, Wang Zhihua, Song Wei, et al. Efficient small-object detection in underwater images using the enhanced YOLOv8 network[J]. Applied Sciences, 2024, 14(3): 1095.
[21] Rezatofighi H, Tsoi N, Gwak J , et al. Generalized intersection over union: a metric and a loss for bounding box regression[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway:IEEE,2019: 658-666.
[22] Ultralytics. YOLO performance metrics[EB/OL]. [2026-03-03].https://docs.ultralytics.com/guides/yolo-performance-metrics/.
[23] Wang Li, Tang Chao. Effective small crack detection based on tunnel crack characteristics and an anchor-free convolutional neural network[J/OL]. Scientific Reports, 2024[2026-07-29].ttps://www.nature.com/articles/s41598-024-60454-3.

【作者简介】

张力文（1999—），通信作者（email: 3023068268@qq.com），男，江苏溧阳人，本科，助理工程师，研究方向：计算机视觉。

高章超（1996—），男，安徽安庆人，本科，工程师，研究方向：数字化、BIM 技术。

董乐天（1999—），男，上海人，硕士研究生，助理工程师，研究方向：空间信息技术。

党维维（1983—），女，陕西富平人，硕士研究生，高级工程师，研究方向：大型基础设施工程数字化、工程咨询。

（收稿日期：2026-06-30 修回日期：2026-08-05）

（上接第 46 页）

参考文献：

[1] 乔善平，吴鹏，赵燕，等 .Java 程序自动评测系统的设计与实现 [J]. 计算机教育 ,2026(5):247-252.
[2] 徐静 .AI 赋能 Java 实验教学：语法向工程的链式重构 [J]. 电脑知识与技术 ,2026,22(8):166-168.
[3] 计卫星，王贵珍，李侃 . 编译原理语言认知实验设计与实践 [J]. 计算机教育 ,2019(8):94-97.
[4] 张芮斌，石佳伟 . 基于注意力增强型 RNN 的 Java 代码上下文感知补全应用研究 [J]. 软件，2026,47(2):49-51.
[5] 张桓，徐丽 .Java 语言程序设计与实现：微课版 [M]. 北京：人民邮电出版社，2023.
[6]Markus Gebhard. Java2Html: Java to HTML Converter[EB/OL].[2026-07-29]. http://www.java2html.de/.
[7] Duvall P, Matyas S, Glover A. Continuous integration: improving software quality and reducing risk[M]. Boston: Addison-Wesley, 2007.
[8] Fowler M. Refactoring: improving the design of existing code[M]. Boston: Addison-Wesley, 1999.
[9] JavaParser. JavaParser Documentation[EB/OL].[2026-07-29]. https://javaparser.org/.
[10] 林育蓓，汤德佑，汤娜 . 数据库技术及应用 [M]. 北京：机械工业出版社，2024.
[11] 何炎祥 . 编译原理 [M]. 武汉：华中科技大学出版社，2023.
[12] 陈火旺，刘春林，谭庆平，等 . 程序设计语言编译原理 [M]. 北京：国防工业出版社，2010.

【作者简介】

陈剑（1983—），男，湖南祁东人，硕士研究生、高级工程师，研究方向：信息技术、区块链等。

田睿晨（1997—），女，湖南怀化人，博士研究生、工程师，研究方向：机器学习等。

（收稿日期：2026-07-08 修回日期：2026-08-06）