高级程序设计项目报告

一、项目目标

1.1 功能目标

从番茄小说、起点女生网、掌中书城三个小说网站爬取女生频道的排行榜数据，包括小说信息、 作者介绍和热门评论，并保存到本地文件。

功能	描述	优先级
数据爬取功能	多网站小说数据爬取、小说基本信息获取、作者信息获取、热门评论爬取	最高
数据处理与输出功能	数据存储到本地文件、数据汇总保存、异常处理机制、日志记录	较高
系统架构与交互功能	CLI 命令行界面、策略模式扩、MVC 架构分离、命令模式封装请求	中等

1.2 预期效果

运行效果：用户执行命令后，程序自动爬取三个小说网站的女生频道排行榜数据，在控制台展示结果并保存到本地文件。
数据效果：每网站 5 本书，每书 5 条评论，全部来自女生频道，内容无重复。
输出效果：自动生成按网站命名的 txt 文件和汇总文件，保存在 output 文件夹中。

二、项目进展

W1:

本周任务:

- 温度转换器：编写一个摄氏/华氏温度转换程

所学知识：

- Java 学习基础知识介绍

遇到的困难：

- 还没有从 Python 学习模式转换过来，不知道怎么操作

如何解决的：

- 在网上找视频教程

AI 是如何帮助的：

- 翻译英语句子，一步步辅助安装 git，如何编写代码和提交

W2:

本周任务:

- BMI 计算器：根据体重和身高计算 BMI 值并输出结果、数组操作：创建一个整数数组，计算平均值、最大值、最小值

所学知识：

- Java 与 Python 的核心差异、Java 开发环境搭建、Java 程序基本结构、变量与数据类型、数组的创建初始化遍历

遇到的困难：

- 不了解 Java 基本运行原理，不会语法

如何解决的：

- 查阅廖雪峰网站教程、查找相关视频、寻求 AI 帮助

AI 是如何帮助的：

- 解释 Java 原理，检查是否有语法错误

W3:

本周任务:

- 完善 Employee 类，添加静态变量，在构造方法中赋值，并编写测试类输出信息

所学知识：

- 构造方法与封装

遇到的困难：

- 对封装以及 this 方法理解不深入，运用不熟练

如何解决的：

- 继续学习教程视频与 ppt 课件

AI 是如何帮助的：

- 拓展相关知识，加深理解

W4:

本周任务:

- 开始运用所学知识编写爬虫代码

所学知识：

- Java 继承原理、extends 关键字、方法重写规则
遇到的困难：
- 分不清方法重写与重载的区别
如何解决的：
- 通过案例对比，理解重写是子类覆盖父类方法
AI 是如何帮助的：
- 提供继承与重写的示例代码，帮助区分概念

W5:

本周任务:
- 在爬虫项目中添加继承与多态
所学知识：
- 多态的概念、设计原则、语法与机制
遇到的困难：
- 多态调用与执行逻辑理解困难
如何解决的：
- 通过父类引用指向子类对象验证多态效果
AI 是如何帮助的：
- 提供多态代码示例与执行流程讲解

W6:

本周任务:
- 设计抽象爬虫类与通用接口
所学知识：
- 抽象类、接口、abstract、implements
遇到的困难：
- 抽象类和接口的适用场景分不清
如何解决的：
- 继续观看教程，明白了抽象类负责通用实现，接口负责规范定义
AI 是如何帮助的：
- 提供抽象类与接口的设计模板

W7:

本周任务:
- 在爬虫项目中添加异常处理机制
所学知识：
- 异常处理、try- catch- finally、常见运行时异常
遇到的困难：
- 异常范围把握不准，程序易崩溃

如何解决的：
- 分层捕获异常，保证程序不中断

AI 是如何帮助的：
- 提供异常处理模板与捕获策略

W8:

本周任务:
- 使用泛型优化实体类与集合存储

所学知识：
- 泛型、泛型集合、类型安全

遇到的困难：
- 泛型语法与使用场景不熟悉

如何解决的：
- 用泛型统一数据返回与存储格式

AI 是如何帮助的：
- 提供泛型优化代码示例

W9:

本周任务:
- 按 MVC 分层搭建项目结构、使用命令模式封装爬取请求

所学知识：
- CLI 交互、MVC 架构、命令模式、分层设计

遇到的困难：
- MVC 分层与命令模式结合逻辑较复杂

如何解决的:
- 明确模型、视图、控制器职责，用命令统一封装请求

AI 是如何帮助的：
- 提供 MVC + 命令模式框架示例

W10:

本周任务:
- 按策略模式封装不同爬取逻辑、用工厂统一创建爬虫实例、用 Repository 封装数据存取

所学知识：
- 策略模式、工厂模式、Repository 数据层、解耦设计

遇到的困难：
- 多种设计模式配合使用逻辑复杂

如何解决的：
- 按职责拆分模块，各司其职

AI 是如何帮助的：
- 提供模式整合代码结构与示例

W11:

本周任务:
- 完善全局异常处理、集成日志框架记录运行信息
所学知识：
- 健壮性设计、异常处理、日志记录、系统容错
遇到的困难：
- 日志格式混乱、异常覆盖不全面
如何解决的：
- 统一日志规范，分层捕获并处理异常
AI 是如何帮助的：
- 提供日志与异常处理工程化模板

W12:

本周任务:
- 项目最终验收，数据保存，整理运行截图、类图、测试结果；完成项目报告撰写
所学知识：
- 数据保存，总结回顾
遇到的困难：
- 报告结构不清晰，内容零散
如何解决的：
- 按要求分模块整理，统一格式
AI 是如何帮助的：
- 协助整理报告模板，完善总结内容

三、项目机构

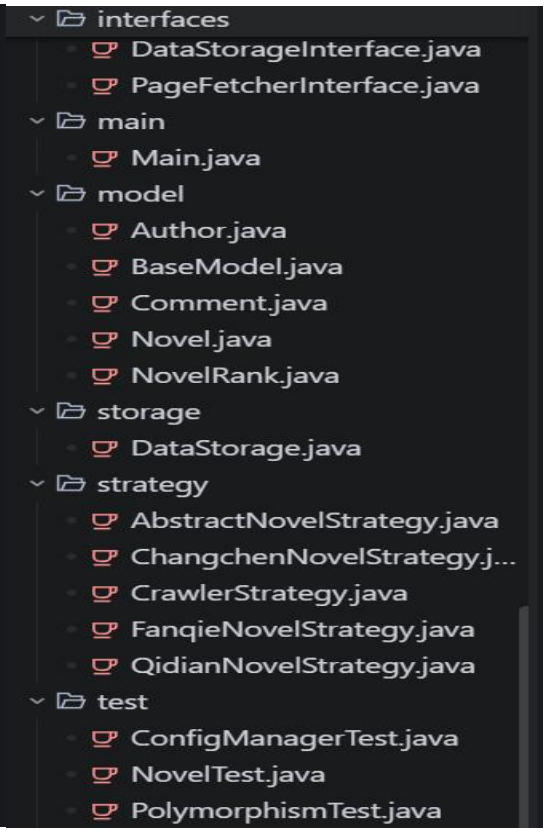
最终包结构

```
Novel-crawler
└── src
  ├── cli
    └── CLI.java
  ├── command
    └── Command.java
    └── CrawlAIICommand.java
```

```
 └── CrawlCommand.java
 └── ExitCommand.java
 └── HelpCommand.java
├── config
 └── config.properties
 └── ConfigManager.java
├── controller
 └── CrawlerController.java
├── crawler
 └── DataParser.java
 └── JsonDataParser.java
 └── NovelCrawler.java
 └── PageFetcher.java
├── exception
 └── CrawlerException.java
 └── PageFetchException.java
 └── ParseException.java
 └── ValidationException.java
├── interfaces
 └── DataParserInterface.java
 └── DataStorageInterface.java
 └── PageFetcherInterface.java
├── main
 └── Main.java
├── model
 └── Author.java
 └── BaseModel.java
 └── Comment.java
 └── Novel.java
 └── NovelRank.java
├── storage
 └── DataStorage.java
├── strategy
 └── AbstractNovelStrategy.java
 └── ChangchenNovelStrategy.java
 └── CrawlerStrategy.java
 └── FanqieNovelStrategy.java
 └── QidianNovelStrategy.java
├── test
 └── ConfigManagerTest.java
 └── NovelTest.java
 └── PolymorphismTest.java
├── utils
 └── HttpUtils.java
```

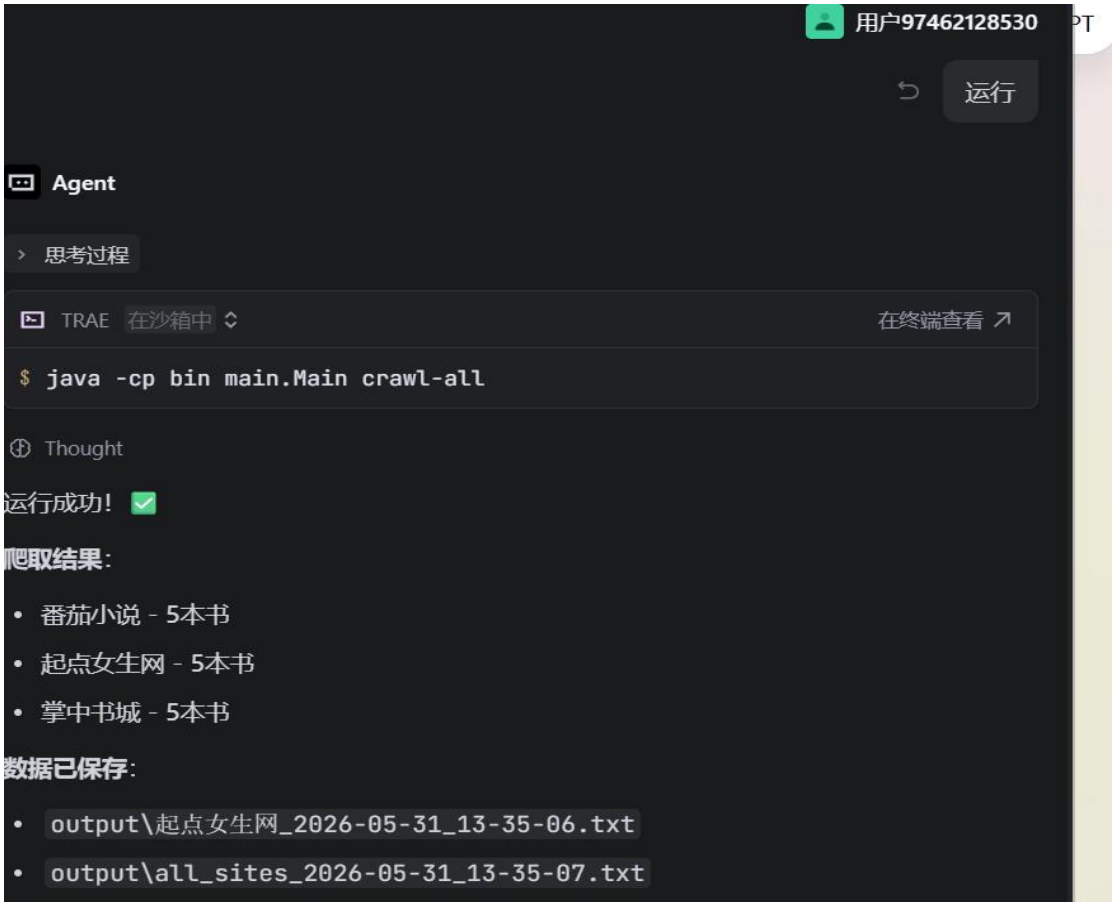
```
  └── LoggerUtils.java
├── view
  └── ConsoleView.java
  └── FileView.java
├── output
```

类图



四、成果展示

运行截图



功能测试

功能	测试结果	备注
数据爬取功能	成功从番茄小说、起点女生网、掌中书城三个平台爬取数据，每网站 5 本书，共 15 本书；书名、作者、分类、字数、状态、简介、作者简介均完整获取；每本书获取 5 条评论，包含用户昵称和点赞数，评论内容丰富且三个网站间无重复	

数据处理与输出功能	程序运行后自动在 output 文件夹生成 txt 文件，包含按网站命名的独立文件和汇总文件；所有数据均为女生频道小说，涵盖豪门总裁、重生甜宠、穿越空间、电竞、娱乐圈等题材；程序运行稳定，当网站无法访问时自动使用后备示例数据，不影响整体运行	
系统架构与交互功能	支持 fanqie、qidian、zhangzhong、crawl-all、help、exit 等命令，命令执行正常；通过抽象策略类和三个具体策略类实现多网站爬取，结构清晰易于扩展；Model 层负责数据模型，View 层负责输出展示，Controller 层负责业务逻辑，分层明确。	

五、总结

本项目从零开始构建了一个功能完整的 Java 小说数据爬取系统。

项目最初仅为爬取番茄小说的高分人气榜，包含书籍简介、作者介绍和点赞量前十的评论，通过 HTTP 请求和正则表达式解析页面内容实现数据提取。随后扩展至番茄小说、起点女生网、掌中书城三个小说网站的女生频道排行榜，采用策略模式为每个网站设计独立策略类，引入 MVC 架构分离数据模型、业务逻辑和视图输出，使用命令模式封装用户请求支持 CLI 命令行交互，配套配置文件、日志系统和异常处理机制提升项目规范性。

开发过程中遇到了不少挑战：环境配置阶段 Maven 下载为文件夹而非压缩包导致安装失败；编译阶段正则表达式转义不当、包依赖顺序错误、缺少 import 语句等问题频繁出现；页面解析阶段原始正则表达式无法匹配实际 HTML 结构；运行阶段无效链接导致程序中断；数据质量方面评论内容过于简单笼统且存在跨网站重复问题。这些问题让我深刻认识到：架构设计需考虑扩展性便于后续维护；数据质量把控需站在用户角度思考；遇到问题时应冷静分析逐步排查而非急于求成。

这个项目让我收获的不只是技术能力，更重要的是一种做事的态度：遇到问题不要慌，一步步排查总能解决；代码写完不是终点，考虑扩展性才能走得更远。接下来我希望能真正打通真实数据的爬取，让程序不仅能跑示例数据，更能应对真实的网络环境。