

Text RPG + 地图行走的完整示例流程

0. 游戏开始 & 初始地图

```
===== Echoes of the Forgotten Forest
=====
```

You wake up near the Village of Dawn.
You feel a strange connection to this forest...

Controls:

```
w = up, s = down, a = left, d = right
m = show map
e = explore / interact
q = quit
```

Your HP: 40/40 MP: 15/15

Type 'm' to see the map.

> m

地图显示 (玩家用 @ 表示)

World Map:

```
# # # # # # #
#       C   T #
# F F F . T #
# F @ F . . #
#   V   . M #
# # # # # # #
```

Legend:

```
@  You (Player)
V  Village of Dawn
F  Forest of Mist
C  Whispering Canyon
T  Bell Tower Ruins
M  Mirror of Truth (final boss)
#  Mountain (cannot pass)
.  Path / Grassland
```

1. 在地图上移动 (Walking on the Map)

玩家输入方向：

```
> w      (move up)
```

You move north...

```
# # # # # # #
#      C      T #
# F @ F . T #
# F F F . . #
#   V   . M #
# # # # # # #
```

Current Location: Forest of Mist

再往上走：

```
> w
```

You move north...

```
# # # # # # #
#   @ C      T #
# F F F . T #
# F F F . . #
#   V   . M #
# # # # # # #
```

You feel a chilling breeze from the canyon...

Current Location: Whispering Canyon

现在学生就能“看见”：

- 玩家在 F 区移动
- 再往上是 C 区（峡谷）
- 再往右可以走到 T 区（塔）

2. 在行走中触发遇怪 (Random Encounter on Move)

当玩家在森林或峡谷走动时，有一定概率遇怪（例如 30% 的概率）：

> s (玩家从 c 向下走向森林)

You move south...

```
# # # # # # #  
#       C   T #  
# F @ F . T #  
# F F F . . #  
#   V   . M #  
# # # # # # #
```

Current Location: Forest of Mist

As you step between the trees, something moves in the shadows...

A Slime appears!

=== BATTLE START ===

Player HP: 40/40 MP: 15/15

Slime HP: 20/20

此时进入战斗模式。

3. 战斗过程 (Battle Demo)

玩家有几种操作：

Choose your action:

- 1) attack
- 2) skill fireball
- 3) use potion
- 4) run

> 1

You attack the Slime for 7 damage!

Slime HP: 13/20

Slime attacks you for 4 damage!

Your HP: 36/40

> 2

You cast Fireball!

The Slime takes 11 damage!

Slime HP: 2/20

Slime tries to attack but misses!
战斗结束:

> 1

You defeat the Slime!

You found: Health Potion ×1
You gain 5 EXP.

You return to the map...

```
# # # # # # #  
#       C   T #  
# F @ F . T #  
# F F F . . #  
#   V   . M #  
# # # # # # #
```

Current Location: Forest of Mist

4. 一路走到塔 (T) 去打小 Boss

玩家决定向右走:

> d

You move east...

```
# # # # # # #  
#       C   T #  
# F F @ . T #  
# F F F . . #  
#   V   . M #  
# # # # # # #
```

Current Location: Path to the Bell Tower

> d

You move east...

```
# # # # # # #
#       C   T #
# F F F @ T #
# F F F . . #
#   V     . M #
# # # # # # #
```

You arrive at the Bell Tower Ruins...

触发小 Boss 战:

A towering knight steps in your way.

Tower Guardian:

"Halt. Only those who confront their past may enter."

=== BOSS FIGHT ===

Player HP: 38/45 MP: 20/20

Guardian HP: 85/85

学生可以很直观理解:

- 地图上从森林走到塔
- 到塔自动触发 Boss 战

5. 继续行走，到最终 Boss 区域 (M)

击败 Tower Guardian 后，玩家可以前往 M 区:

> s

You move south...

```
# # # # # # #
#       C   T #
# F F F . T #
# F F F . @ #
#   V     . M #
# # # # # # #
```

Current Location: Mirror Lake

A glowing mirror stands silently on the water's surface...
触发最终 Boss:

A shadow steps out of the mirror.

Mirror Shade:

"I am you. The part you tried to forget."

=== FINAL BATTLE START ===

Player HP: 50/50 MP: 30/30

Mirror Shade HP: 120/120

6. 游戏结束 (Win / Lose) 示例

WIN 结局:

You strike the final blow!

Mirror Shade shatters into pieces of light...

"You have accepted your past. The forest no longer binds you."

The mist fades away.

Sunlight returns to the forest.

=== YOU WIN ===

Thank you for playing.

LOSE 结局:

Your HP drops to 0...

You fall to your knees.

The forest grows cold and silent.

=== GAME OVER ===

Try again?

(1) Yes (2) No

Java RPG 4 周实践项目完整设计 (Text RPG)

Week 1 — 核心角色系统与战斗基础 (角色 / 敌人 / 战斗模型)

学习目标

- 掌握面向对象建模：抽象类 / 继承 / 多态
- 会设计 Character、Player、Enemy
- 设计属性 (HP, MP, attack, defense 等)
- 构建最基本的战斗循环 (Turn-based battle)

本周模块

1. 角色系统 (Character Hierarchy)
2. 基础战斗系统 (BattleEngine)
3. 控制台主循环 (Console Game Loop)

课堂任务（两节课）

◆ 第 1 节：角色与继承模型

建立：

```
Character (abstract)
    ↑
Player      Enemy
```

字段：

```
String name;
int hp, mp;
int attack, defense;
```

方法：

```
boolean isAlive();
void takeDamage(int dmg);
void attack(Character target);
abstract void displayStatus();
```

练习点：

- 构造方法继承
- `super()` 调用
- 方法重写（Enemy 重写 `displayStatus`）

◆ 第 2 节：战斗系统 **BattleEngine** 初版

Demo 示例：

```
BattleEngine engine = new BattleEngine(player, enemy);
```



```
engine.startBattle();
```

核心流程：

Player turn → Enemy turn → Check alive → Loop

打印战斗日志：

You attack Slime for 8 damage!

Slime attacks you for 3 damage!

用 **Scanner** 输入命令：

```
> attack
```

```
> status
```



线下任务（结合 Vibe Coding）

1. 填充更多 Enemy 类型，如 Goblin, Dragon（AI 可帮生成属性）
2. 实现 defense 计算
3. 美化 displayStatus 输出
4. 扩写 BattleEngine：添加 miss/crit（AI 可生成概率表）



Week 1 最终产出

- 可创建角色、可生成怪物
- 可以进行最基本的战斗
- 一个跑通的 main 函数

—————

Week 2 — 背包系统 / 道具系统 / 技能系统 (Introducing Collections)

学习目标

- 使用 `List`, `Map` 存储游戏对象
- 设计接口: `Skill`, `Item`
- 多态应用在技能与道具使用上
- 体验策略模式的思想（可简单解释）

本周模块

1. `Inventory` 背包系统
2. 道具系统 `Item`
3. 技能系统 `Skill`

课堂任务（两节课）

◆ 第 1 节: `Inventory` (`List`) + 道具系统 (`Item` 接口)

设计接口:

```
interface Item {  
    String getName();  
    void use(Character user);  
}
```

示例实现:

- `HealthPotion`
- `ManaPotion`

背包类：

```
class Inventory {
    private List<Item> items;

    public void addItem(Item item) {}
    public void useItem(int index, Character user) {}
    public void displayItems() {}
}
```

练习点：

- 用 `ArrayList<Item>`
- 道具使用后从 `List` 移除
- 道具调用对角色属性操作 (`hp++`, `mp++`)

◆ 第 2 节：技能系统 Skill（接口 + 多态）

接口：

```
interface Skill {
    String getName();
    void apply(Character caster, Character target);
}
```

技能示例：

```
class Fireball implements Skill { ... }
class Heal implements Skill { ... }
```

在 `Player` 中加入：

```
List<Skill> skills;
```

课堂练习：

- 实现技能菜单
- 玩家输入选择技能编号
- 触发技能效果、扣 MP

线下任务（结合 Vibe Coding）

- 设计 3 个怪物技能
- 设计 Buff/Debuff class（进阶可选）
- 添加 cooldown（冷却时间）
- 用 `HashMap<String, Item>` 做可快速查找的道具仓库

Vibe Coding Prompt 推荐：

“帮我为我的文字 RPG 生成 10 个技能（包括名字、描述、伤害公式、MP 消耗）。”

Week 2 最终产出

- 可使用技能
- 可使用道具
- 背包管理系统
- 战斗系统升级为“可技能 + 可道具”的版本

Week 3 — 世界系统 / 地图 / 事件系统（引入 Map 和组合关系）

学习目标

- 使用组合（HAS-A）组织大型系统
- 使用 Map 构建游戏世界
- 完成简易场景系统 & 事件触发机制
- 更复杂设计：Game Engine 层次

本周模块

1. **Location** (场景)
2. **GameMap** (地图)
3. 事件系统 (进入场景触发事件)
4. 命令解析器 (**CommandParser**)

课堂任务 (两节课)

◆ 第 1 节: 设计 **Location** 与世界地图 (**Map**)

场景类:

```
class Location {  
    private String name;  
    private String description;  
    private List<Enemy> enemies;  
    private List<Item> loot;  
}
```

世界结构:

```
class GameWorld {  
    private Map<String, Location> locations;  
}
```

学生任务:

- 建立 3-5 个 Location
- 每个 Location 配置不同怪物
- 实现 basic movement:

```
> go forest  
> go cave
```

◆ 第 2 节: **Command** 系统 (命令解析器)

简易 DSL:

```
attack
use potion
skill fireball
go cave
talk villager
status
```

命令解析器：

```
class CommandParser {
    public Command parse(String input) { ... }
}
```

练习点：

- `input.split(" ")`
- 识别命令与参数
- 使用 `Map<String, Command>`（可选）



线下任务（结合 Vibe Coding）

- 生成完整地图（3–6 场景）
- 添加 NPC（可继承自 `Character` 或独立类）
- 添加探索事件：

```
You found a chest! (+HealthPotion)
A trap! (-5 HP)
```

Vibe Coding Prompt 示例：

“帮我创建一个 5 个区域的 RPG 文本地图，每个区域的描述、怪物列表、战利品列表。”



Week 3 最终产出

- 世界探索系统
- 多个场景（`Location`）
- 移动命令 `go xxx`

- 简单事件系统（开宝箱 / 遇怪）
- 游戏已经可玩并有“探索感”

Week 4 — 完整化、重构、文档、展示（完成最终项目）

学习目标

- 学习整理系统架构（高内聚 / 低耦合）
- 学习简单设计模式（可选：策略 / 工厂）
- 完成文档（UML / README）
- 小组 Demo 展示（演示游戏流程）

本周模块

1. 项目整合与重构
2. UML & 文档完成
3. 最终展示（Demo Day）

课堂任务（两节课）

◆ 第 1 节：代码重构 & 增强功能

可选增强：

- Factory 自动生成敌人 / 道具
- Strategy Pattern 应用于技能
- 使用 Stream 优化搜索功能

- 添加游戏存档 / 读取

其他改善：

- 把 GameEngine 拆分成更清晰的逻辑（UI / 战斗 / 世界）

◆ 第 2 节：文档与展示准备

- **UML 类图（至少 Class Diagram）**
 - Player / Enemy / Skill / Item / Inventory / Location / GameEngine
- 系统说明文档
- **Demo 展示脚本**
- 完整视频录屏（可选）



线下任务（结合 Vibe Coding）

- 用 AI 帮忙生成 UML class diagram（你可以帮检查是否合理）
- 用 AI 帮忙生成完整 README 模板
- 让 AI 帮忙重写代码使其更 clean
- 扩展剧情、技能、Boss 设定



Week 4 最终产出（最终验收）

- 完整可玩的 Text RPG
- UML 设计文档
- 代码仓库（GitHub 或 zip）
- Demo 演示（录屏或现场）
- 设计报告（由 AI 辅助润色）



总结：4 周项目达成能力

通过这 4 周项目，将掌握：

- 完整的面向对象设计能力（继承 / 多态 / 组合 / 封装 / 接口）
- 熟练使用 **Java Collections**（**List / Map / Set**）
- 基本的设计模式意识（策略、工厂、分层设计）
- 开发一个“能跑起来的系统”的能力
- 文档交付能力（UML / 说明文档 / **Demo**）
- **AI-assisted Development**（**Vibe Coding**）技能

Library Management System – Console Demo 示例

说明:

- > 后面是用户输入
- 其他是程序输出
- 界面全英文

1 启动程序 & 主菜单

```
===== Library Management System
=====
```

```
Welcome to the Library Management System.
Please choose an option:
```

```
1) List all books
2) List all members
3) Search books by title
4) Borrow a book
5) Return a book
6) Show member's borrowed books
0) Exit
```

```
> 1
```

2 查看所有图书 (List all books)

```
--- Book List ---
ID: B001  [Print]  The Hobbit                by J.R.R. Tolkien
(3/ 5 copies)
ID: B002  [Print]  Clean Code                by Robert C.
Martin  (2/ 2 copies)
ID: E100  [E-Book] Java Basics (E-Book)     by Some Author
(unlimited)
ID: B003  [Print]  Algorithms Illustrated    by Jane Doe
(1/ 1 copies)
```

Total books: 4

(Press Enter to continue)

>

程序返回主菜单:

- 1) List all books
- 2) List all members
- 3) Search books by title
- 4) Borrow a book
- 5) Return a book
- 6) Show member's borrowed books
- 0) Exit

> 2

3 查看所有读者 (List all members)

--- Member List ---

ID: 1 Name: Alice

ID: 2 Name: Bob

ID: 3 Name: Charlie

Total members: 3

(Press Enter to continue)

>

回到主菜单:

> 3

4 按标题关键字搜索图书 (Search books by title)

--- Search Books by Title ---

Enter keyword:

> code

Search results for "code":

```
ID: B002  [Print]  Clean Code  by Robert C. Martin  (2/ 2
copies)
```

```
(Press Enter to continue)
```

```
>
```

学生可以看出：搜索是模糊匹配，大小写不敏感。

5 借书 (Borrow a book) — 成功案例

```
> 4
```

```
--- Borrow a Book ---
```

```
Enter member ID:
```

```
> 1
```

```
Enter book ID:
```

```
> B002
```

```
Trying to borrow book B002 for member 1...
```

```
Success!
```

```
Member "Alice" borrowed "Clean Code".
```

```
Remaining copies of "Clean Code": 1/2
```

```
(Press Enter to continue)
```

```
>
```

再看一次书籍列表：

```
> 1
```

```
--- Book List ---
```

```
ID: B001  [Print]  The Hobbit                      by J.R.R. Tolkien
(3/ 5 copies)
```

```
ID: B002  [Print]  Clean Code                      by Robert C.
Martin  (1/ 2 copies)
```

```
ID: E100  [E-Book] Java Basics (E-Book)          by Some Author
(unlimited)
```

```
ID: B003  [Print]  Algorithms Illustrated         by Jane Doe
(1/ 1 copies)
```

「Alice 已经借走 1 本 Clean Code，所以可借数量从 2 变成了 1。」

6 再借同一本书，直到库存用完 (Borrow until full)

我们让 Bob 去借同一本书：

```
> 4
```

```
--- Borrow a Book ---
```

```
Enter member ID:
```

```
> 2
```

```
Enter book ID:
```

```
> B002
```

```
Trying to borrow book B002 for member 2...
```

```
Success!
```

```
Member "Bob" borrowed "Clean Code".
```

```
Remaining copies of "Clean Code": 0/2
```

```
(Press Enter to continue)
```

```
>
```

再让 Charlie 试图借同一本书（这次应该失败）：

```
> 4
```

```
--- Borrow a Book ---
```

```
Enter member ID:
```

```
> 3
```

```
Enter book ID:
```

```
> B002
```

```
Trying to borrow book B002 for member 3...
```

```
Failed: No available copies of "Clean Code".
```

```
(Available: 0/2)
```

```
(Press Enter to continue)
```

```
>
```

这里学生可以直观看到：图书数量 / 借阅状态被系统正确维护了。

7 查看某个用户当前借了什么书 (Show member's borrowed books)

> 6

--- Show Member's Borrowed Books ---

Enter member ID:

> 1

Borrowed books of member "Alice":

1) B002 Clean Code (Borrowed on: 2025-04-01, Due: 2025-04-15)

Total borrowed: 1

(Press Enter to continue)

>

再看 Bob 的:

> 6

Enter member ID:

> 2

Borrowed books of member "Bob":

1) B002 Clean Code (Borrowed on: 2025-04-02, Due: 2025-04-16)

Total borrowed: 1

Charlie 没借到书:

> 6

Enter member ID:

> 3

Borrowed books of member "Charlie":

No active borrow records.

(Press Enter to continue)

>

8 还书 (Return a book) — 成功 + 再看库存变化

现在让 Alice 还掉她借的那本书:

> 5

--- Return a Book ---

Enter member ID:

> 1

Enter book ID:

> B002

Trying to return book B002 for member 1...

Success!

Member "Alice" returned "Clean Code".

Available copies of "Clean Code": 1/2

(Press Enter to continue)

>

查看图书列表:

> 1

--- Book List ---

ID: B001	[Print]	The Hobbit	by J.R.R. Tolkien
			(3/ 5 copies)

ID: B002	[Print]	Clean Code	by Robert C.
			Martin (1/ 2 copies)

ID: E100	[E-Book]	Java Basics (E-Book)	by Some Author
			(unlimited)

ID: B003	[Print]	Algorithms Illustrated	by Jane Doe
			(1/ 1 copies)

再查看 Alice 当前借阅:

```
> 6
Enter member ID:
> 1

Borrowed books of member "Alice":

No active borrow records.

(Press Enter to continue)
>
```

9 一些异常用例示例（学生可以自己实现）

支持这些情况并输出提示信息：

- 借书时输入不存在的 memberId：

```
> 4
--- Borrow a Book ---
Enter member ID:
> 999

Error: Member with ID 999 does not exist.
• 借书时输入不存在的 bookId:
```

```
Enter book ID:
> B999

Error: Book with ID B999 not found.
• 还书时，此用户并未借过该书：
```

```
> 5
--- Return a Book ---
Enter member ID:
> 2
Enter book ID:
> B001

Failed: Member "Bob" has no active borrow record for book
B001.
这些示例都可以作为“进阶要求”或“加分点”。
```


10. 退出系统 (Exit)

> 0

Exiting Library Management System...
Goodbye!

Library Management System – Student Project Document (学生版任务文档)

项目周期：4 周（每周 2 节课 + 课下练习）

你将从零开始设计一个完整的图书管理系统，使用 **Java + 面向对象设计 + 集合框架 + 控制台界面**。

Project Goal (项目目标)

通过本项目，你将完成一个可以在命令行中运行的图书馆管理系统，能够：

- 添加图书与用户
- 借书 / 还书
- 查询与搜索
- 统计与排序
- 使用面向对象设计完成系统模块
- 使用 `List`, `Map`, `ArrayList`, `HashMap`, `Comparator`, `Streams`
- 输出 UML 类图与项目文档

最终，你将演示你的系统并提交完整项目。



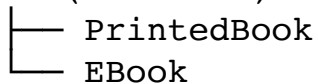
Overall System Structure（系统概览）

你将逐步构建如下模块：

- Book（抽象类）
 - PrintedBook
 - EBook
- Member
- BorrowRecord
- Library（系统核心）
- 控制台界面（Console UI）
- 可选：数据持久化（文件读写）

系统结构示例（简化 UML）：

Book (abstract)



Member

BorrowRecord

Library (contains Maps: id → Book, id → Member)

Console UI (Main)



Week 1 — 核心类设计 & 最基础功能



本周目标

- 学习如何用类建模现实对象（图书、用户）
- 理解抽象类与继承
- 构建 Library 系统基本骨架
- 实现最小可运行版本（添加图书/用户 + 打印列表）

本周任务

✓ 1. 创建 **Book** 抽象类

包含字段：

`id, title, author, totalCopies, availableCopies`

包含方法：

- `boolean canBorrow()`
- `void borrowOne()`
- `void returnOne()`
- `abstract String getInfo()`

✓ 2. 创建两个子类

`PrintedBook`

`EBook`

要求实现 `getInfo()` 方法。

✓ 3. 创建 **Member** 类

包含：

`id, name`

可打印用户信息。

✓ 4. 创建 **Library** 类（初版）

使用：

`List<Book> books`

`List<Member> members`

实现以下方法：

- `addBook(Book b)`
- `addMember(Member m)`
- `listBooks()`

- `listMembers()`

✓ 5. 创建 Console 菜单 (Main.java)

最小原型可运行如下菜单：

- 1) List books
- 2) Add book
- 3) List members
- 0) Exit

Week 2 — 借书 / 还书系统 + 搜索功能

本周目标

- 理解组合 (Library has BorrowRecord)
- 使用 Map 加速查找
- 实现借书与还书的逻辑
- 实现标题搜索

本周任务

✓ 1. 创建 **BorrowRecord**

包含字段：

```
Book book
Member member
LocalDate borrowDate
LocalDate dueDate
boolean returned
```

✓ 2. 将 **Library** 改为 **Map** 结构

```
Map<String, Book> booksById
Map<Integer, Member> membersById
List<BorrowRecord> records
```

✓ 3. 实现借书 / 还书逻辑

```
borrowBook(memberId, bookId)
```

```
returnBook(memberId, bookId)
```

处理各种错误情况：

- 用户不存在
- 图书不存在
- 无可借库存

✓ 4. 搜索功能

使用标题关键字搜索图书：

```
searchByTitle(keyword)
```

✓ 5. Console 菜单加入命令：

3) Borrow book

4) Return book

5) Search book

Week 3 — 排序、统计、文件读写（可选）

本周目标

- 熟练使用 `Collections.sort + Comparator`
- 学习 Java Stream 基本使用
- （可选）掌握文件读写，实现系统数据保存

本周任务

✓ 1. 排序功能

按标题排序所有书：

```
listBooksSortedByTitle()
```

按作者排序：

```
listBooksSortedByAuthor()
```

使用：

```
Collections.sort(list, Comparator.comparing(Book::getTitle))
```

✓ 2. 统计功能

实现以下任意两个：

- 查看某用户目前借阅的书籍
- 查看某本书被借阅的次数
- 找出借阅次数最多的成员

✓ 3. 文件持久化（可选）

将 Book/Member 数据保存到本地文件：

```
saveToFile()
```

```
loadFromFile()
```

格式可用 CSV / 简单文本。

Week 4 — 重构、UML、最终展示

本周目标

- 清理代码、改善结构
- 输出 UML 类图
- 编写 README 文档
- 完成最终项目 Demo 展示

本周任务

✓ 1. 重构（Refactoring）

检查：

- 是否有重复代码？

- 是否可以拆分为多个类/服务？（例如 BookService、MemberService）
- 是否需要封装更多内容？

✓ 2. UML 类图

使用任意工具（PlantUML / Draw.io / 手绘）完成：

必须包含：

- Book（抽象类）
- PrintedBook / EBook
- Member
- BorrowRecord
- Library

✓ 3. README 文档

内容包括：

- 项目描述
- 功能列表
- 如何运行
- 样例输出
- 类结构解释

✓ 4. 最终演示（Demo Day）

演示推荐顺序：

- 1) 列出所有图书
- 2) 添加图书
- 3) 注册用户
- 4) 借书
- 5) 还书
- 6) 搜索功能
- 7) 排序与统计



项目最终交付内容 (Final Deliverables)

- 完整可运行的 Java 控制台图书管理系统
- UML 类图
- README 文档
- 项目代码（打包或上传至 GitHub）
- Demo 展示（现场 or 录屏）



Bonus (选做加分项)

你可以选择实现：

- 书籍分类系统
- 用户等级（普通/管理员）
- 图书推荐系统（基于借阅历史）
- GUI 版本（JavaFX/Swing）
- JSON 数据加载（使用 Gson）