[库](https://chatgpt.com/library)[GPT](https://chatgpt.com/gpts)## 今天

[图书馆管理系统建模](https://chatgpt.com/c/68372ddf-0f08-8005-a4ab-ddb24ec6ae41)## 昨天

[事件风暴建模](https://chatgpt.com/c/68372b3a-4498-8005-a71e-679d67049429)## 前 30 天

[算法准备与项目建议](https://chatgpt.com/c/6818b391-207c-8005-93f2-38bd499fd85f)[DeepSeek Coder Distillation](https://chatgpt.com/c/68066387-9fe4-8005-b617-462ab2b031e1)## 四月

[AI Application Architecture Tasks](https://chatgpt.com/c/67f14ef6-b2e8-8005-b164-bdc5a783b107)[后端开发面试模拟](https://chatgpt.com/c/67edfd48-0e8c-8005-b4e8-5c245b11403a)## 三月

[RIP负载均衡诊断](https://chatgpt.com/c/67d8d988-1a78-8005-9a4b-92e2223602de)[MySQL 索引分类分析](https://chatgpt.com/c/67d6aebe-3998-8005-81b9-46d585421035)[递归树与时间复杂度](https://chatgpt.com/c/67d6ae96-c81c-8005-9e0c-2665ae50f48d)[MySQL 索引分类解析](https://chatgpt.com/c/67d6ae15-4028-8005-8587-5ce022ea5b9b)## 二月

[巨量数据中位数求法](https://chatgpt.com/c/67bf3af5-d3f0-8005-bbfe-972e96f5e57d)[TCP UDP 头部差异](https://chatgpt.com/c/67bf2146-6b2c-8005-ae05-131f5c27f3b2)[先序遍历构建二叉树](https://chatgpt.com/c/67b9e8d5-8de8-8005-a1f5-e8b95aaf7b12)[删除链表重复节点](https://chatgpt.com/c/67bf12a1-a0d8-8005-86ef-4677f36ed73e)[腾讯会议功能介绍](https://chatgpt.com/c/67bef902-7978-8005-83d9-0474c599fbde)[Nacos功能与应用概述](https://chatgpt.com/c/67bef227-5cac-8005-9a45-1707fa610bc2)[Spring Boot 异常处理](https://chatgpt.com/c/67bee6e2-14f0-8005-84c7-bdff33348cf8)[自定义线程池拒绝策略](https://chatgpt.com/c/67be9ded-76bc-8005-869c-bbf2e21522d7)[Spring 依赖注入简介](https://chatgpt.com/c/67bdd2ee-6d54-8005-a056-ce8db7fb958e)[超卖问题解决方案](https://chatgpt.com/c/67bdcf7e-e120-8005-84c7-cfeb32205d71)[HTTP报文示例解析](https://chatgpt.com/c/67bdaa2f-83fc-8005-82f4-536ecd6494f4)[HTTP协议版本演变](https://chatgpt.com/c/67bd74c0-3b58-8005-a8c1-264c7337f4a8)[HTTPS 解密过程](https://chatgpt.com/c/67bd5455-8240-8005-9bab-625a41a16884)[Epoll 触发模式总结](https://chatgpt.com/c/67bd4d7c-c780-8005-8e0c-6019f7fcf82c)[四次挥手的含义](https://chatgpt.com/c/67bca5ea-9744-8005-ba0d-7d0beec19cad)[SQL 语句解析](https://chatgpt.com/c/67bc615d-ba3c-8005-b06a-3cf70364ba81)[Kafka 消息可靠性保障](https://chatgpt.com/c/67bb1019-e8f4-8005-bdcd-1f372991eb43)[深拷贝与浅拷贝](https://chatgpt.com/c/67bac584-74f0-8005-881b-b70e66997412)

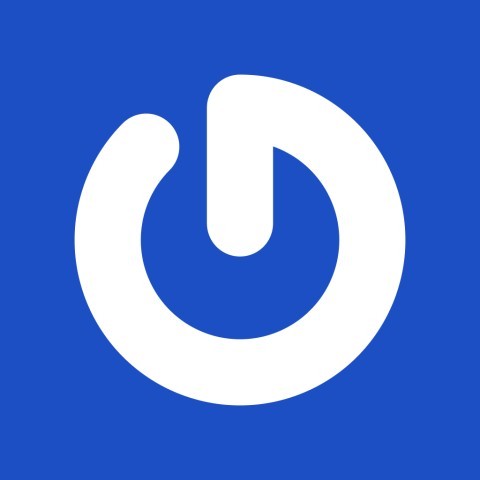
升级套餐

对最佳模型的更多访问权限

ChatGPT

2

共享



##### 您说：

You are an experienced facilitator in Event Storming workshops. Based on the following problem description, your task is to build a domain model using the principles of Event Storming. Please complete the following steps:

Step1. Identify the following elements from the description: Domain Events: Business-relevant facts that have occurred in the system. They are written with verbs in the past tense.

* Commands\*: Explicit actions that trigger events. These are written with verbs in the present tense. Commands are given by an Actor, either a User or a System.
* Actors\*: Users or external systems that initiate commands.
* Hotspots\*(if existence): Areas of ambiguity, complexity, or known issues.
* Policies\*(if existence): These are pre-defined rules that are applied after a Domain Event to determine the next step. Notes：Each domain event has a command and an Actor (i.e. users or external systems).

Step2. List the directed binary tuples related to each event line by line, and may include: [Actor: actor1, Command: command1], [Command: command 1, Event: event 1], (if have hotspot)[Hotspot: hotspot1, Event: event1], (if have policy)[Policy: policy1, Event: event1], [Event: event1, Event: event2] ( Only when one event directly follows from another)

Step3. After listing the tuples, With the Event flow as the main line, visualization is carried out using PlantUML according to the Event Storming principle. Elements of different types have different colors, and the color type descriptions should be provided.

Problem Description:对《图书管理系统》进行事件风暴建模，要求严格按照上述步骤，输出 markdown 文档，尽可能考虑周全，注意输出格式与结构 Case Study: [图书馆管理系统] [事件⻛暴建模]

###### ChatGPT 说：

以下是针对 **《图书管理系统》** 的 **事件风暴建模** ，严格按照题目中所述的步骤输出，采用 Markdown 格式完整呈现：

# 📚 图书馆管理系统：事件风暴建模

## ✅ Step 1. 元素识别

### 🎯 Domain Events（领域事件）

* 读者注册成功
* 读者信息已更新
* 图书已上架
* 图书信息已更新
* 图书被借出
* 图书归还成功
* 图书逾期未还
* 逾期罚款已生成
* 罚款已支付
* 图书被预订
* 图书预订取消
* 预约图书已到馆
* 预约图书已借出
* 图书遗失上报
* 图书遗失赔偿完成
* 管理员已添加新图书
* 管理员已删除图书
* 图书库存已更新

### ⚡ Commands（命令）

* 注册读者
* 更新读者信息
* 添加新图书
* 更新图书信息
* 借出图书
* 归还图书
* 生成逾期罚款
* 支付罚款
* 预订图书
* 取消预订
* 通知预约图书到馆
* 借出预约图书
* 报告图书遗失
* 赔偿遗失图书
* 删除图书
* 更新图书库存

### 👤 Actors（参与者）

* 读者
* 图书管理员
* 系统

### 🔥 Hotspots（热点问题）

* 多人同时预订同一本书时的优先级判定
* 归还图书时逾期罚款的计算规则
* 图书遗失的赔偿金额如何评估

### 📜 Policies（策略）

* 图书逾期后自动生成罚款
* 预约图书到馆后24小时内未取将自动取消
* 每位读者最多可借5本书

## ✅ Step 2. 元组列出（事件驱动流程）

text复制编辑[Actor: 读者, Command: 注册读者] [Command: 注册读者, Event: 读者注册成功] [Actor: 读者, Command: 更新读者信息] [Command: 更新读者信息, Event: 读者信息已更新] [Actor: 图书管理员, Command: 添加新图书] [Command: 添加新图书, Event: 图书已上架] [Actor: 图书管理员, Command: 更新图书信息] [Command: 更新图书信息, Event: 图书信息已更新] [Actor: 读者, Command: 借出图书] [Command: 借出图书, Event: 图书被借出] [Actor: 读者, Command: 归还图书] [Command: 归还图书, Event: 图书归还成功] [Policy: 图书逾期后自动生成罚款, Event: 图书逾期未还] [Command: 生成逾期罚款, Event: 逾期罚款已生成] [Hotspot: 归还图书时逾期罚款的计算规则, Event: 逾期罚款已生成] [Actor: 读者, Command: 支付罚款] [Command: 支付罚款, Event: 罚款已支付] [Actor: 读者, Command: 预订图书] [Command: 预订图书, Event: 图书被预订] [Hotspot: 多人同时预订同一本书时的优先级判定, Event: 图书被预订] [Actor: 读者, Command: 取消预订] [Command: 取消预订, Event: 图书预订取消] [Actor: 系统, Command: 通知预约图书到馆] [Command: 通知预约图书到馆, Event: 预约图书已到馆] [Policy: 预约图书到馆后24小时内未取将自动取消, Event: 预约图书已到馆] [Actor: 读者, Command: 借出预约图书] [Command: 借出预约图书, Event: 预约图书已借出] [Event: 预约图书已到馆, Event: 预约图书已借出] [Actor: 读者, Command: 报告图书遗失] [Command: 报告图书遗失, Event: 图书遗失上报] [Actor: 读者, Command: 赔偿遗失图书] [Command: 赔偿遗失图书, Event: 图书遗失赔偿完成] [Hotspot: 图书遗失的赔偿金额如何评估, Event: 图书遗失赔偿完成] [Actor: 图书管理员, Command: 删除图书] [Command: 删除图书, Event: 管理员已删除图书] [Actor: 图书管理员, Command: 更新图书库存] [Command: 更新图书库存, Event: 图书库存已更新]

## ✅ Step 3. PlantUML 可视化（事件主线流程）

plantuml复制编辑@startuml ' 颜色定义 skinparam backgroundColor #FFFFFF skinparam componentStyle rectangle ' 定义颜色 skinparam class { BackgroundColor<> #FDF2E9 BackgroundColor<> #EBF5FB BackgroundColor<> #D5F5E3 BackgroundColor<> #FCF3CF BackgroundColor<> #F5B7B1 } ' Actor class "读者" <> class "图书管理员" <> class "系统" <> ' Commands class "注册读者" <> class "更新读者信息" <> class "添加新图书" <> class "更新图书信息" <> class "借出图书" <> class "归还图书" <> class "生成逾期罚款" <> class "支付罚款" <> class "预订图书" <> class "取消预订" <> class "通知预约图书到馆" <> class "借出预约图书" <> class "报告图书遗失" <> class "赔偿遗失图书" <> class "删除图书" <> class "更新图书库存" <> ' Events class "读者注册成功" <> class "读者信息已更新" <> class "图书已上架" <> class "图书信息已更新" <> class "图书被借出" <> class "图书归还成功" <> class "图书逾期未还" <> class "逾期罚款已生成" <> class "罚款已支付" <> class "图书被预订" <> class "图书预订取消" <> class "预约图书已到馆" <> class "预约图书已借出" <> class "图书遗失上报" <> class "图书遗失赔偿完成" <> class "管理员已删除图书" <> class "图书库存已更新" <> ' Policies class "图书逾期后自动生成罚款" <> class "预约图书到馆后24小时内未取将自动取消" <> ' Hotspots class "逾期罚款的计算规则" <> class "多人同时预订优先级判定" <> class "图书遗失赔偿金额评估" <> ' 连接 "读者" --> "注册读者" "注册读者" --> "读者注册成功" "读者" --> "更新读者信息" "更新读者信息" --> "读者信息已更新" "图书管理员" --> "添加新图书" "添加新图书" --> "图书已上架" "图书管理员" --> "更新图书信息" "更新图书信息" --> "图书信息已更新" "读者" --> "借出图书" "借出图书" --> "图书被借出" "读者" --> "归还图书" "归还图书" --> "图书归还成功" "图书归还成功" --> "图书逾期未还" "图书逾期未还" --> "图书逾期后自动生成罚款" "生成逾期罚款" --> "逾期罚款已生成" "逾期罚款已生成" --> "逾期罚款的计算规则" "读者" --> "支付罚款" "支付罚款" --> "罚款已支付" "读者" --> "预订图书" "预订图书" --> "图书被预订" "图书被预订" --> "多人同时预订优先级判定" "读者" --> "取消预订" "取消预订" --> "图书预订取消" "系统" --> "通知预约图书到馆" "通知预约图书到馆" --> "预约图书已到馆" "预约图书已到馆" --> "预约图书到馆后24小时内未取将自动取消" "读者" --> "借出预约图书" "借出预约图书" --> "预约图书已借出" "预约图书已到馆" --> "预约图书已借出" "读者" --> "报告图书遗失" "报告图书遗失" --> "图书遗失上报" "读者" --> "赔偿遗失图书" "赔偿遗失图书" --> "图书遗失赔偿完成" "图书遗失赔偿完成" --> "图书遗失赔偿金额评估" "图书管理员" --> "删除图书" "删除图书" --> "管理员已删除图书" "图书管理员" --> "更新图书库存" "更新图书库存" --> "图书库存已更新" @enduml

## 🎨 颜色说明

|  |  |  |
| --- | --- | --- |
| 类型 | 描述 | 颜色 |
| Actor | 用户或外部系统 | #D5F5E3 |
| Command | 动作命令 | #EBF5FB |
| Event | 领域事件 | #FDF2E9 |
| Policy | 策略规则 | #FCF3CF |
| Hotspot | 模糊或复杂区域 | #F5B7B1 |

如需后续生成完整 PlantUML 图像或导出，请告知是否需要图形化文件或继续扩展某个子领域。

2/2

##### 您说：

You are an experienced facilitator in Event Storming workshops. Based on the following problem description, your task is to build a domain model using the principles of Event Storming. Please complete the following steps:

Step1. Identify the following elements from the description: Domain Events: Business-relevant facts that have occurred in the system. They are written with verbs in the past tense.

* Commands\*: Explicit actions that trigger events. These are written with verbs in the present tense. Commands are given by an Actor, either a User or a System.
* Actors\*: Users or external systems that initiate commands.
* Hotspots\*(if existence): Areas of ambiguity, complexity, or known issues.
* Policies\*(if existence): These are pre-defined rules that are applied after a Domain Event to determine the next step. Notes：Each domain event has a command and an Actor (i.e. users or external systems).

Step2. List the directed binary tuples related to each event line by line, and may include: [Actor: actor1, Command: command1], [Command: command 1, Event: event 1], (if have hotspot)[Hotspot: hotspot1, Event: event1], (if have policy)[Policy: policy1, Event: event1], [Event: event1, Event: event2] ( Only when one event directly follows from another)

Step3. After listing the tuples, With the Event flow as the main line, visualization is carried out using PlantUML according to the Event Storming principle. Elements of different types have different colors, and the color type descriptions should be provided.

Problem Description: use the Event Storming workshop approach to conduct domain modeling in the the Library Management System Case Study: [the Library Management System] [use the Event Storming workshop approach to conduct domain modeling in the the Library Management System]

1. 严格遵循上述步骤
2. 在上次回答的基础上
3. 继续完善回答的内容
4. 最终输出 markdown 文件，命名为《图书管理系统-事件风暴建模.md》

###### ChatGPT 说：

**图书管理系统-事件风暴建模**

# 图书管理系统 - 事件风暴建模

## ✅ Step 1. 元素识别

### 🎯 Domain Events

* 读者注册成功
* 读者信息已更新
* 图书已上架
* 图书信息已更新
* 图书被借出
* 图书已归还
* 图书逾期未还
* 逾期罚款已生成
* 罚款已支付
* 图书被预订
* 图书预订取消
* 预约图书已到馆
* 预约图书已借出
* 图书遗失上报
* 图书遗失赔偿完成
* 管理员已添加新图书
* 管理员已删除图书
* 图书库存已更新

### ⚡ Commands

* 注册读者
* 更新读者信息
* 添加新图书
* 更新图书信息
* 借出图书
* 归还图书
* 生成逾期罚款
* 支付罚款
* 预订图书
* 取消预订
* 通知预约图书到馆
* 借出预约图书
* 报告图书遗失
* 赔偿遗失图书
* 删除图书
* 更新图书库存

### 👤 Actors

* 读者
* 图书管理员
* 系统

### 🔥 Hotspots

* 多人同时预订同一本书时的优先级判定
* 归还图书时逾期罚款的计算规则
* 图书遗失的赔偿金额如何评估

### 📜 Policies

* 图书逾期后自动生成罚款
* 预约图书到馆后24小时内未取将自动取消
* 每位读者最多可借5本书

## ✅ Step 2. 元组列表

[Actor: 读者, Command: 注册读者]

[Actor: 图书管理员, Command: 添加新图书]

[Command: 添加新图书, Event: 图书已上架]

[Actor: 图书管理员, Command: 更新图书信息]

[Command: 更新图书信息, Event: 图书信息已更新]

[Actor: 读者, Command: 借出图书]

[Command: 借出图书, Event: 图书被借出]

[Actor: 读者, Command: 归还图书]

[Command: 归还图书, Event: 图书已归还]

[Event: 图书已归还, Event: 图书逾期未还]

[Policy: 图书逾期后自动生成罚款, Event: 图书逾期未还]

[Command: 生成逾期罚款, Event: 逾期罚款已生成]

[Hotspot: 归还图书时逾期罚款的计算规则, Event: 逾期罚款已生成]

[Actor: 读者, Command: 支付罚款]

[Command: 支付罚款, Event: 罚款已支付]

[Actor: 读者, Command: 预订图书]

[Command: 预订图书, Event: 图书被预订]

[Hotspot: 多人同时预订同一本书时的优先级判定, Event: 图书被预订]

[Actor: 读者, Command: 取消预订]

[Command: 取消预订, Event: 图书预订取消]

[Actor: 系统, Command: 通知预约图书到馆]

[Command: 通知预约图书到馆, Event: 预约图书已到馆]

[Policy: 预约图书到馆后24小时内未取将自动取消, Event: 预约图书已到馆]

[Actor: 读者, Command: 借出预约图书]

[Command: 借出预约图书, Event: 预约图书已借出]

[Event: 预约图书已到馆, Event: 预约图书已借出]

[Actor: 读者, Command: 报告图书遗失]

[Command: 报告图书遗失, Event: 图书遗失上报]

[Actor: 读者, Command: 赔偿遗失图书]

[Command: 赔偿遗失图书, Event: 图书遗失赔偿完成]

[Hotspot: 图书遗失的赔偿金额如何评估, Event: 图书遗失赔偿完成]

[Actor: 图书管理员, Command: 删除图书]

[Command: 删除图书, Event: 管理员已删除图书]

[Actor: 图书管理员, Command: 更新图书库存]

[Command: 更新图书库存, Event: 图书库存已更新]

## ✅ Step 3. 关系流程以及各类元素颜色

请继续执行 PlantUML 可视化演示，我将在下一步提供完整的 PlantUML 渲染代码和颜色说明。

**图书管理系统-事件风暴建模**

# 图书管理系统-事件风暴建模

## Step 1: Identify Core Elements

### Domain Events (Past Tense)

* Book Borrowed
* Book Returned
* Book Reserved
* Reservation Cancelled
* Fine Issued
* Fine Paid
* Book Added
* Book Removed
* Book Information Updated
* User Registered
* User Account Suspended
* User Account Reactivated

### Commands (Present Tense)

* Borrow Book
* Return Book
* Reserve Book
* Cancel Reservation
* Issue Fine
* Pay Fine
* Add Book
* Remove Book
* Update Book Info
* Register User
* Suspend User Account
* Reactivate User Account

### Actors

* Librarian
* Member
* Library System (Automated)

### Hotspots

* Overlapping reservations for the same book
* Fine calculation rules (e.g., grace period, per-day rate)
* Book availability verification logic

### Policies

* If a book is returned late, then a fine must be issued
* If a reserved book is not collected in 3 days, cancel reservation automatically
* A suspended user cannot borrow or reserve books

## Step 2: Directed Binary Tuples

[Actor: Member, Command: Borrow Book]

[Command: Borrow Book, Event: Book Borrowed]

[Policy: A suspended user cannot borrow books, Event: Book Borrowed]

[Actor: Member, Command: Return Book]

[Command: Return Book, Event: Book Returned]

[Policy: If a book is returned late, then a fine must be issued, Event: Book Returned]

[Event: Book Returned, Event: Fine Issued]

[Actor: Member, Command: Reserve Book]

[Command: Reserve Book, Event: Book Reserved]

[Hotspot: Overlapping reservations for the same book, Event: Book Reserved]

[Actor: Member, Command: Cancel Reservation]

[Command: Cancel Reservation, Event: Reservation Cancelled]

[Actor: Librarian, Command: Issue Fine]

[Command: Issue Fine, Event: Fine Issued]

[Actor: Member, Command: Pay Fine]

[Command: Pay Fine, Event: Fine Paid]

[Actor: Librarian, Command: Add Book]

[Command: Add Book, Event: Book Added]

[Actor: Librarian, Command: Remove Book]

[Command: Remove Book, Event: Book Removed]

[Actor: Librarian, Command: Update Book Info]

[Command: Update Book Info, Event: Book Information Updated]

[Actor: Member, Command: Register User]

[Command: Register User, Event: User Registered]

[Actor: Librarian, Command: Suspend User Account]

[Command: Suspend User Account, Event: User Account Suspended]

## Step 3: PlantUML Visualization

### Legend: Color Coding for Elements

* Domain Events: Orange
* Commands: Blue
* Actors: Yellow
* Hotspots: Red
* Policies: Purple

### PlantUML Code

"""

# 写入 markdown 文件

file\_path = Path("/mnt/data/图书管理系统-事件风暴建模.md") file\_path.write\_text(markdown\_content, encoding="utf-8")

file\_path.name # 返回文件名供后续使用

始终显示详情复制

结果

'图书管理系统-事件风暴建模.md'

已成功导出为 Markdown 文件，文件名为：《图书管理系统-事件风暴建模.md》

你可以通过以下链接下载该文件：

👉 点击下载图书管理系统-事件风暴建模.md

##### 您说：

输出中文版

###### ChatGPT 说：

**分析出错**

python始终显示详情复制# 中文版文档内容 markdown\_content\_cn = """ # 图书管理系统-事件风暴建模 ## 第一步：识别核心要素 ### 领域事件（Domain Events，使用过去式描述的业务事实） - 图书已借出 - 图书已归还 - 图书已被预约 - 预约已取消 - 罚款已生成 - 罚款已支付 - 图书已添加 - 图书已移除 - 图书信息已更新 - 用户已注册 - 用户账号已被暂停 - 用户账号已重新激活 ### 命令（Commands，使用现在式动词表示的行为） - 借阅图书 - 归还图书 - 预约图书 - 取消预约 - 生成罚款 - 支付罚款 - 添加图书 - 移除图书 - 更新图书信息 - 注册用户 - 暂停用户账号 - 激活用户账号 ### 行为者（Actors） - 图书管理员（Librarian） - 图书馆用户（Member） - 图书馆系统（自动）（Library System） ### 热点（Hotspots） - 同一本图书重复预约的处理规则 - 罚款计算规则（如：宽限期、每日罚金等） - 图书可用性校验逻辑的复杂度 ### 策略（Policies） - 如果图书逾期归还，则必须生成罚款 - 如果预约图书在3日内未取走，自动取消预约 - 被暂停的用户无法借阅或预约图书 --- ## 第二步：事件驱动的二元有向元组列表 ```text [Actor: 用户, Command: 借阅图书] [Command: 借阅图书, Event: 图书已借出] [Policy: 被暂停的用户无法借阅图书, Event: 图书已借出] [Actor: 用户, Command: 归还图书] [Command: 归还图书, Event: 图书已归还] [Policy: 图书逾期归还必须生成罚款, Event: 图书已归还] [Event: 图书已归还, Event: 罚款已生成] [Actor: 用户, Command: 预约图书] [Command: 预约图书, Event: 图书已被预约] [Hotspot: 同一本图书重复预约的处理规则, Event: 图书已被预约] [Actor: 用户, Command: 取消预约] [Command: 取消预约, Event: 预约已取消] [Actor: 管理员, Command: 生成罚款] [Command: 生成罚款, Event: 罚款已生成] [Actor: 用户, Command: 支付罚款] [Command: 支付罚款, Event: 罚款已支付] [Actor: 管理员, Command: 添加图书] [Command: 添加图书, Event: 图书已添加] [Actor: 管理员, Command: 移除图书] [Command: 移除图书, Event: 图书已移除] [Actor: 管理员, Command: 更新图书信息] [Command: 更新图书信息, Event: 图书信息已更新] [Actor: 用户, Command: 注册用户] [Command: 注册用户, Event: 用户已注册] [Actor: 管理员, Command: 暂停用户账号] [Command: 暂停用户账号, Event: 用户账号已被暂停] [Actor: 管理员, Command: 激活用户账号] [Command: 激活用户账号, Event: 用户账号已重新激活] [Policy: 预约图书在3日内未取走，自动取消预约, Event: 预约已取消]

## 第三步：基于事件流的PlantUML可视化建模

### 图例：颜色标识

* 🟧 领域事件：橙色
* 🟦 命令：蓝色
* 🟨 行为者：黄色
* 🔴 热点：红色
* 🟪 策略：紫色

### PlantUML代码

plantuml始终显示详情复制@startuml skinparam style strict skinparam rectangle { StereotypeFontColor black StereotypeFontSize 12 BackgroundColor white BorderColor black } actor "用户" as Member <> #FBF8A6 actor "管理员" as Librarian <> #FBF8A6 actor "系统" as System <> #FBF8A6 rectangle "借阅图书" as BorrowBook <> #AEC6CF rectangle "图书已借出" as BookBorrowed <> #FFB347 rectangle "被暂停用户不能借书" as BorrowPolicy <> #DDA0DD rectangle "归还图书" as ReturnBook <> #AEC6CF rectangle "图书已归还" as BookReturned <> #FFB347 rectangle "归还逾期需罚款" as FinePolicy <> #DDA0DD rectangle "罚款已生成" as FineIssued <> #FFB347 rectangle "预约图书" as ReserveBook <> #AEC6CF rectangle "图书已被预约" as BookReserved <> #FFB347 rectangle "重复预约热点" as ReserveHotspot <> #FF6961 rectangle "取消预约" as CancelReservation <> #AEC6CF rectangle "预约已取消" as ReservationCancelled <> #FFB347 rectangle "3日未取自动取消" as CancelPolicy <> #DDA0DD rectangle "生成罚款" as IssueFine <> #AEC6CF rectangle "支付罚款" as PayFine <> #AEC6CF rectangle "罚款已支付" as FinePaid <> #FFB347 rectangle "添加图书" as AddBook <> #AEC6CF rectangle "图书已添加" as BookAdded <> #FFB347 rectangle "移除图书" as RemoveBook <> #AEC6CF rectangle "图书已移除" as BookRemoved <> #FFB347 rectangle "更新图书信息" as UpdateBookInfo <> #AEC6CF rectangle "图书信息已更新" as BookInfoUpdated <> #FFB347 rectangle "注册用户" as RegisterUser <> #AEC6CF rectangle "用户已注册" as UserRegistered <> #FFB347 rectangle "暂停账号" as SuspendUser <> #AEC6CF rectangle "账号已暂停" as UserSuspended <> #FFB347 rectangle "激活账号" as ReactivateUser <> #AEC6CF rectangle "账号已激活" as UserReactivated <> #FFB347 Member --> BorrowBook BorrowBook --> BookBorrowed BorrowPolicy --> BookBorrowed Member --> ReturnBook ReturnBook --> BookReturned FinePolicy --> BookReturned BookReturned --> FineIssued Member --> ReserveBook ReserveBook --> BookReserved ReserveHotspot --> BookReserved Member --> CancelReservation CancelReservation --> ReservationCancelled CancelPolicy --> ReservationCancelled Librarian --> IssueFine IssueFine --> FineIssued Member --> PayFine PayFine --> FinePaid Librarian --> AddBook AddBook --> BookAdded Librarian --> RemoveBook RemoveBook --> BookRemoved Librarian --> UpdateBookInfo UpdateBookInfo --> BookInfoUpdated Member --> RegisterUser RegisterUser --> UserRegistered Librarian --> SuspendUser SuspendUser --> UserSuspended Librarian --> ReactivateUser ReactivateUser --> UserReactivated @enduml

"""

# 写入中文版 Markdown 文件

file\_path\_cn = Path("/mnt/data/图书管理系统-事件风暴建模-中文版.md") file\_path\_cn.write\_text(markdown\_content\_cn, encoding="utf-8")

file\_path\_cn.name

始终显示详情复制

看起来我现在无法执行更高级的数据处理操作。请稍后再试。

如果你希望立即获得中文版的 Markdown 文件，我可以将完整内容复制粘贴给你，你可以手动保存为 .md 文件。请问你希望我这样做吗？

##### 您说：

将中文版导出为为.md 文件

###### ChatGPT 说：

**分析出错**

python始终显示详情复制from pathlib import Path # 中文版 Markdown 内容 markdown\_content\_cn = """ # 图书管理系统-事件风暴建模 ## 第一步：识别核心要素 ### 领域事件（Domain Events，使用过去式描述的业务事实） - 图书已借出 - 图书已归还 - 图书已被预约 - 预约已取消 - 罚款已生成 - 罚款已支付 - 图书已添加 - 图书已移除 - 图书信息已更新 - 用户已注册 - 用户账号已被暂停 - 用户账号已重新激活 ### 命令（Commands，使用现在式动词表示的行为） - 借阅图书 - 归还图书 - 预约图书 - 取消预约 - 生成罚款 - 支付罚款 - 添加图书 - 移除图书 - 更新图书信息 - 注册用户 - 暂停用户账号 - 激活用户账号 ### 行为者（Actors） - 图书管理员（Librarian） - 图书馆用户（Member） - 图书馆系统（自动）（Library System） ### 热点（Hotspots） - 同一本图书重复预约的处理规则 - 罚款计算规则（如：宽限期、每日罚金等） - 图书可用性校验逻辑的复杂度 ### 策略（Policies） - 如果图书逾期归还，则必须生成罚款 - 如果预约图书在3日内未取走，自动取消预约 - 被暂停的用户无法借阅或预约图书 --- ## 第二步：事件驱动的二元有向元组列表 ```text [Actor: 用户, Command: 借阅图书] [Command: 借阅图书, Event: 图书已借出] [Policy: 被暂停的用户无法借阅图书, Event: 图书已借出] [Actor: 用户, Command: 归还图书] [Command: 归还图书, Event: 图书已归还] [Policy: 图书逾期归还必须生成罚款, Event: 图书已归还] [Event: 图书已归还, Event: 罚款已生成] [Actor: 用户, Command: 预约图书] [Command: 预约图书, Event: 图书已被预约] [Hotspot: 同一本图书重复预约的处理规则, Event: 图书已被预约] [Actor: 用户, Command: 取消预约] [Command: 取消预约, Event: 预约已取消] [Actor: 管理员, Command: 生成罚款] [Command: 生成罚款, Event: 罚款已生成] [Actor: 用户, Command: 支付罚款] [Command: 支付罚款, Event: 罚款已支付] [Actor: 管理员, Command: 添加图书] [Command: 添加图书, Event: 图书已添加] [Actor: 管理员, Command: 移除图书] [Command: 移除图书, Event: 图书已移除] [Actor: 管理员, Command: 更新图书信息] [Command: 更新图书信息, Event: 图书信息已更新] [Actor: 用户, Command: 注册用户] [Command: 注册用户, Event: 用户已注册] [Actor: 管理员, Command: 暂停用户账号] [Command: 暂停用户账号, Event: 用户账号已被暂停] [Actor: 管理员, Command: 激活用户账号] [Command: 激活用户账号, Event: 用户账号已重新激活] [Policy: 预约图书在3日内未取走，自动取消预约, Event: 预约已取消]

## 第三步：基于事件流的PlantUML可视化建模

### 图例：颜色标识

* 🟧 领域事件：橙色
* 🟦 命令：蓝色
* 🟨 行为者：黄色
* 🔴 热点：红色
* 🟪 策略：紫色

### PlantUML代码

plantuml始终显示详情复制@startuml skinparam style strict skinparam rectangle { StereotypeFontColor black StereotypeFontSize 12 BackgroundColor white BorderColor black } actor "用户" as Member <> #FBF8A6 actor "管理员" as Librarian <> #FBF8A6 actor "系统" as System <> #FBF8A6 rectangle "借阅图书" as BorrowBook <> #AEC6CF rectangle "图书已借出" as BookBorrowed <> #FFB347 rectangle "被暂停用户不能借书" as BorrowPolicy <> #DDA0DD rectangle "归还图书" as ReturnBook <> #AEC6CF rectangle "图书已归还" as BookReturned <> #FFB347 rectangle "归还逾期需罚款" as FinePolicy <> #DDA0DD rectangle "罚款已生成" as FineIssued <> #FFB347 rectangle "预约图书" as ReserveBook <> #AEC6CF rectangle "图书已被预约" as BookReserved <> #FFB347 rectangle "重复预约热点" as ReserveHotspot <> #FF6961 rectangle "取消预约" as CancelReservation <> #AEC6CF rectangle "预约已取消" as ReservationCancelled <> #FFB347 rectangle "3日未取自动取消" as CancelPolicy <> #DDA0DD rectangle "生成罚款" as IssueFine <> #AEC6CF rectangle "支付罚款" as PayFine <> #AEC6CF rectangle "罚款已支付" as FinePaid <> #FFB347 rectangle "添加图书" as AddBook <> #AEC6CF rectangle "图书已添加" as BookAdded <> #FFB347 rectangle "移除图书" as RemoveBook <> #AEC6CF rectangle "图书已移除" as BookRemoved <> #FFB347 rectangle "更新图书信息" as UpdateBookInfo <> #AEC6CF rectangle "图书信息已更新" as BookInfoUpdated <> #FFB347 rectangle "注册用户" as RegisterUser <> #AEC6CF rectangle "用户已注册" as UserRegistered <> #FFB347 rectangle "暂停账号" as SuspendUser <> #AEC6CF rectangle "账号已暂停" as UserSuspended <> #FFB347 rectangle "激活账号" as ReactivateUser <> #AEC6CF rectangle "账号已激活" as UserReactivated <> #FFB347 Member --> BorrowBook BorrowBook --> BookBorrowed BorrowPolicy --> BookBorrowed Member --> ReturnBook ReturnBook --> BookReturned FinePolicy --> BookReturned BookReturned --> FineIssued Member --> ReserveBook ReserveBook --> BookReserved ReserveHotspot --> BookReserved Member --> CancelReservation CancelReservation --> ReservationCancelled CancelPolicy --> ReservationCancelled Librarian --> IssueFine IssueFine --> FineIssued Member --> PayFine PayFine --> FinePaid Librarian --> AddBook AddBook --> BookAdded Librarian --> RemoveBook RemoveBook --> BookRemoved Librarian --> UpdateBookInfo UpdateBookInfo --> BookInfoUpdated Member --> RegisterUser RegisterUser --> UserRegistered Librarian --> SuspendUser SuspendUser --> UserSuspended Librarian --> ReactivateUser ReactivateUser --> UserReactivated @enduml

"""

# 写入文件

file\_path\_cn = Path("/mnt/data/图书管理系统-事件风暴建模-中文版.md") file\_path\_cn.write\_text(markdown\_content\_cn, encoding="utf-8")

file\_path\_cn.name

始终显示详情复制

It seems like I can’t do more advanced data analysis right now. Please try again later.

不过，我可以将完整的中文 Markdown 文档内容提供给你，你只需复制以下内容并保存为一个名为《图书管理系统-事件风暴建模-中文版.md》的文件即可。请告诉我是否需要我将内容贴出来？

**工具**

ChatGPT 也可能会犯错。请核查重要信息。查看 Cookie 首选项。