投递附件 01 ｜ 徐强 ｜ 应聘：社区运营 · 小山学堂学习助教

学习笔记：我把《Alice 工程方法论》拆成了 18 讲私教课

原书：洛小山《Alice 方法论》（github.com/itshen/Alice_methodology，本地离线阅读） ｜ 产出时间：2026.10 ｜ 全部内容为我本人拆解、命题与写作

> **一句话说明：**我读完了这本书，然后没有停在「读过」——我把它重做了一遍：拆成 18 讲、每讲写讲义与作业、排成 21 天计划、用打卡表推进、设三关验收，最后做了一个可运行的结课项目。这份附件就是那套东西的目录与样例。

18 讲	15.2 万字	7.6 万字	43 个文件
讲义全部完成	讲义	作业	计划·打卡·速查卡

为什么把它当附件：岗位要求「把讲不清的知识点整理成图解、笔记或短视频，答过的问题沉淀成 FAQ」。这套材料证明的不是我会读书，而是**我具备把一整套知识拆成可教、可练、可验收的课程的能力**——而这正是助教每天在做的事。

目录

节	内容	说明
一	18 讲总表	每讲一句话 + 一句口诀，可直接改成社群打卡卡片或考前速刷表
二	教法样例	以第 01 讲为例，展示一次完整备课：结论先行 → 生活比喻 → 对照表 → 自测与作业
三	学习运营闭环	21 天计划表、三关验收、复盘模板、三条防弃课规则（开营 / 打卡 / 盯卡点的缩小版）
四	结课项目	「迷你 Alice」的分层架构与 6 个交付物，含可运行的 Python 实现
五	附录速查卡	十二范式一页纸节选 + 我能直接给社群用的四样东西

阅读建议：时间紧可只看「一」和「三」；想判断我怎么讲一课，看「二」；想看动手能力，看「四」。

一、课程结构：18 讲总表（每讲一句话 + 口诀）

讲	主题	一句话	口诀
01	Agent 难在哪	模型负责聪明，框架负责靠谱；复杂度主要在状态管理，其次才是模型能力	循环只是壳，状态才是魂
02	五大设计哲学	五条哲学是五把裁判尺子，吵架时拿出来量	稳接口、清状态、分内外、可替换、留痕迹
03	系统地图	地图不是画给老板看的，是故障定位器 + 调用权限表	上层指挥，下层干活；同级平调，隔层绕行
04	主循环	不是「调模型 + 跑工具」，是一圈被六道工序保护的「想—做—记—再想」	先准备，后循环；四道刹车
05	工具系统	工具的价值不在「能执行」，而在「能声明」——让框架敢并行、敢放权	函数只会干，契约才敢排
06	上下文管理	按紧张程度分级加力：能用规则删的绝不问 LLM，要问 LLM 的绝不碰近期原文	先删过程、再去重、后摘要、末重写
07	记忆系统	不是找最强的库，而是给每类信息配最合适的住处	桌面、家规、档案、账房、三张卡
08	多 Agent	不是让 AI 变聪明，是把「装不下」和「只能串行」拆开解决；综合不能外包	先分型，再分人；派的活，自己收
09	权限系统	把「信任」拆成可配置的档位、可读的规则、可追溯的记录	默认关门，显式开门；拦一次，就记一笔
10	MCP 协议	把「给 Agent 加能力」从改代码变成插设备——统一形状、带前缀、懒连接、全都有上限	插上先问会什么，名字带前缀才不撞
11	Skill 系统	补「工具」和「Prompt」之间的空白：工具给手，Skill 给流程	工具管能做，Skill 管怎么做
12	自进化	难点不是「让它能改」，而是「让它改不坏」	能改的划层，不能改的砌墙
13	模型路由	先定义一套所有厂商都能翻译的事件模型，把方言全关进适配器	一个插座，一排转头
14	安全体系	先分级、再上锁：只锁保险箱，不锁整栋楼	口令磨成钥，钥匙不落盘
15	可观测性	不是上线后补的日志，是设计时就埋好的三本账	记录仪看因果，仪表盘看趋势
16	Prompt 工程	不是「把话说全」，是把不变的和会变的分开放	稳定在前，动态在后
17	十二范式	12 条不是技巧，是 12 个「给状态划边界」的位置	契约·事件·分层·防环·权限·懒载……
18	活人感设计	不是加一层可爱的皮，是几万次交互里始终是同一个人	设定要细，身份要锁

完整版见《附录 D ｜ 全课程一句话总结（考前 30 分钟速刷）》，共 5 节：一句话表、口诀表、数字清单、高频追问、口头复述题。

二、教法样例：第 01 讲怎么讲「Agent 到底难在哪」

1) 开场先给结论和问题，不讲定义

> 整本书的「为什么」。作者一上来就怼了一句行业名言——Anthropic 说 Agent 就是「一个简单的 while 循环」。作者说：这句话是对的，**但这是 1% 的真相。**
>
> 学完这讲，你能回答三个问题：① 为什么「能跑」的 Agent 和「靠谱」的 Agent 是两码事？② 一个 Agent 产品真正难的地方，到底是哪 6 个？③ 模型越来越强了，为什么还非得「做工程」？

2) 再用一个生活比喻承载抽象概念（讲义原文节选）

> 假设你要开一家餐厅。你请了一位**米其林大厨**（手艺顶级）。但是餐厅是这样的：客人点菜靠喊，后厨听不清，上错菜；冰箱从来不盘点，客人点「鱼香肉丝」发现肉没了；三个厨师同时炒一道菜，互相把锅端走；谁都能进库房，贵重食材说没就没；出了事没人知道是哪一步错的，只能骂服务员。
>
> **结果：餐厅必倒闭。**而且客人吃完会说「这菜其实挺好吃啊，怎么店没了？」这就是 Agent 产品的现状。

餐厅里	Agent 里
米其林大厨	大模型（LLM，比如 GPT、Claude）
点菜靠喊 / 上错菜	上下文管理混乱，信息传不到位
冰箱不盘点	记忆系统缺失，说过的话转头就忘
三个厨师抢一口锅	多 Agent 协作没有边界
谁都能进库房	权限系统没做，工具随便调
出事只能骂服务员	可观测性缺失，查不到是哪一步的锅

3) 每讲结尾固定三件套：自测 5 问 + A/B/C 三组作业 + 一句话结论

- **A 组（选择/填空）**：概念是否认对了，用来抓「以为自己懂了」。
- **B 组（动手）**：画系统地图、写伪代码、填契约表，必须产出东西。
- **C 组（复述）**：用自己的话讲一遍，不许念原文。（打卡表里写死了规则：作业 C 复述栏必须打勾，其他栏可以有欠账，这一栏不行。）

这样设计的原因：学员「听懂」和「会讲」差了整整一个层级，而助教要做的就是把中间这段路铺出来。

三、学习运营闭环：计划、打卡、验收、复盘（我自己在跑的那一套）

1) 21 天计划表（节选）

天	内容	讲义	作业A	作业B	C 复述	今天最大的收获（一句话）
D1	第01讲 序章	☐	☐	☐	☐	
D2	第02讲 五大哲学	☐	☐	☐	☐	
D3	第03讲 系统地图	☐	☐	☐	☐	
D4	第04讲 主循环	☐	☐	☐	☐	
D5	第05讲 工具系统	☐	☐	☐	☐	
D6	第06讲 上下文管理	☐	☐	☐	☐	
D7	**复盘日（第 1 关）**	—	—	—	—	
D8	第07讲 记忆系统	☐	☐	☐	☐	

完整 21 天：18 讲 + 2 个复盘日 + 1 个结课日。三条防弃课规则：①「没人能一遍读懂，看不懂就跳，只抓每讲一句话结论」；②「每讲只背 1 个核心结论 + 1 个类比」；③「断了一天不要补两天，直接从当天继续——补课是弃课的开始」。

2) 三关验收（不达标就回炉，不硬推进）

关	节点	必须能不看资料做到

第 1 关	D7 · 能说清骨架	白纸画出 Agent 系统地图（分几层、每层干嘛）；说出主循环的 6 道工序；一句话解释「为什么工具描述比工具代码重要」
第 2 关	D13 · 能写伪代码	一页纸写出迷你 Agent：主循环 + 2 个工具 + 上下文压缩 + 三层记忆 + 权限决策；说出多 Agent 三种模式适用场景；说出记忆五层分别存什么
第 3 关	D21 · 能讲给别人听	给完全不懂的人讲清「自进化为什么危险」「可观测性为什么难」「什么叫活人感」；12 个毕业考题口述答对 9 个以上

3) 每周复盘模板（只填三行，防止复盘变成写作文）

> 这周最没懂的一个概念是：_____ ｜ 十二范式里我能立刻讲出来的是：_____ ｜ 还讲不出来的是：_____

四、结课项目：设计你自己的「迷你 Alice」

学完 18 讲之后做这个。规则只有一句话：**用书里的 12 个范式，把「一个能自己干活的 AI 员工」想清楚、画出来、写明白。**80% 的分在设计文档，20% 是可选加分——但这个项目我给自己的要求是「跑起来」。

```
用户界面（UI）
   ↓
会话层（Session）：管一轮轮对话
   ↓
Agent 主循环（Loop）：问模型 → 执行工具 → 塞回去
   ↓
工具层 / 上下文层 / 记忆层 / 权限层
   ↓
模型供应层（LLM Provider 适配器）
   ↓
可观测层（日志 / 追踪）—— 横穿所有层
```

六个交付物（缺一不可）：① 分层架构图（6 层，标清每层负责什么、不负责什么）；② 主循环设计（六步 + 终止条件 + 压缩触发点 + 急停）；③ 三个工具的契约表（输入输出、错误、副作用、是否需确认）；④ 记忆五层分配表（谁写、谁读、多久清）；⑤ 权限决策矩阵（哪些默认放行、哪些必须问人）；⑥ 可观测三本账（记录仪 / 仪表盘 / 维修本）。

已落地：除设计文档外，我另写了可运行的 `mini_alice.py`（约 1.2 万字节）与沙箱验证脚本，把主循环、工具注册、上下文压缩、记忆分层、权限确认跑通了一遍。

五、附录速查卡（可发给学员的版本）

《十二范式一页纸》节选

#	范式	一句话
1	契约先行	先定输入输出和错误语义，再写实现；接口稳定了，模型和行为才能随便换
2	事件驱动	层与层之间传事件，不传对象；谁需要谁订阅，别硬编码调用链
3	分层隔离	上层指挥、下层干活；同级只平调，不许隔层绕行
4	防环	任何循环都要有终止条件：轮数上限、超时、用户中断，一个都不能少
5	权限最小化	默认关门，显式开门；拦一次就记一笔
6	懒加载	Skill / MCP 都是「目录常在，正文按需」，别一上来全塞进上下文

完整 12 条 +《名词速查卡》（约 60 个术语的大白话解释）+《设计自查清单》（30 个勾选项）随附在课程附录里，可直接改写成学员版 FAQ。

我能给社群直接用的现成东西

- **「考点—口诀」对照表：**18 讲全量，学员考前 30 分钟速刷，可直接做成社群打卡卡片；
- **三关验收题：**每一关 3 道口述题，助教拿来当 1v1 抽检脚本；
- **FAQ 沉淀模板：**「问题原文 → 学员卡在哪 → 图解/一句话 → 延伸练习」，我平时写《问题与修复全记录》就是这个结构；
- **活动复盘模板：**目标 / 参与人数 / 完成率 / 卡点 TOP3 / 下一场改什么（我在项目里做数据复盘用的同一套）。

本附件所有材料均为本人原创产出，源文件在本机可随时提供原始 Markdown 与课程计划：18 讲讲义 × 43 个文件。如需我一节一讲地现场讲解，也可以随时约。