



程序设计训练 (Rust 语言)

第 11 讲: AI Agent 大作业第二次习题课

Rust 程序设计训练教学团队

清华大学

2025—2026 学年度夏季学期

本讲内容

11.1 概览	2
11.2 今天的关键任务	5
11.3 公开展示盘点	7
11.4 共性问题	10
11.5 为 9.10 分课堂展示做准备	21
11.6 后续时间节点	25
11.7 答疑 / Q&A	30

11.1 概览

11.1.1 当前进展

- 已有 **66** 位同学在网络学堂“Agent 公开展示”讨论区发布了设计文档摘要和项目链接。
- 大作业流程已经进入 **互相试用** 阶段：教学团队和同学们都在实际试用、反馈。
- 本次点评基于 **真实的试用反馈**，反馈同学们的共性问题。

11.1.2 本节课内容

- **今天的任务**：9.8 试用与反馈截止
- 公开展示盘点：呈现形式、技术栈、常见的“Agent 专用化”做法
- 共性优点 与 **共性问题**
- 为 9.10 分课堂展示做准备
- 后续时间节点

11.2 今天的任务

11.2.1 截止 9.8 23:59：试用与反馈

- 今晚 23:59 前，至少 **试用 3 位同学的作品**，并在对应的帖子下 **提交反馈**。
- 这对应总分里的“评论与试用参与度”**5 分**。
- 反馈要真实、具体，不能一眼就是 AI 生成的套话。

11.3 公开展示盘点

11.3.1 呈现形式与技术栈

- 界面形式多样：Web 界面、Desktop 应用、CLI、TUI、小程序等，覆盖面广。
- 后端普遍采用了 axum / tokio / serde / sqlite 等第三方库，进行了前后端分离的经典做法。
- 模型接入以 OpenAI 兼容接口为主，也能接 GLM、DeepSeek 等模型。
- 常见“专用化”做法：领域知识库、专用工具、结构化输出校验、Skills 等。

11.3.2 共性优点

- 场景小而真实、定位清晰，而不是“套壳的通用 Agent”。
- 多数把“专用”落到代码和工具，而不是只靠提示词工程。
- 界面与交互逐渐成型，实现了基础功能 R1 - R6。
- 有的同学已经做得相当完整。

11.4 共性问题

11.4.1 环境配置与“能不能用起来”

- 配置环境比较繁琐，建议提供 prebuilt 二进制、多平台构建（比如用 GitHub Actions 自动构建）。
- README 等文档可能与代码不一致，常见的情况是代码改了，文档没跟上改动。
- 需要明确配置目录和数据目录的划分，避免用户换目录启动后找不到之前的数据。
- 如果项目依赖外部平台、被墙的 API、反爬或特定开源库，试用同学容易卡在环境配置上，建议为项目使用者提供较为详细的配置说明和基本的 troubleshooting 指导。

11.4.2 实时进度与等待体验

- 非对话、长耗时的任务（导入资料、出题、批量处理、推理等等）
没有 streaming / 进度。
- 建议长任务实时渲染进度、使用 SSE 流式输出，或后台批量生成，避免“卡住”的观感。
- 或者在用户等待的同时，允许用户做一些其他的操作，进行一定的异步执行。

11.4.3 错误处理与可调试性

- 上传大文件可能 **静默失败**；API Key 配错、网络失败、超出 Token 限制时，往往只报“错误”而 **不给具体原因**。用户看到之后，不知道怎么调试。
- 建议把后端错误转换成用户能看懂的中文提示，提供 API 配置测试、展示 API Response 或错误日志，让用户能自己排查。
- 建议多打印一些日志，许多程序跑起来以后，命令行没有任何输出，此时 Agent 如果出现问题，较难排查。

11.4.4 数据与工具是否“真被用到”

- 上传的资料、配图有时看起来 **没有被 Agent 真正用到**，或什么时候被调用的，用户并不能感知。
- 建议让“知识库 / 工具被调用”的过程 **可见、可追踪**，并明确告知支持的文件格式。
- 可以考虑用户体验，参考一些 Code Agent 展示每一步工具调用的方式。

11.4.5 安全、权限与隐私

- 登录、权限和身份校验要严格，别让用户能伪造身份读到别人的记录。
- API Key 以明文展示或回显，也属于常见问题。
- 处理个人账单、身份、健康等 **敏感数据**要格外谨慎；建议脱敏或给演示用样例数据。
- 非必要不要运行在高权限，做好成本控制，别让 Agent 无约束长时间运行。

11.4.6 代码质量与“是否真正理解”

- 虽然代码主要是 AI 生成，但你还是需要很了解你的代码，保证可维护、可解释；展示/答辩时，可能有同学或者教学团队来问实现细节。
- “能跑就行”和“明白实现”是两回事。

11.4.7 交互引导与体验

- 很多应用缺少入门引导，用户不知道从哪里开始。有份 README 至少能按文档走，但也有一点门槛；要是连文档都没有，用户只能去读代码，体验很差。
- 多站在用户的角度用一遍自己的项目，想想“用户想要什么”。
- 界面上存在小问题，影响用户体验。

11.4.8 功能深度

- 除了基础功能，同学们后续还可以继续根据网络学堂上反馈的建议去实现一些新的功能。
- 这些更多是“锦上添花”，说明不少项目完成度不错，可把它们当作后续迭代的参考。

11.4.9 网络学堂互动

- 大多数网络学堂上的互动是非常认真的，可以看到大家在认真思考选题、上手试用以后，给出了真实的反馈。
- 也有一部分互动比较敷衍，一眼就是 AI 生成的敷衍内容，没有自己思考和试用，不能给其他同学一个很好的反馈。
- 9.10 号分课堂的互评是很重要的环节，请同学们尊重其他同学的劳动成果，要认真客观地打分。

11.4.10 请给出有价值的反馈

- 说清楚 **你试用了什么、在哪一步卡住、哪里好用哪里不好用**，最好能给一个可复现的具体场景。
- 避免空泛的夸奖、照抄别人的话，或者一眼就是 AI 生成的套话，这种反馈对作者没有帮助。
- 换位思考：你希望得到什么样的反馈，就给别人什么样的反馈。

11.5 为 9.10 分课堂展示做准备

11.5.1 展示形式

11.5 为 9.10 分课堂展示做准备

- 9.10 上午上课时间，随机分到 **五个分课堂**，总时间预计不到两小时。
- 每人 **5 分钟** 介绍 + 功能演示，**2 分钟** 左右问答。
- 课堂展示占 **25 分**：同学互评 + 助教 / 老师共同打分。
- 打分方式：课堂在线实时实名打分，给同一个分课堂里所有同学从多个维度（Agent 是否解决实际问题、现场展示效果、问答环节效果、Agent 功能与完成度、技术方案合理性）打分。
- 可以给自己打分，但不计入分数。

11.5.2 怎么把 5 分钟讲好

11.5 为 9.10 分课堂展示做准备

- 一句定位：给谁用、解决什么痛点；再用 1 分钟讲清楚场景背景，让没玩过、没接触过的同学也能听懂。
- **演示很重要**：提前准备可复现的演示用例；检查好网络、API、环境依赖；备好截图 / 离线 / 本地缓存等后备方案。
- 讲清楚“通用 Agent 做不好，我做了哪些 **专用** 定制”。
- 准备一页内容讲做大作业的经验心得，尤其是 AI 方面。

- 老师 / 助教可能深挖：**你的项目的方方面面。**
- 必须 **能解释清楚自己项目的每一部分**；解释不清会被质疑是否自己有深度参与，影响分数。

11.6 后续时间节点

11.6.1 时间线

- 9.6：公开展示发布（已完成）。
- **9.8：试用 + 反馈（今天 23:59 截止）。**
- 9.10：分课堂展示（每人 5 分钟 + 2 分钟问答）。
- 9.13：提交全部材料（网络学堂 + 清华 Git）。

11.6.2 9.13 提交清单

- 网络学堂：完整设计文档（PDF）、AI 对话历史（真实原始记录）、AI 开发开销明细表（按阶段的人时 / 调用量 / Token / 费用）。
- 清华 Git：项目源代码，附 README。
- 提交前对照检查一次，别漏交。

11.6.3 提交前检查清单

- 对照 R1—R6：核心逻辑用 Rust、界面、模型可配置、实时进度与打断、历史管理、Token 与费用统计。
- 设计文档四部分齐全：痛点分析、场景定制、系统架构、技术选型。
- README 与代码一致：怎么编译、怎么配置 Endpoint/Key、怎么运行、附演示用例。

11.6.4 再次强调

- 9.13 之前都可以修改你的项目。
- AI 对话历史必须是 **真实原始记录**，不得伪造或篡改时间戳和内容。
- 你必须对你的项目有足够的熟悉，禁得住提问。
- 期待 9.10 的分课堂展示！祝大家讲清楚、演示好。

11.7 答疑 / Q&A

- 欢迎同学们现场提问，也可以课后在课程群里提问。