

Socratic Debug Mentor 2.6 使用说明

面向初学者编程调试训练的苏格拉底式引导工具

一句话定位

本应用不是替学生直接改代码的答案生成器，而是把学生带回“看证据、提猜想、做验证、再修正”的调试过程，让每一次报错都变成一次可复用的推理训练。

1. 项目介绍

Socratic Debug Mentor 是一个桌面端编程学习辅助应用，适合 Python、JavaScript、C++、Java 等初学者调试场景。它通过左侧代码与终端、右侧导师对话、临时验证窗口和学习记录抽屉，帮助学生在遇到错误时保持主动思考。

- 核心教学目标：让学生学会把报错、输出、变量状态和代码逻辑联系起来。
- 核心交互方式：导师用问题引导，而不是直接给最终答案或完整修复代码。
- 核心流程：观察、猜想、验证、修正四个阶段递进，避免学生跳过推理过程。
- 核心产出：每轮对话会沉淀聊天记录、学生画像和知识点摘要，方便复盘。

2. 使用背景

初学者调试时常见的问题不是不会复制一段修复代码，而是不知道怎样从现象推到原因：看不懂报错、没意识到输出与目标不一致、猜想太模糊，或者一上来就大改原文件。这个项目的设计背景正是把“调试思维”显性化。

课堂痛点	应用的对应设计
学生急着要答案	导师坚持追问证据与判断，不直接代写修复。
学生没有说清目标	观察期优先追问程序目标、当前现象和报错信息。
学生猜想不可验证	猜想期要求写成“我怀疑……因为……”的可检验判断。
学生直接改原文件	验证期先进入临时代码副本，小实验不会影响原文件。
学生反复犯同类错误	知识点摘要用中文显示问题画像和下一步检查。

3. 界面组成

区域	作用	高效用法
代码区	打开、编辑或新建本地代码文件。支持类似 IDLE 的语法高亮开关。	先保证左侧是当前真实代码，再进入引导。
终端区	运行主程序并查看报错、输出或交互输入。	先点击“运行程序”，把真实输出作为观察证据。
导师区	显示苏格拉底式追问、阶段引导和下一步动作。	不要只读导师话，务必在输入框写下自己的判断。
阶段栏	显示观察、猜想、验证、修正四个阶段。	看当前阶段决定该写目标、猜想、验证结果还是修改计划。

区域	作用	高效用法
临时验证窗口	在原代码副本中做最小实验，并使用独立控制台运行。	只改验证所需的最小部分，不在这里完成大修。
学习文件抽屉	查看聊天记录、学生画像和知识点摘要。	课后用知识点摘要复盘自己容易卡住的模式。

4. 四阶段调试流程

阶段	学生要做什么	进入下一步的标志
观察	说清程序目标、当前现象、报错或不符合预期的输出。	目标和问题已经清楚，可以开始猜原因。
猜想	写一句可验证判断，例如“我怀疑问题在循环条件，因为输出少算了一项”。	猜想足够明确，能够设计一次小实验。
验证	在临时代码窗口里运行最小测试，观察输出是否支持猜想。	验证结果支持或反驳猜想，形成下一步依据。
修正	回到左侧原代码做最小修改，再运行主程序复测。	主程序通过或出现新问题；新问题回到观察期。

关键原则

四阶段必须递进。不要在观察不清楚时猜修法，也不要没有明确猜想时进入验证。这样做会慢一点，但能真正训练调试能力。

5. 快速开始

- 1 启动应用。如果出现开场动画，点击“进入学习空间”。
- 2 在右侧设置中配置模型。默认模型为 Gemini；如需其他接口，选择“自定义模型”并填写 API 服务、Model ID、API Key、Base URL 和代理。
- 3 点击左侧文件按钮打开代码文件，或直接粘贴代码。
- 4 点击“运行程序”，在终端区获得真实输出或报错。
- 5 在底部输入框按当前阶段写下内容：观察期写目标和现象，猜想期写怀疑点和理由。
- 6 点击“开始引导”，阅读导师问题，并按提示继续补充。
- 7 当系统提示进入验证时，确认后进入临时验证窗口，编辑验证代码并运行测试。
- 8 根据验证结果选择“我认为符合预期”或“记录验证问题”，再回到原代码完成最小修正。

6. 如何最高效使用

6.1 学生使用方法

- 每次提问前先运行主程序，不要凭印象描述错误。
- 观察期不要写“帮我看”，要写“我希望程序……但现在……”。
- 猜想期不要写“我不会”，要写“我怀疑问题在……因为……”。
- 验证期只做一个小实验，不要在临时代码里顺手改完整答案。
- 修正期只改最小必要位置，改完必须运行主程序复测。

- 每节课结束后看“知识点摘要”，把下一步检查变成自己的调试清单。

6.2 教师使用方法

- 课堂开始前统一配置模型和代理，减少学生在 API 配置上分心。
- 要求学生在每次点击“开始引导”前先写当前阶段记录。
- 观察学生是否跳过猜想直接改代码，把阶段栏作为课堂提醒工具。
- 用“知识点摘要”定位学生反复出现的问题模式，例如循环边界、变量拼写、类型转换、条件判断。
- 评价重点放在调试过程：证据是否清楚、猜想是否可检验、验证是否最小、修正后是否复测。

7. 模型配置说明

当前版本的模型选择被简化为两类：Gemini 默认模型和自定义模型。这样可以避免学生误以为一长串模型都是系统内置可用模型；除 Gemini 以外，其余服务都应由使用者手动配置。

选项	适用场景	需要填写
Gemini	使用默认 Gemini 调用方式。	API Key; 必要时填写代理。
自定义模型	连接 OpenAI 兼容接口、Anthropic、DeepSeek、Qwen、OpenRouter 等。	API 服务、Model ID、API Key、Base URL; 必要时填写代理。

配置建议

如果同一组 API 在其他程序里能用，但本应用调用失败，优先检查 Base URL、Model ID、API 服务协议类型和代理地址是否一致。

8. 临时验证窗口

当学生有明确猜想后，应用会先出现一个进入验证的确认窗口。确认后才会打开临时验证窗口。这个窗口使用原代码副本，运行结果和左侧原文件互不影响。

- 可编辑验证代码：可以添加 print、缩小输入范围、只保留相关分支或变量。
- 独立验证控制台：支持查看输出，也支持在程序需要 input 时输入内容。
- 成功按钮有条件：需要先运行测试并正常结束，才能确认“符合预期”。
- 验证失败会回到猜想期：学生需要记录看到的输出，再重新提出更准确的猜想。

9. 学习记录与知识点摘要

应用会把学习过程保存在本地文件夹中，包括 conversation-records、student-profiles 和 student-issues。右侧抽屉中“知识点摘要”会把内部记录转换成学生可读的中文摘要，不显示原始代码片段。

记录类型	用途
聊天记录	复盘每轮师生对话、阶段判断和导师提示。
学生画像	帮助教师理解学生的调试习惯和求助倾向。
知识点摘要	用中文归纳学生常见问题、最近表现和下一步检查。

10. 常见问题

问题	处理方法
模型调用失败	检查 API Key、Model ID、Base URL、API 服务类型和代理。
进入不了验证	先在猜想期写出明确可验证的原因判断。
验证按钮不能点	先运行验证代码，并等待进程正常结束。
验证代码影响原文件吗	不会。验证窗口使用临时副本，原文件只在左侧编辑器中修改。
知识点里为什么没有代码	这是刻意设计：知识点摘要面向学生复盘，只显示中文问题画像，不暴露原始代码。
什么时候回到观察期	主程序复测后出现新报错或新现象时，应重新观察。

11. 推荐课堂流程

时长	组织方式
10 分钟	教师演示一次完整四阶段：运行、观察、猜想、验证、修正。
20 分钟	学生独立用应用调试一个小错误，必须保留阶段记录。
10 分钟	两人一组互看知识点摘要，说出自己的一个高频问题。
5 分钟	教师总结：本节课不是谁最快改对，而是谁的证据链最清楚。

最终提醒

最高效的用法不是让 AI 变得更像答案库，而是让学生每次都留下一个清楚的小推理：我看到什么、我怀疑什么、我怎样验证、我改了什么。