

实验 14：使用 AI 辅助开发测试

一、实验目的

- 1、掌握使用 AI 编程工具（Cursor）辅助 Java 和 Web 前端开发的流程
- 2、学习利用 AI 生成、解释和调试代码的技术
- 3、掌握 AI 辅助测试用例编写和代码优化的方法
- 4、学会将 AI 工具集成到日常开发工作中，提高开发效率

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 IntelliJ IDEA Community。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

- 1、安装配置 Cursor IDE
- 2、AI 辅助开发
- 3、AI 辅助测试

六、实验内容及步骤

1、安装配置 Cursor

步骤 1：下载安装 Cursor IDE

访问 Cursor 官网 (<https://www.cursor.com/>) 下载安装包，安装 Cursor 并启动，创建账户（支持 GitHub 登录），熟悉界面布局，特别是 AI 对话面板和快捷键。



图1 Cursor官网

步骤 2：配置开发环境

在 Cursor 中安装 Java、Vue、Spring Boot 等插件，配置项目文件路径，了解 Cursor 的主要 AI 功能：

- / 命令：触发代码生成、解释、调试等操作
- Ctrl+K：在编辑器中直接与 AI 对话
- 代码补全和智能提示

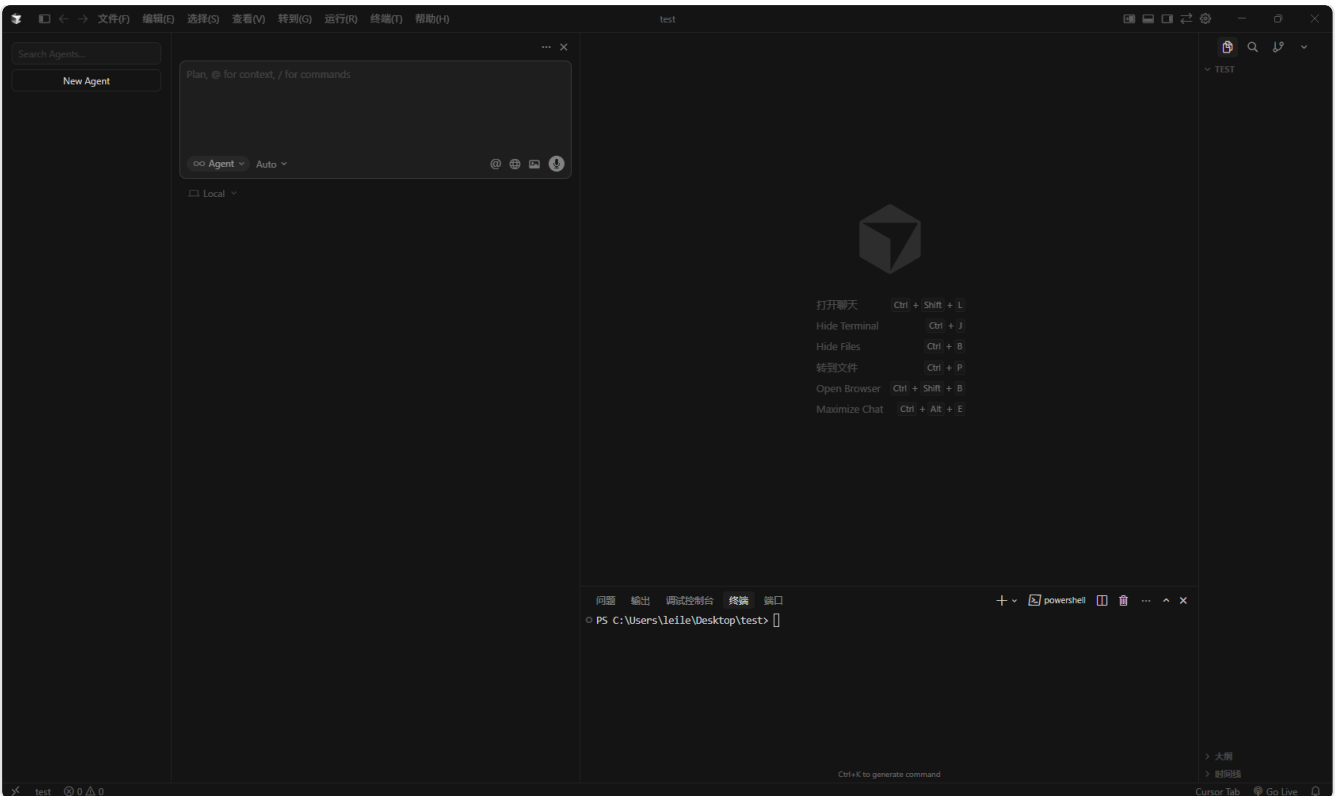


图2 Cursor客户端

步骤 3：熟悉基本操作

在 Cursor 中尝试以下 AI 功能：

- 代码生成：输入"/生成" + 功能描述
- 代码解释：选中代码，输入"/解释"
- 代码优化：选中代码，输入"/优化"
- 调试代码：输入"/调试" + 问题描述
- 测试生成：输入"/测试" + 功能描述

2、AI 辅助开发

步骤 1：使用 AI 辅助前端开发

示例如下：

请为"中医药健康管理系统"设计原型，当前阶段仅聚焦于前端功能，暂不涉及后端接口实现、身份认证逻辑或与其他模块的集成。

功能要求：

1. 支持账号密码登录（用户名/手机号 + 密码）
2. 提供"记住我"或"自动登录"选项（若适用）
3. 登录失败时需有明确的错误提示（如"账号或密码错误"、"验证码已过期"等）
4. 成功登录后示意性跳转至系统首页（无需设计首页内容，仅需表示跳转逻辑）

设计规范要求：

1. 遵循现代医疗健康类应用的UI风格：简洁、可信、易用
2. 适配主流设备尺寸（仅支持PC端访问）
3. 尽量使用View UI Plus组件，保留原始样式，如需修改，通过属性方式修改，尽量不修改组件样式

步骤 2：AI 辅助生成登录页面代码

示例如下：

生成Vue3登录页面，使用View UI Plus组件库，包含以下元素：

1. 登录表单：用户名/手机号输入框、密码输入框、记住我复选框、登录按钮
2. 表单验证：用户名必填、密码长度6-20位
3. 错误提示：使用\$Message组件显示错误信息
4. 页面布局：居中显示，背景色#f5f7fa
5. 样式：使用scoped样式，符合医疗系统简洁风格

步骤 3：AI 辅助生成管理系统主框架

示例如下：

修订模块：管理系统（kc-ai-man）

请为"中医药健康管理系统"设计原型，当前阶段仅聚焦于前端功能，暂不涉及后端接口实现、身份认证逻辑或与其他模块的集成。

管理系统的功能规划如下：

1. 患者管理（一级）：支持患者病情分组、患者风险评定等
2. 健康档案（一级）：支持患者档案信息的管理，档案信息包括病史记录、四诊信息（多次）、中医诊断（多次）、西医诊断（多次）、体质辨识（患者端提交）、症状自查（患者端提交）、指标上报（患者端提交）
3. 慢病服务（一级）：中医评估（二级）、方案制定（二级）、健康打卡（二级）、疗效评价（二级）
4. 系统管理（一级）：权限管理（二级）、数据字典（二级）、消息通知（二级）、日志审计（二级）

设计要求：

1. 使用Layout布局，包含顶部导航、左侧菜单、右侧内容区
2. 左侧菜单支持两级菜单结构
3. 使用View UI Plus组件

4. 响应式设计，适配PC端
5. 每个一级菜单对应一个路由页面

步骤 4：AI 辅助生成患者管理页面

示例如下：

生成患者管理页面，包含以下功能：

1. 患者列表表格：显示患者ID、姓名、性别、年龄、联系电话、病情分组、风险等级
2. 搜索功能：可按姓名、电话、病情分组搜索
3. 操作按钮：新增、编辑、删除、查看详情
4. 分页功能
5. 使用View UI Plus的Table、Button、Input、Select、Page等组件

步骤 5：AI 辅助生成 Java 后端代码

示例如下：

生成Patient实体类，对应数据库表，包含以下字段：

- id: Long (主键)
- name: String (患者姓名)
- gender: String (性别)
- age: Integer (年龄)
- phone: String (联系电话)
- idCard: String (身份证号)
- address: String (地址)
- illnessGroup: String (病情分组)
- riskLevel: String (风险等级)
- createTime: LocalDateTime (创建时间)
- updateTime: LocalDateTime (更新时间)

要求：

1. 使用JPA注解
2. 使用Lombok注解简化代码
3. 实现序列化接口
4. 添加字段校验注解

步骤 6：AI 辅助生成 Spring Boot 控制器

示例如下：

生成患者管理的RESTful API控制器，包含以下接口：

1. POST /api/patients - 新增患者

2. PUT /api/patients/{id} - 更新患者
3. DELETE /api/patients/{id} - 删除患者
4. GET /api/patients/{id} - 获取患者详情
5. GET /api/patients - 分页查询患者列表
6. GET /api/patients/search - 条件搜索患者

要求:

1. 使用@RestController
2. 统一的响应格式
3. 参数校验
4. 异常处理
5. Swagger文档注解

步骤 7: AI 辅助生成 Service 层代码

示例如下:

生成患者管理服务层, 包含以下方法:

1. addPatient(Patient patient) - 添加患者
2. updatePatient(Long id, Patient patient) - 更新患者
3. deletePatient(Long id) - 删除患者
4. getPatientById(Long id) - 根据ID查询患者
5. getPatients(Integer page, Integer size, String keyword) - 分页查询患者
6. searchPatients(PatientQuery query) - 条件搜索患者

要求:

1. 使用@Service注解
2. 添加事务管理
3. 异常处理
4. 日志记录
5. 参数校验

3、AI 辅助测试

步骤 1: AI 辅助生成单元测试

示例如下:

为PatientService生成JUnit 5单元测试, 包含以下测试用例:

1. 测试添加患者功能
2. 测试更新患者功能
3. 测试删除患者功能

4. 测试查询患者功能
5. 测试分页查询功能
6. 测试条件搜索功能

要求：

1. 使用Mockito模拟依赖
2. 包含边界条件测试
3. 包含异常情况测试
4. 测试覆盖率尽可能高
5. 使用AssertJ进行断言

步骤 2：AI 辅助生成集成测试

示例如下：

生成患者管理控制器的集成测试，包含以下测试：

1. 测试新增患者接口
2. 测试更新患者接口
3. 测试删除患者接口
4. 测试查询患者接口
5. 测试分页查询接口
6. 测试参数校验
7. 测试异常处理

要求：

1. 使用@SpringBootTest
2. 使用TestRestTemplate
3. 使用内存数据库H2
4. 每个测试方法独立
5. 清理测试数据

步骤 3：AI 辅助前端测试

示例如下：

为登录页面生成Vitest测试，包含以下测试用例：

1. 测试页面渲染
2. 测试表单验证
3. 测试登录成功
4. 测试登录失败
5. 测试记住我功能

要求：

1. 使用Vue Testing Library
2. 模拟用户交互
3. 模拟API调用
4. 测试UI更新
5. 测试错误提示

七、实验考核

1、本课程实验考核方案

本课程实验考核采用【实验智能评】【实验随堂查】方式开展，根据不同的实验内容选择不同的考核方式。

【实验智能评】：实验完成后提交 GitLab，通过自动化代码评审工具进行评分。

【实验随堂查】：在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

2、本实验考核要求

本实验考核方式：实验智能评

实验 13-15 作为本课程第 4 次实验考核。

考核要求：

- (1) 学生汇报本课程实验的最终成果，由评审组进行评分。
- (2) 学生汇报要脱离自身的开发环境，基于教学信创云计算平台进行汇报。
- (3) 学生汇报通过浏览器访问业务和展示部署成果，不使用多媒体课件和视频演示。