**trading\_floor.py功能规格说明书（Functional Specification）**（中文版）：

**功能规格说明书**

**1. 概述**

本系统为一个交易员自动化执行框架，旨在在市场开盘期间或指定条件下周期性地运行多个交易员任务。该系统通过不同的 AI 模型驱动交易员，以实现多样化的交易策略模拟或自动化操作。

**2. 功能描述**

**2.1 环境变量配置**

系统支持通过**环境变量配置**以下参数：

* **RUN\_EVERY\_N\_MINUTES**
  + 类型：整数
  + 描述：系统循环执行交易员任务的时间间隔（分钟）。
  + 默认值：60
* **RUN\_EVEN\_WHEN\_MARKET\_IS\_CLOSED**
  + 类型：布尔值（true/false）
  + 描述：当市场关闭时，是否仍然执行交易员任务。
  + 默认值：false
* **USE\_MANY\_MODELS**
  + 类型：布尔值（true/false）
  + 描述：是否启用多模型模式，即为不同交易员分配不同的 AI 模型。
  + 默认值：false

**2.2 模型配置**

根据 USE\_MANY\_MODELS 的值，系统支持以下两种模式：

* **单一模型模式**
  + 所有交易员使用相同的模型：gpt-4o-mini
* **多模型模式**
  + 不同交易员分别使用以下模型：
    - gpt-4.1-mini
    - deepseek-chat
    - gemini-2.5-flash-preview-04-17
    - grok-3-mini-beta

**2.3 交易员信息**

系统预设了 4 名交易员：

| **名字** | **姓氏** |
| --- | --- |
| Warren | Patience |
| George | Bold |
| Ray | Systematic |
| Cathie | Crypto |

每位交易员绑定一个 AI 模型，用于执行交易逻辑。

**2.4 系统功能流程**

**2.4.1 系统启动**

* 系统启动后，会加载环境变量，并初始化日志跟踪器（LogTracer）。
* 控制台打印：
* Starting scheduler to run every <RUN\_EVERY\_N\_MINUTES> minutes

**2.4.2 创建交易员**

* 系统通过 create\_traders() 方法，根据交易员姓名及配置的模型名称，生成交易员列表。

**2.4.3 周期性执行任务**

* 通过异步循环，每隔 RUN\_EVERY\_N\_MINUTES 分钟，执行以下步骤：
  + 检查市场是否开盘，除非 RUN\_EVEN\_WHEN\_MARKET\_IS\_CLOSED 设置为 true。
    - 若市场关闭且配置为不运行，打印：
    - Market is closed, skipping run
  + 若市场开盘或允许运行，则并行运行所有交易员的 run() 方法。
* 持续执行循环，直至程序终止。

**3. 模块依赖**

系统依赖以下模块或库：

* dotenv 用于加载环境变量。
* asyncio 用于实现异步循环。
* tracers 的 LogTracer 用于日志跟踪。
* agents 的 add\_trace\_processor 用于注册追踪器。
* market 模块的 is\_market\_open 用于判断市场开闭状态。
* traders 模块的 Trader 类，用于定义交易员逻辑。

**4. 异常处理**

* 若市场关闭且 RUN\_EVEN\_WHEN\_MARKET\_IS\_CLOSED 为 false，则跳过执行交易员逻辑。
* 若环境变量配置异常（例如非整数输入），系统会回退至默认值。

**5. 日志信息**

系统通过 LogTracer 实现对交易员运行流程的跟踪，便于后续调试及监控。

**6. 扩展性**

未来可支持：

* 自定义交易员数量和名称。
* 动态配置 AI 模型。
* 更精细的市场时间表和休市规则。
* 更丰富的交易员策略及多任务分工。