

日历事件管理器实验报告

一. 程序功能介绍

日历事件管理器是一款基于 Python Tkinter 与 pandas 框架开发的桌面应用程序，为用户提供直观高效的日程管理解决方案日历展示系统，他包括了一下几个功能：

日历日期部分：

动态日历显示：以网格形式展示当前月份日历，支持年月快速切换，同时支持一件选择今天的日期。

智能高亮标识：今日日期以金色背景突出显示，包含事件的日期使用浅蓝色背景标记，若同时以蓝色的背景为主。

周系统：顶部显示标准工作周标识，而且在后续支持单双周区分，更有效安排学习任务。
事件管理部分：

事件编辑界面：10 行事件表格支持同时编辑，包含时间/时间段输入框、任务内容自由编辑、完成度状态选择（未开始/进行中/已完成/延期/取消），同时支持完成度自由书写，很自由

智能事件处理：事件按时间自动排序，时间段和时间点区分进行记录，空事件行自动忽略且不会对 excel 产生过多影响，实时保存并记录到 excel 中

批量操作功能：单日事件清空，选择性删除事件，格式刷高级复制（核心创新功能）
数据保存与处理部分：

Excel：自动加载"记录.xlsx"初始数据，保存时生成标准 Excel 格式，同时支持 excel 直接修改以及在程序内修改，相比较而言更加方便

智能格式处理：日期多格式支持，时间自动标准化，使得读取更加流畅，这里 excel 内部该表尽量使用原始格式

数据安全保障：设置自动保存功能，同时有手动保存双保险，在关闭窗口时未保存更改提示而且可以再保存一遍

格式刷功能：三种复制模式，即按星期复制（指定多个星期几，自动复制到未来四周的对应日期）、单双周复制（指定星期几，隔周复制到未来八周）、日期范围复制（设置起止日期，复制到区间内所有日期）同时涵盖所有时间，一键整理所有任务，并且对于格式刷内容都会默认填待评价的评价。

智能任务生成与优化：集成大语言模型 DeepSeek，通过提供个人日程信息与反馈，可以自动推荐学习任务或进行时间优化建议。例如根据历史完成度自动标记高优先任务、生成每日目标总结等。

自然语言交互支持：未来扩展中用户可使用自然语言输入如“安排一周数学复习计划”，系统将调用大模型完成解析并自动填充事件表格，现在准确度不够。

二. 项目各模块与类设计细节

主类 CalendarApp 架构

CalendarApp 类是整个应用程序的核心，负责初始化界面、管理事件数据和协调各模块工作。在初始化过程中，它创建了主窗口，设置了初始大小和标题，并初始化了事件存储结构。同时，它尝试从 Excel 文件加载已有事件，然后创建用户界面组件，并更新日历显示。最后，它设置了窗口关闭事件的处理，确保在关闭前提示保存未保存的更改。事件数据结构采用字典形式，键为日期字符串（格式为"YYYY-MM-DD"），值为事件列表，每个事件是一个包含时间、任务和完成度的字典。

核心方法包括：

创建界面组件（create_widgets）

更新日历显示（update_calendar）显示选定日期的事件（show_events）

保存当前编辑的事件（save_current_events）

保存所有事件到 Excel（save_events_to_excel）

从 Excel 加载事件（load_events_from_excel）

应用格式刷功能（apply_format_brush）

处理窗口关闭事件（on_closing）

数据处理引擎

日期处理子系统负责多种日期格式的解析和转换。parse_excel_date 方法支持 5 种日期格式的解析，包括"2023.05.01"这样的点分隔格式、pandas 时间戳、datetime 对象、"2023-05-01"这样的标准格式以及 Excel 序列日期。format_excel_date 方法则将日期统一转换为"年.月.日"格式，以便在 Excel 中保持一致性。

时间处理子系统包括时间标准化和转换功能。normalize_time 方法统一时间分隔符，处理时间段（如将"9:00-12:00"转换为标准格式），并补全单个时间点（如将"9"转换为"09:00"）。time_to_minutes 方法将时间字符串转换为分钟数，用于事件排序。

用户界面模块

界面布局分为左右两个主要区域。左侧日历包含控制栏和日历网格；右侧事件区占 1/3 宽度，包含日期标题栏、事件表格和操作按钮栏，控制栏包含月份下拉框、年份下拉框以及操作按钮（今天、保存、加载）。日历网格是一个 6 行 7 列的按钮矩阵，用于显示当前月份的日期。每个日期按钮可以显示当天的日期，并根据是否包含事件或是否为今天进行高亮显示。

事件表格是一个 10 行 4 列的输入区域，每行包含时间输入框、任务输入框、完成度下拉框和格式刷按钮。时间输入框用于输入时间或时间段，任务输入框用于输入任务描述，完成度下拉框提供五个选项供用户选择，格式刷按钮则用于触发高级复制功能。

操作按钮栏包含三个按钮：保存更改、删除事件和清空事件。动态组件管理通过多个列表实现：`self.day_buttons` 管理日历网格中的日期按钮，`self.time_entries`、`self.task_entries` 和 `self.completion_entries` 分别管理事件表格中的时间输入框、任务输入框和完成度下拉框，`self.format_brush_buttons` 管理格式刷按钮。

核心功能实现

日历渲染引擎（`update_calendar` 方法）首先获取选择的年月，然后使用 `monthcalendar` 生成该月的日历数据。接着，它遍历 6 行 7 列的日历网格，为每个网格设置相应的日期。对于非当月的日期，禁用对应的按钮；对于当天日期，设置金色背景；对于包含事件的日期，设置浅蓝色背景。

事件保存逻辑（`save_current_events` 方法）遍历事件表格的 10 行，收集有效的事件数据。对于每个有效事件，它标准化时间字符串，并确保完成度有默认值。收集到的事件列表按时间排序后，更新到当前选定日期的事件字典中。如果当前日期没有事件，则从字典中删除该日期。最后标记数据已修改，并更新日历显示。

格式刷实现（`apply_format_brush` 方法）首先从选中的行获取源事件的时间段和任务描述。然后根据用户选择的复制模式（按星期、单双周或日期范围）计算目标日期列表。对于每个目标日期将源事件添加到该日期的事件列表中，并重新排序。最后，更新日历显示并标记数据已修改。

大语言模型模块

通过对 LLM API, Model Integrator 等类模块的调用，大幅度提升其智能程度，从而使得用户能自由定义日程要求，可以进一步实现时间智能高效利用。

三.项目分工

范思滕：UI 部分，实验报告

黄家蔚：智能规划部分，视频录制

四. 项目总结与反思

这个项目使用了 Python Tkinter 进行 ui 制作，整体上手难度较低，比较简介，提升了对于 python 底层逻辑的理解。通过对多元的类函数的使用进一步提升了程序的调理与可阅读性，整体程序选题立意扎根与同学们的生活，通过进一步优化时间排表更加有利于同学们的课余学习以及时间安排，并且该程序与人工智能相关结合，通过调用大模型 api 进一步实现智能化安排时间与任务，从而大幅提升对于时间规划的认识以及时间安排的效率。

DeepSeek API 支持智能分析与自然语言日程生成，极大拓展了日历管理的能力，具备良好的智能化与拓展潜力。而在数据处理上并没有选择 json 文件，而是选择 excel 表格进行处理，一是结合了本学期学习的知识，进一步提升对于 pandas 库等的熟悉，二是利用天然 excel 优秀的数据处理能力与功能，从而进一步实现对于数据等高效的处理与保存，同时 excel 更容易进行规划，更容易直接通过更改表格来完成另一种方向内容的改变（即插即用）更加符合设计此程序的初心。

同时项目也存在一些不足之处，受限于人工智能调用水平并不能做到调用的内容以及所显示的内容完全合理，同时软件整体 ui 比较简陋，并不是很能满足用户对于个性化页面设计等的满足，而且没有系统端的接口，类似于自启动，闹钟等比较合理的日历附带功能并没有完全实现，总的来讲提升空间较大，还需要在日后的实践中继续努力。