
WX To Do 任务管理器

软件设计文档

王鑫*

电子信息与电气工程学院

2023 年 4 月 24 日

1 引言

WX To Do 是一款在 Linux 系统下基于 Python 和 Qt 框架开发的任务管理器应用，旨在帮助用户简单高效地管理和追踪待办事项。本文档将描述 WX To Do 任务管理器的软件设计细节，包括软件体系结构、控制过程设计和用户界面设计。本任务管理器实现了任务的添加、编辑、删除、标记完成、设置到期时间等功能。

2 软件体系结构

WX To Do 任务管理器采用模块化设计，分为以下三个主要模块：

- **主窗口模块**：负责初始化应用、创建主窗口，负责程序的主要界面和功能实现。
- **任务列表模块**：负责存储和管理用户的任务数据。
- **任务类模块**：定义单个任务的属性和方法。
- **主题模块**：负责更换背景颜色或更改主题，实现个性化使用。

2.1 模块关系

各模块之间的关系如下：

- 主窗口模块调用任务列表模块，将任务信息展示在列表中。
- 主窗口模块调用任务类模块，实现任务的创建、编辑、删除功能。
- 主窗口模块调用主题模块，实现背景颜色的更改和主题的更改。

*520021910700, wangxin.1@sjtu.edu.cn

2.2 主窗口模块

主窗口模块包含一个 `MainWindow` 类, 继承自 `QMainWindow`, 定义在 `ui.py` 中。`MainWindow` 类具有以下属性:

- **`task_list_widget`**: 一个 `QListWidget` 实例, 用于展示用户的任务列表
- **`task_list`**: 一个 `TaskList` 实例, 用于存储和管理用户的任务数据

`MainWindow` 类的方法主要负责任务的添加、编辑、删除、标记完成、设置到期时间等功能的实现, 以及数据的加载和保存。

2.3 任务列表模块

任务列表模块包含一个 `TaskList` 类, 用于实现任务在界面中的列表显示, 定义在 `task_list.py` 中。

`TaskList` 类包含以下方法:

- **`create_task`**: 在任务列表中添加新的任务。
- **`sort_tasks_by_priority`**: 根据优先级对列表中的任务进行排序。
- **`remove_task`**: 从任务列表中删除任务。

2.4 任务类模块

任务类模块包含一个 `Task` 类, 用于表示单个任务的属性和方法, 定义在 `task.py` 中。`Task` 类具有以下属性:

- **标题 (`title`)**: 任务的描述。
- **完成状态 (`completed`)**: 表示任务是否已完成。
- **到期时间 (`due_date`)**: 任务的到期时间。

`Task` 类具有以下方法:

- **构造函数**: 初始化任务实例, 设置标题、完成状态和到期时间。
- **`set_completed`**: 设置任务的完成状态。
- **`set_due_date`**: 设置任务的到期时间。

2.5 主题模块

主题模块包含一个 Theme 类，用于设置背景颜色或背景图片，定义在 theme.py 中。Theme 类具有以下方法：

- **set_background_image**: 用户设置背景图片。
- **get_style_sheet**: 返回当前所用背景图片所在路径。

3 控制过程设计

WX To Do 任务管理器的控制过程主要包括以下几个部分：

- **任务的添加、删除**: 通过 MainWindow 类的方法实现任务的添加、编辑、删除功能，并在操作完成后调用 save_data 函数将数据保存到 JSON 文件。
- **任务的标记完成、设置到期时间**: 通过 MainWindow 类的方法实现任务的标记完成、设置到期时间功能，并在操作完成后调用 save_data 函数将数据保存到 JSON 文件。
- **任务数据的加载和保存**: 在 MainWindow 类的构造函数中调用 load_data 函数加载已保存的任务数据，通过 save_data 函数将任务数据保存到 JSON 文件。
- **用户个性化设置的加载与保存**: 在 MainWindow 类的构造函数中，从 JSON 文件中加载已保存的用户设置（如背景颜色和图片），并应用这些设置到用户界面。在用户更改背景颜色或保存图片时，将这些设置保存到 JSON 文件中。

3.1 任务的添加、删除

- **添加任务**: 在 MainWindow 类中，用户可以通过点击添加按钮或者输入任务标题后按回车键来添加任务。添加任务时，将创建一个新的 Task 实例，并将其添加到 TaskList 实例的 tasks 列表中。同时，将任务显示在 task_list_widget 中。
- **删除任务**: 在 MainWindow 类中，用户可以通过右键点击任务并选择“Delete”来删除任务。删除任务时，先根据任务标题从 TaskList 实例的 tasks 列表中移除该任务，然后从 task_list_widget 中移除任务。最后，调用 save_data 方法将数据保存到 JSON 文件。

3.2 任务的标记完成、设置到期时间

- **标记任务完成**: 在 MainWindow 类中，用户可以通过右键点击任务并选择“Mark as Completed”来标记任务完成。标记任务完成时，先更新 Task 实例的完成状态，然后

修改任务在 `task_list_widget` 中的显示效果（改变字体颜色）。最后，调用 `save_data` 方法将数据保存到 JSON 文件。

- **设置任务到期时间：**在 `MainWindow` 类中，用户可以通过右键点击任务并选择“Due Today”或“Due Tomorrow”来设置任务到期时间。设置任务到期时间时，先计算新的到期时间，然后更新 `Task` 实例的到期时间。最后，调用 `save_data` 方法将数据保存到 JSON 文件。

3.3 任务数据的加载和保存

- **加载数据：**在 `MainWindow` 类的构造函数中，调用 `load_data` 函数从 JSON 文件中加载已保存的任务数据。加载数据时，需要将 JSON 数据转换为 `Task` 实例，并将其添加到 `TaskList` 实例的 `tasks` 列表中。同时，将任务显示在 `task_list_widget` 中
- **保存数据：**在 `MainWindow` 类中，通过 `save_data` 函数将任务数据保存到 JSON 文件。保存数据时，需要将 `Task` 实例转换为 JSON 数据，并将 JSON 数据写入文件。

3.4 个性化设置数据的加载与保存

- **加载数据：**在 `MainWindow` 类的构造函数中，调用 `load_settings` 函数从 JSON 文件中加载已保存的个性化设置数据，同时调用相关函数设置背景色或主题。
- **保存数据：**在 `MainWindow` 类中，通过 `save_settings` 函数将任务数据保存到 JSON 文件。保存数据时，需要注意正确保存图片路径。

4 图形用户界面 (GUI) 设计

WX To Do 任务管理器的用户界面主要由以下几个部分组成：

- **标题及标语：**位于窗口顶部，包含软件名称 WX To Do 与标语 `manage everything everywhere all at once`。
- **任务列表：**位于窗口中部，展示用户的任务。用户可以通过右键点击任务进行编辑，弹出上下文菜单，提供标记完成、设置到期时间和删除任务等功能。
- **添加任务按钮：**位于窗口底部工具栏，用户可以点击此按钮来添加任务。
- **个性化设置按钮：**位于窗口底部工具栏，用户可以点击背景色或背景图片按钮打开设置菜单，更改背景颜色和主题。

5 模块间的交互

WX To Do 任务管理器中的各个模块通过相互调用实现功能的协同。以下是主要模块间交互的描述：

- **MainWindow 类与 TaskList 类交互**：MainWindow 类负责接收用户输入，根据用户操作调用 TaskList 类的函数实现任务的添加、删除、查找和排序等功能。例如，添加任务时，MainWindow 类会调用 TaskList 类的 `add_task` 函数将任务添加到任务列表中。
- **TaskList 类与 Task 类交互**：TaskList 类中的任务列表由 Task 实例组成。TaskList 类需要调用 Task 类的函数来设置或获取任务的属性，例如设置任务的完成状态、获取任务的标题等。同时，TaskList 类负责管理 Task 实例的生命周期，例如在任务列表中添加或删除任务。
- **MainWindow 类与 Task 类交互**：MainWindow 类需要调用 Task 类的函数来设置或获取任务的属性，以便在界面上显示任务的信息或根据用户操作更新任务的属性。例如，当用户标记任务完成时，MainWindow 类需要调用 Task 类的 `set_completed` 函数来设置任务的完成状态。

6 数据存储

由于用户的任务数据或个性化设置数据较少，WX To Do 任务管理器使用 JSON 格式的文件来存储任务数据或个性化设置数据。

数据的存储过程如下：

- **将相关数据转化为 JSON 数据**：在保存相关数据之前，需要将 Task 实例的属性（标题、优先级、完成状态、到期时间等）以及图片的路径和背景色转换为 JSON 数据。可以使用 Python 标准库中的 `json` 模块来实现这一功能。
- **将 JSON 数据写入文件**：将转换后的 JSON 数据写入一个文件中，以便在下次启动程序时加载任务数据。可以使用 Python 的文件 I/O 功能来实现这一功能。

数据的加载过程如下：

- **从文件中读取 JSON 数据**：在程序启动时，需要从之前保存的 JSON 文件中读取任务数据。可以使用 Python 的文件 I/O 功能来实现这一功能。
- **将 JSON 数据转换为所需数据**：将从文件中读取的 JSON 数据转换为 Task 实例或图片路径与背景色，并进行后续的处理。可以使用 Python 标准库中的 `json` 模块来实现这一功能。

7 总结

本设计文档详细描述了 WX To Do 任务管理器的软件体系结构、控制过程设计、用户界面设计、模块间的交互、数据存储和异常处理等方面的内容。该文档可为 WX To Do 的开发提供指导与记录。