

软件设计与建模——任务单 app

1 主要功能

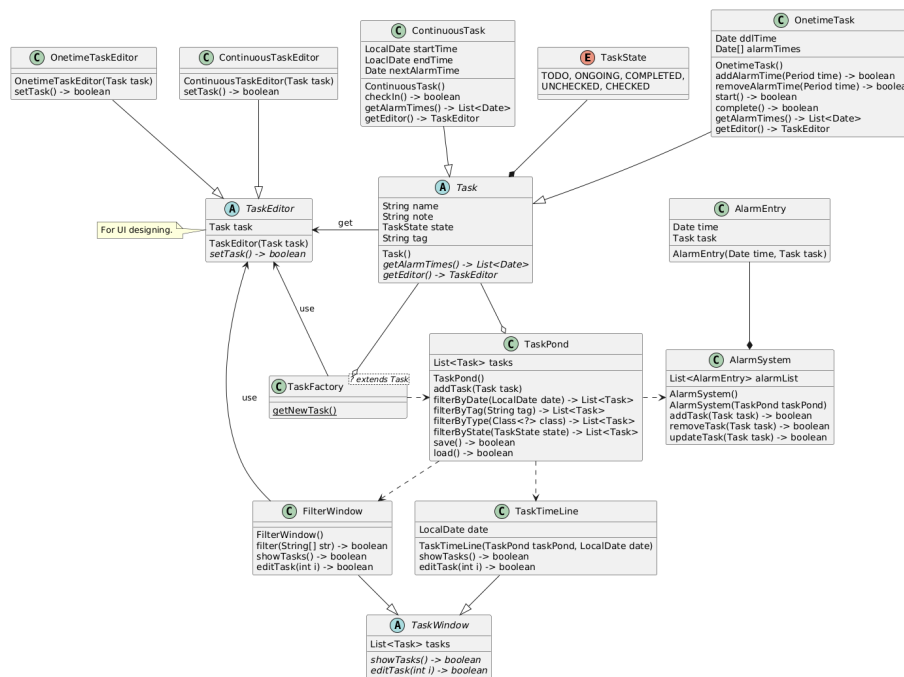
软件主要分为四块功能：

1. 任务持久化存储
2. 任务管理
3. ddl 提醒
4. 图形化界面

接下来结合 UML 图说明这些功能如何实现

2 UML 图

2.1 类图



2.1.1 类的说明

- Task, OnetimeTask, ContinuousTask

Task 是任务单 app 处理的基础单位，分为单次任务和长期任务，它们有着不同的属性，并且通过两个抽象方法 getAlarmTimes 和 getEditor 实现不同的管理方式。

所有任务的公共属性：

1. 名称
2. 详细信息
3. 自定义标签
4. 状态（单次任务和长期任务有着不同的状态集合）

单次任务的属性：

1. 三种状态（代办、进行中、已完成）
2. ddl 时间（精确到分钟）
3. 多个 ddl 提醒时间

长期任务的属性：

1. 开始日期、结束日期（精确到天）
2. 下一次 ddl 时间

- TaskPond

存放所有 task，提供筛选和本地存储的方法

- AlarmEntry, AlarmSystem

ddl 提醒系统，保持和 task pond 的同步

- TaskEditor, OnetimeTaskEditor, ContinuousTaskEditor

为了方便实现 ui，从 task 中独立出来

作为用户和 task 的中间层，提供符合人类习惯的任务管理服务

- TaskWindow, FilterWindow, TaskTimeLine

面向用户的界面，showTasks 和 editTask 两个抽象方法提供对 task 快速管理的支持

2.1.2 类的关系

1. 继承

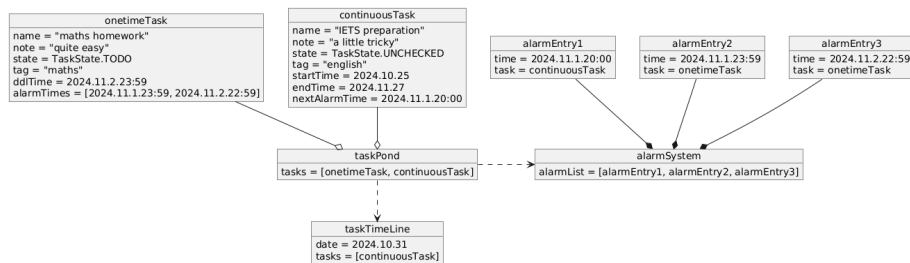
Task, TaskEditor 将两种任务抽象为任务大类，减少了设计复杂度

TaskWindow 将用户界面的功能抽象了出来：一些用户想看见的任务，以及快速编辑它们的方法

2. 聚合

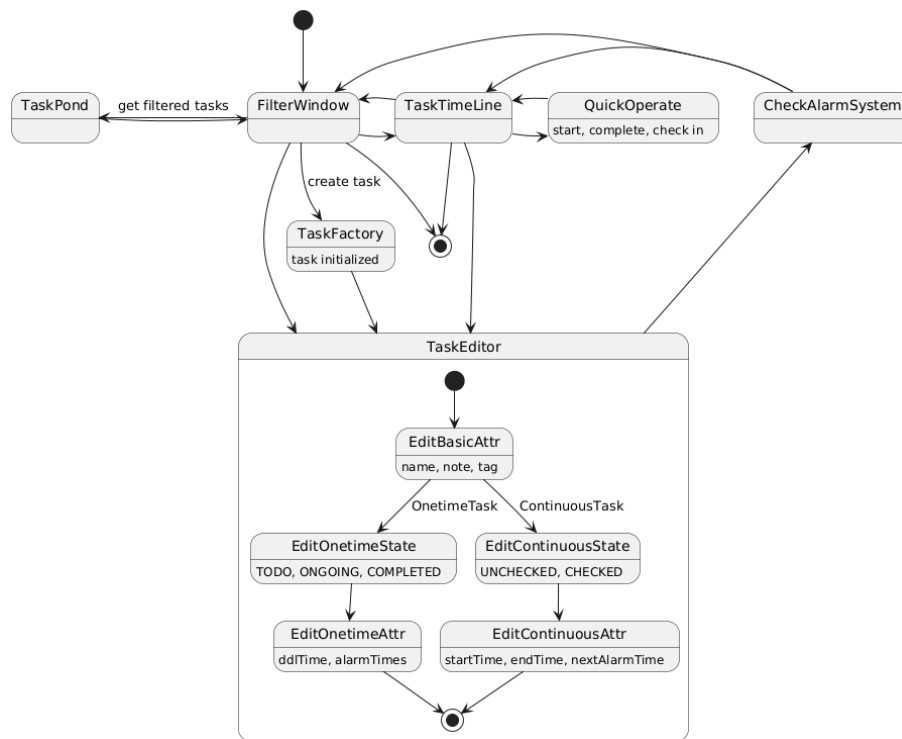
TaskPond 聚合了所有初始化完毕的 Task，承担了数据库的作用

2.2 对象图



以上是对于当系统中存在一个单次任务和一个长期任务，并且任务时间线打开时的状态（后台的筛选界面被省略）

2.3 状态图



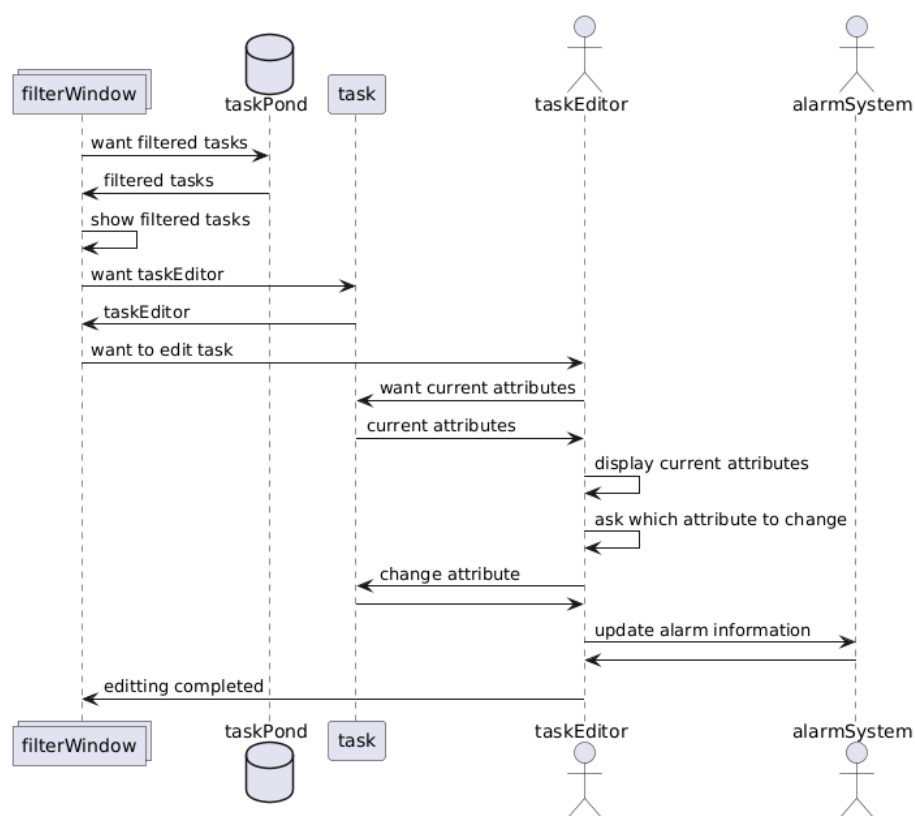
状态图指示了这个系统中的动态行为：

- 筛选窗口可以进行以下操作：
 - 更改筛选条件，得到新的任务序列
 - 新建任务
 - 管理任务
 - 切换到时间线界面
- 时间线界面可以进行以下操作：
 - 任务的快速操作：开始（单次任务）、完成（单次任务）、打卡（长期任务）
 - 管理任务
 - 切换到筛选窗口
- 管理任务的过程

1. 根据任务当前状态和类型显示不同的表单
 2. 等待用户管理
 3. 修改对应 task 的属性
 4. 更新 ddl 提醒系统
- 新建任务

新建一个空任务（包含任务类型），然后进入任务管理

2.4 时序图



以上时序图展示了用户两个动作下发生的交互：首先在 filterWindow 界面对任务进行了筛选，随后对某个任务管理

1. filterWindow 解析筛选条件后对 taskPond 提出访问

2. filterWindow 得到结果并显示
3. filterWindow 找到想编辑的 task，把 workflow 交给它对应的 taskEditor
4. taskEditor 通过 task 的属性显示编辑表单
5. 得到用户选择的结果后，修改 task 的属性
6. 更新 ddl 提醒系统
7. taskEditor 将 workflow 交还 filterWindow

3 UI 界面

3.1 FilterWindow

下拉式选单

DATE	TAG	TYPE	STATE
11.1	STUDY	ONETIME	TODO

筛选结果

NAME	TAG	NOTE	TYPE	STATE
MATH EXAM	STUDY	HARD	ONETIME	TODO

通过上面四个下拉式选单对所有 task 进行筛选，在下方显示筛选结果

3.2 TaskTimeLine

2024年11月							ONETIME			CONTINUOUS		
一	二	三	四	五	六	日	DDL	NAME	STATE	NAME	STATE	NEXT
28 廿六	29 廿七	30 廿八	31 廿九	1 十月	2 初二	3 初三						
4 初四	5 初五	6 初六	7 立冬	8 初八	9 初九	10 初十						
11 十一	12 十二	13 十三	14 十四	15 十五	16 十六	17 十七						
18 十八	19 十九	20 二十	21 廿一	22 小雪	23 廿三	24 廿四						
25 廿五	26 廿六	27 廿七	28 廿八	29 廿九	30 三十	1 十一月						
2 初二	3 初三	4 初四	5 初五	6 大雪	7 初七	8 初八						

- 任务时间线类似日历，显示一个月的日期，点击某天显示该天未完成的单次任务和长期任务
- 点击任务显示详细信息
- 对任务添加删除编辑检索筛选

3.2.1 单次任务

- 显示状态为待办和进行中的单次任务（如果超过 ddl，标注为过期）
- 按 ddl 时间排序，从早到晚
- 显示内容：
 1. 截止日期（还要包括星期几）
 2. 任务名
 3. 状态

3.2.2 长期任务

- 显示所有在开始、结束日期内的长期任务
- 显示内容：
 1. 任务名

2. 今日是否打卡
3. 下次打卡日期

4 设计思路和心得

4.1 低耦合

通过将应用程序划分为若干独立模块，每个模块负责特定功能，使系统更易于开发和维护。模块之间的低耦合性使得后期扩展和修改更加方便。

通过抽象类和继承，将单次任务和长期任务抽象为 Task，减小系统复杂度。

4.2 中间层

将 TaskEditor 从 Task 中分离出，构造用户和数据之间的中间层，使得系统结构更清晰。

清晰定义每个类和模块的职责，避免功能交织和代码冗余。例如，将用户界面、任务管理和数据存储分离开来，使得每一部分的更改都不会影响其他部分。