

# 软件实现与构建——任务单 app

## 1 构建平台

IDE: IntelliJ IDEA

Github Repository: <https://github.com/kyhsdjg/task-list>

## 2 构建过程

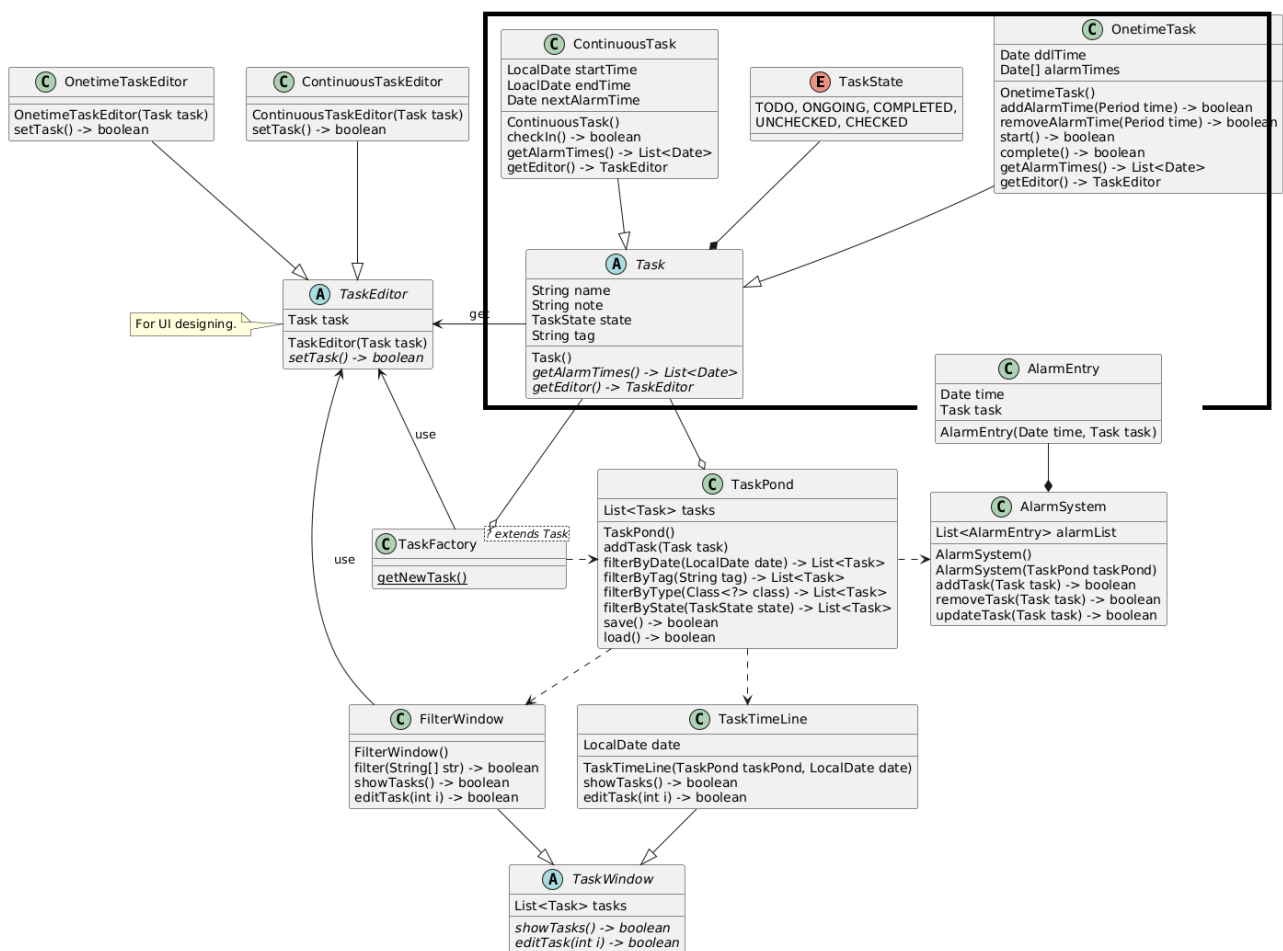
按照 git commit 的时间线介绍

### 2.1 初始化

1. create Github Repository
2. add documents
3. create Java project

### 2.2 Task

对应下图中黑框内的类



它们同属包 `com.github.kyhsdjg.tasklist.data.task` 中

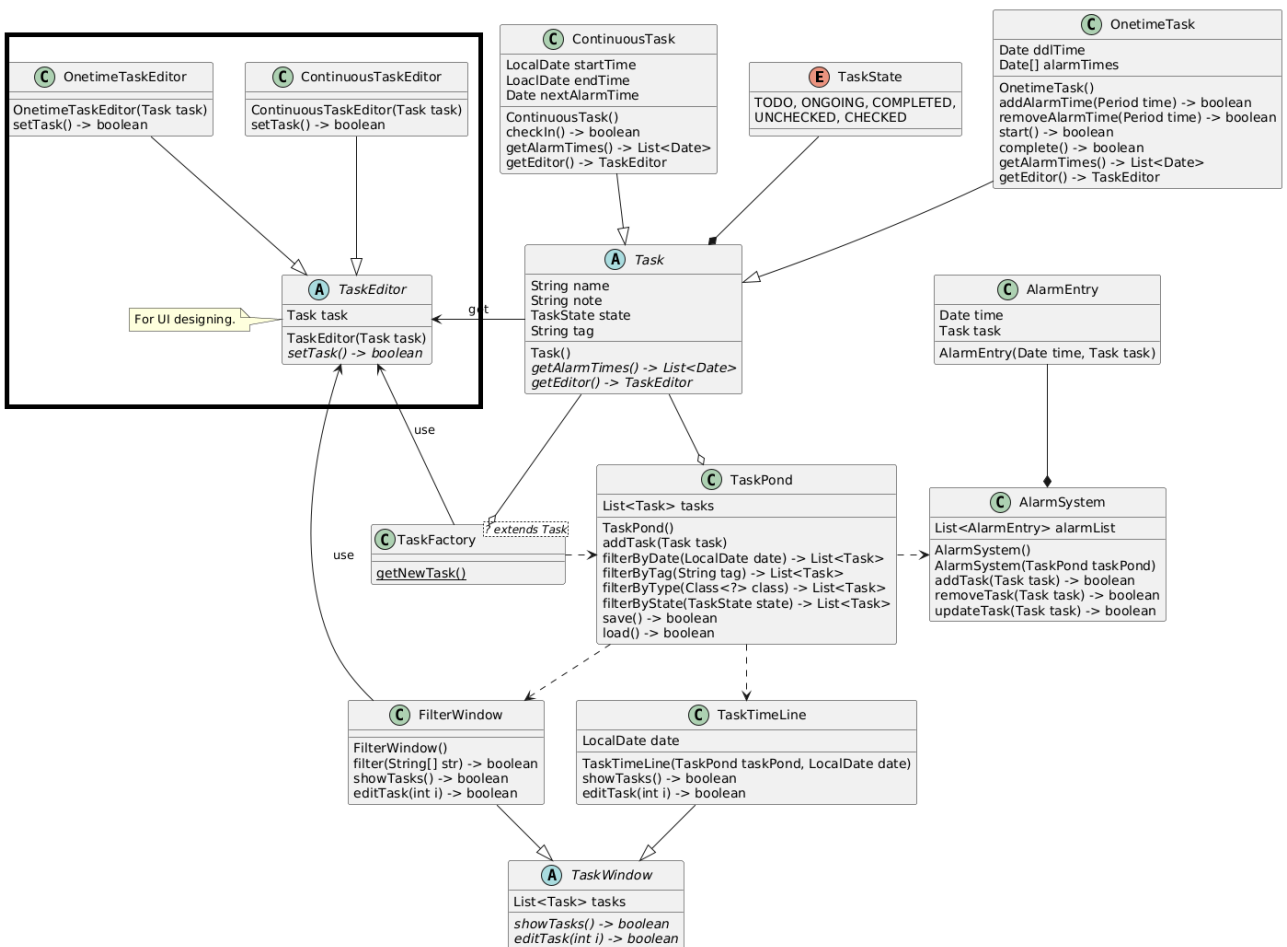
此时 TaskEditor 和 AlarmSystem 还没实现，相应的功能暂时用 TODO 标注等待之后完成。

多态

Task 对 getAlarmTimes 实现了多态，具体而言：

- OnetimeTask 会在删除所有已经过期的 alarmTimes 后，从早到晚返回它存储的所有 alarmTimes;
- ContinuousTask 会返回所有从 nextAlarmTime 到 endDate 的每一天同一时刻的 alarmTimes。

## 2.3 TaskEditor



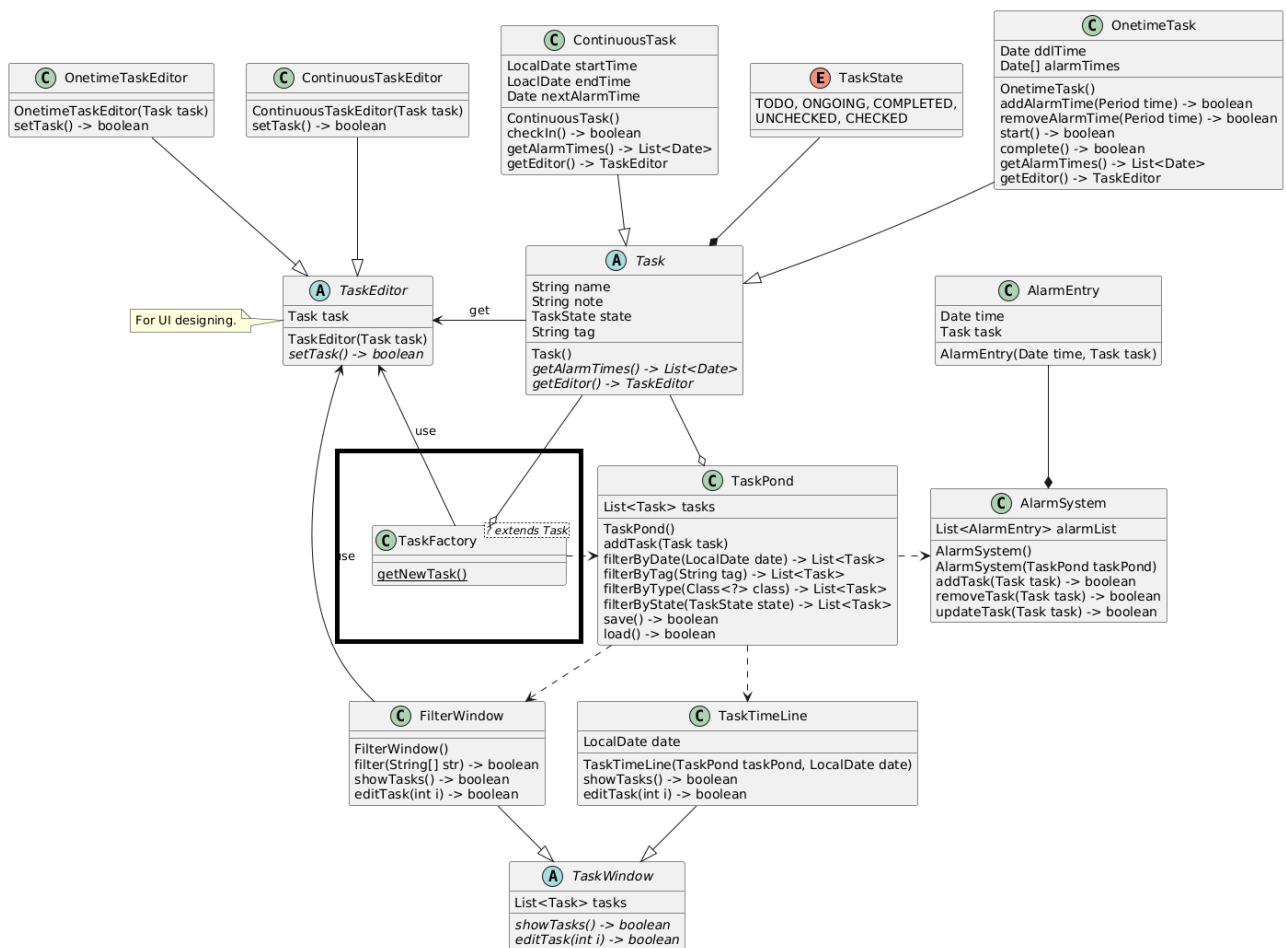
TaskEditor 作为用户和 Task 的中间层，属于类 `com.github.kyhdsjq.tasklist.ui.taskeditor`。

## 访问者模式

TaskEditor 作为独立于 Task 的类，它既要 Task 的两种子类进行不同的操作，而又要尽量少地修改 Task 类本身。

因此这里使用了 **访问者模式** 来实现：在 Task 中创建一个多态抽象函数 getTaskEditor，再让两个子类的实现分别返回对应的两种 TaskEditor 子类。最后，通过调用 TaskEditor 子类的方法对 Task 进行修改。

## 2.4 CLI & TaskFactory



CLI 封装了所有与外部类型相关的交互操作，例如以下方法封装了询问日期的操作

```
public static LocalDate askForLocalDate() {
    while (true) {
        try {
            int year = askForInt("Please enter the year:");
            int month = askForInt("Please enter the month:");
            int day = askForInt("Please enter the day:");

            return LocalDate.of(year, month, day);
        } catch (DateTimeException e) {
            System.out.println("Invalid date entered. Please try
                                again.");
        }
    }
}
```

## 工厂模式

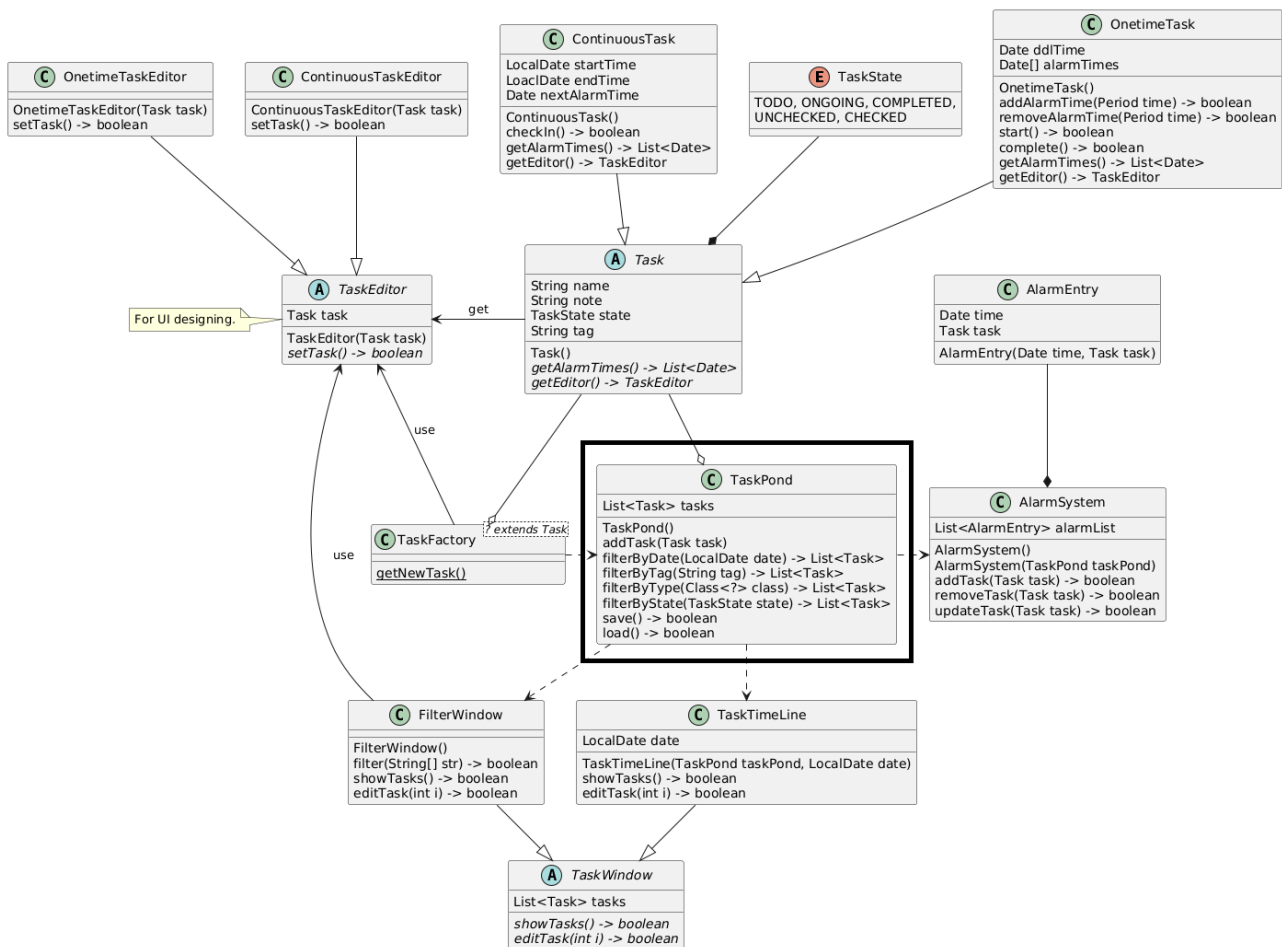
用户在创建一个 Task 后，往往要对它的多项属性进行设置。一般地，一个 Task 只有进行用户的初始化设置后才会被系统接收。

在真正得到新的 Task 前要进行多项操作，因此采用了工厂模式。

TaskFactory 首先会询问用户要构造哪种类型的 Task，接着调用对应的构造函数，最后通过 TaskEditor 让用户进行 Task 属性的设置。

它们都属于包 com.github.kyhdsjq.tasklist.ui 中。

## 2.5 TaskPond

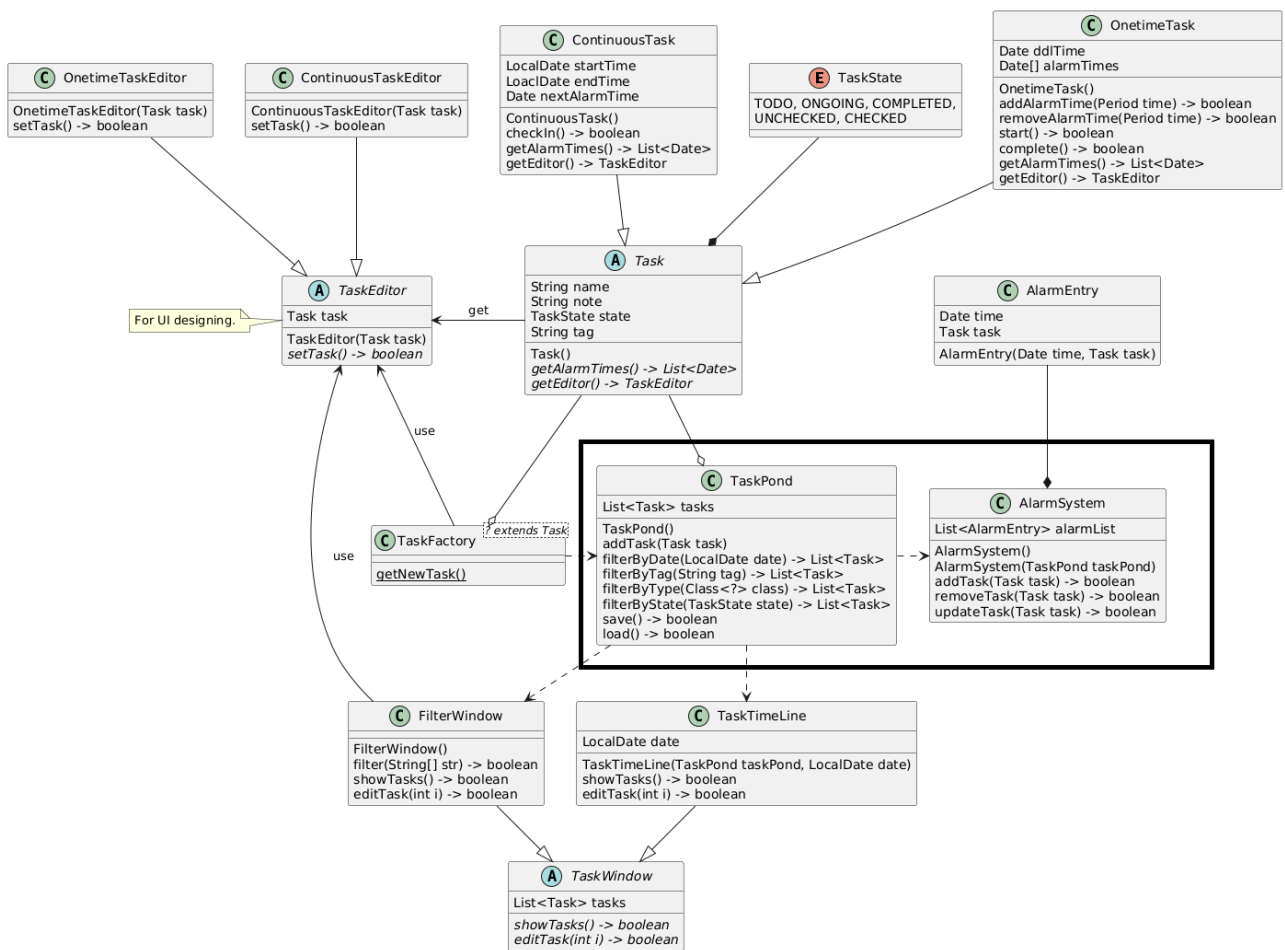


TaskPond 是存放 Task 的数据库，同时提供了筛选 Task 的功能。它存放在 com.github.kyhdsjq.tasklist.data 中。

Task 并不一定在某个 TaskPond 中，它们需要被从 TaskFactory 生出来，再通过 addTask 函数放进 TaskPond 中。

不过可以确保同一个 Task 不会同时存在在两个 Pond 中（加入第二个 Pond 后会从第一个中弹出来）。

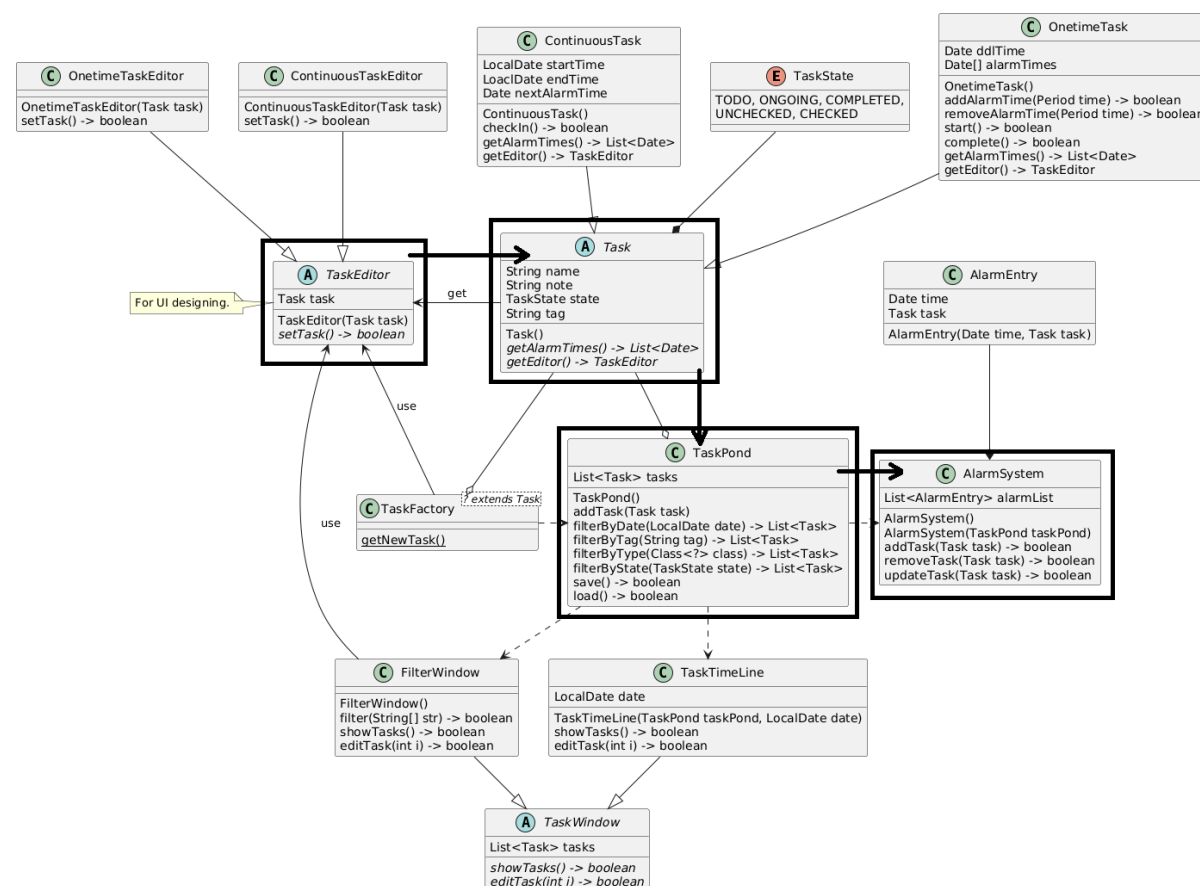
## 2.6 AlarmSystem



AlarmSystem 和 TaskPond 同属 com.github.kyhsdj.tasklist.data。它只有包访问权限，也就是只对 TaskPond 可见。

AlarmSystem 的提醒信息来源于 TaskPond。当修改任务或增减任务时，TaskPond 会将信息同步给 AlarmSystem。

具体而言，任务被通过 TaskEditor 修改后，首先会调用 TaskPond 的 syncTask 方法，然后 TaskPond 会将对应的 task 变化信息传递给 AlarmSystem。（如下图）



AlarmSystem 和 TaskPond 不是直接的从属关系。

TaskPond 一定指向一个 AlarmSystem，但是一个 AlarmSystem 可以同时被多个 TaskPond 指向。（指向关系：该 TaskPond 中的 Task 提醒任务交给指向的 AlarmSystem）

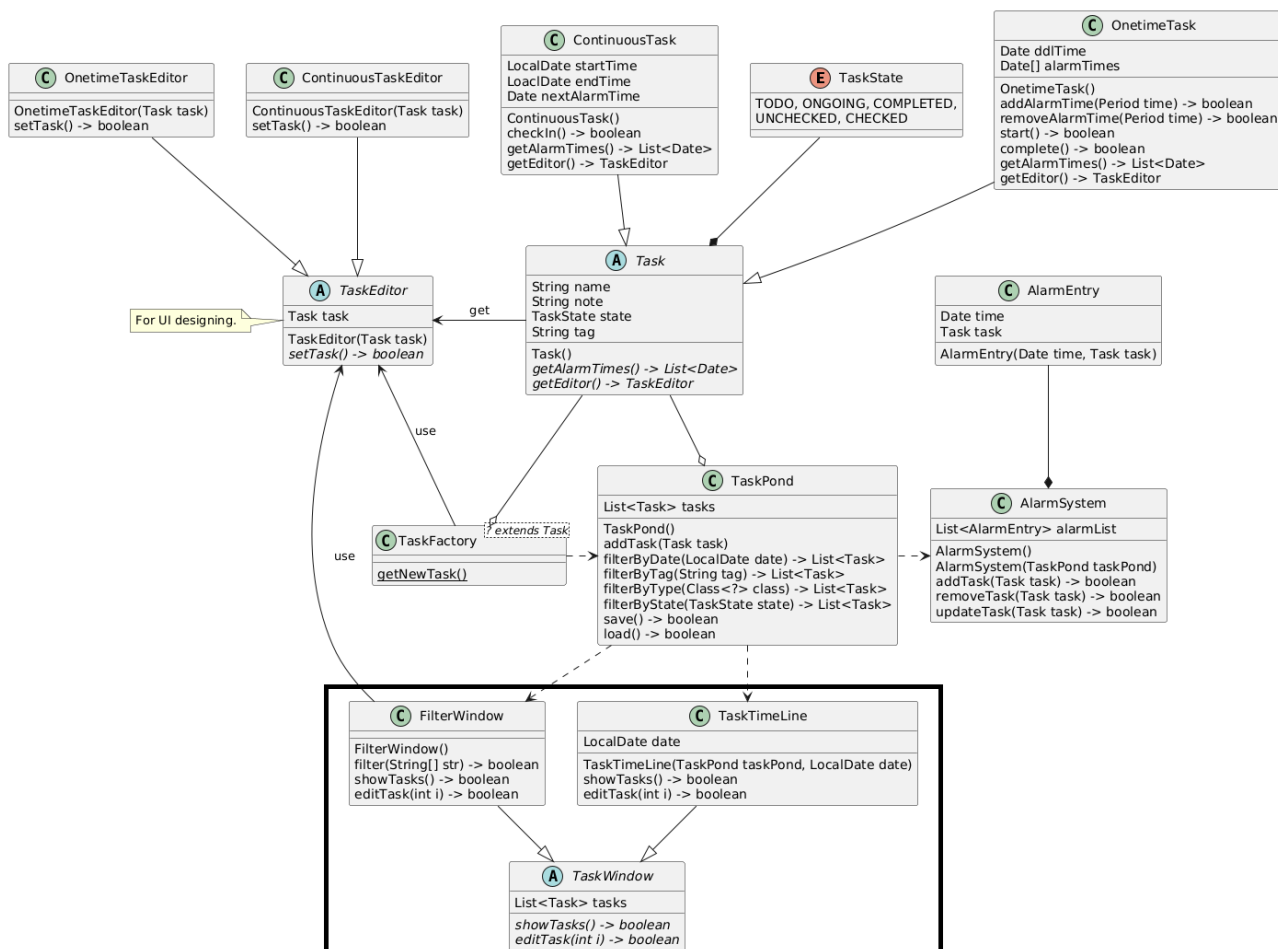
## 并发

通过 ScheduledExecutorService 来实现消息通知，它会在倒计时到达时触发另一个线程执行设定的方法。因此需要处理并发问题。

分析消息通知方法，它和主线程只共用一个数据结构：alarmList。（它按序存放着所有需要通知的时间以及对应的任务）

因此，采用一把大锁保平安的策略：将 alarmList 设置为线程安全 List，并且在所有对它的复杂操作前后上锁。

## 2.7 Window



Window 充当了和用户交互的主要对象，包含大量涉及交互的繁琐代码。

它属于包 `com.github.kyhsdj.tasklist.ui.window`。

Window 提供方法 `windowLoop`，通过接收参数 `TaskPond` 来提供和 `TaskPond` 交互的方式。包含：增删任务，筛选任务，直接对 Window 中的任务进行编辑。（任务展示时按重要程度排序）

## 2.8 SimpleLoop

基于上面的对象，最简洁地实现的任务单软件。

包含两个界面 `filterWindow`, `timelineWindow` 和一个数据库 `taskPond`。（数据库默认构造一个 `AlarmSystem` 并指向它）用户可以选择通过 `filterWindow` 还是 `timelineWindow` 和 `taskPond` 交互。

## 3 代码风格

选用的 IDE IntelliJ IDEA 会通过即时的静态分析警告所有代码可能出现漏洞的地方。

此外，它也会对代码风格进行约束，确保代码的可读性。（否则同样抛出警告）

## 4 LLM 使用

通过 Copilot 协助代码的开发，主要在三个方面帮助了我：

- CLI

帮助我快速完成了 Java 的命令行交互相关的代码

- LocalDateTime

任务单软件中涉及大量有关时间的计算，Copilot 为我推荐了 LocalDateTime 等数据类型，并且给出了各种场景的语句样例，帮助我快速学会了这个数据结构

- 并发

Copilot 帮助我了解到 Java 中的并发如何实现，并根据我的情况给出了恰当的代码样例

面对 LLM 不佳的生成结果，我一般采用以下方法

- 清除上下文重新提问
- 在提问中添加”Let’s think step by step”，来促进它 chain of thought

## 5 Github 上传情况

Fix task: Ensure it won't exist in two ponds at the same tim	origin & main	ZiyanYang	Today 22:11
Implement windows		ZiyanYang	Today 21:08
Refactor: Move all files to new package com.kyhdsjq.github.tasklist		ZiyanYang	Today 12:15
Refactor: Make AlarmSystem package-private & Optimize alarm time handli		ZiyanYang	Today 00:17
Implement concurrent alarm function & Optimize getAlarmTimes()		ZiyanYang	Yesterday 21:54
Implement TaskPond & AlarmSystem (alarming function pending)		ZiyanYang	Yesterday 20:08
Implement CLI & TaskFactory		ZiyanYang	Yesterday 00:05
Implement TaskEditor package (UI interaction pending)		ZiyanYang	2024/11/27 17:36
Implement Task package (interaction with alarm system & task editor pendi		ZiyanYang	2024/11/27 16:26
Create java project structure		ZiyanYang	2024/11/26 22:38
Add docs		ZiyanYang	2024/11/26 22:21
Initial commit		杨子彦*	2024/11/26 22:05

详情见 <https://github.com/kyhdsjq/task-list/commits/main/>

## 6 运行结果

### 6.1 创建任务

在无任务的情况下创建任务：



```

No tasks yet.

Choose an action: [filter, add, remove, edit, exit]
add
Which type of task do you want? [onetime, continuous]
onetime
Current name: "task", would you like to modify it? [y, n]
y
What's your task name?
study
Current note: "", would you like to modify it? [y, n]
y
What's your task note?
study
Current tag: "default", would you like to modify it? [y, n]
n
Current state: "TODO", would you like to modify it? [y, n]
n
Current ddl time: "2024-11-29T23:42:14.420863400", would you like to modify it? [y, n]
n
No alarm times yet.
Would you like to add alarm time? [y, n]
n

```

结果如下，新任务产生：

```

study
Current tag: "default", would you like to modify it? [y, n]
n
Current state: "TODO", would you like to modify it? [y, n]
n
Current ddl time: "2024-11-29T23:42:14.420863400", would you like to modify it? [y, n]
n
No alarm times yet.
Would you like to add alarm time? [y, n]
n

Onetime tasks:

--- 0 ---
name: study
note: study
tag: default
state: TODO
ddlTime: 2024-11-29T23:42:14.420863400

No continuous tasks yet.

Choose an action: [filter, add, remove, edit, exit]

```

## 6.2 修改任务

在上面的情况下修改任务：

```

Choose an action: [filter, add, remove, edit, exit]
edit
Input index of the task you want to edit.
0
Current name: "study", would you like to modify it? [y, n]
y
What's your task name?
study2
Current note: "", would you like to modify it? [y, n]
n
Current tag: "default", would you like to modify it? [y, n]
n
Current state: "TODO", would you like to modify it? [y, n]
n
Current ddl time: "2024-11-29T23:48:44.049515800", would you like to modify it? [y, n]
y
Please enter the year:
2024
Please enter the month:
11
Please enter the day:
29
Please enter the hour:
23
Please enter the minute:
52
No alarm times yet.

```

修改了名字，并且新增了提醒时间

```

Please enter the number of hours:
0
Please enter the number of minutes:
2
Current alarm times:
0. 2024-11-29T23:50
Would you like to remove any of them? [y, n]
n
Current alarm times:
0. 2024-11-29T23:50
Would you like to add alarm time? [y, n]
n

Onetime tasks:

--- 0 ---
name: study2
note:
tag: default
state: TODO
ddlTime: 2024-11-29T23:52

No continuous tasks yet.

Choose an action: [filter, add, remove, edit, exit]

```

## 6.3 任务提醒

稍等一会儿，提醒消息发来了

Alarm! Notice the following tasks:

--- 0 ---

name: study2

note:

tag: default

state: TODO

ddlTime: 2024-11-29T23:52