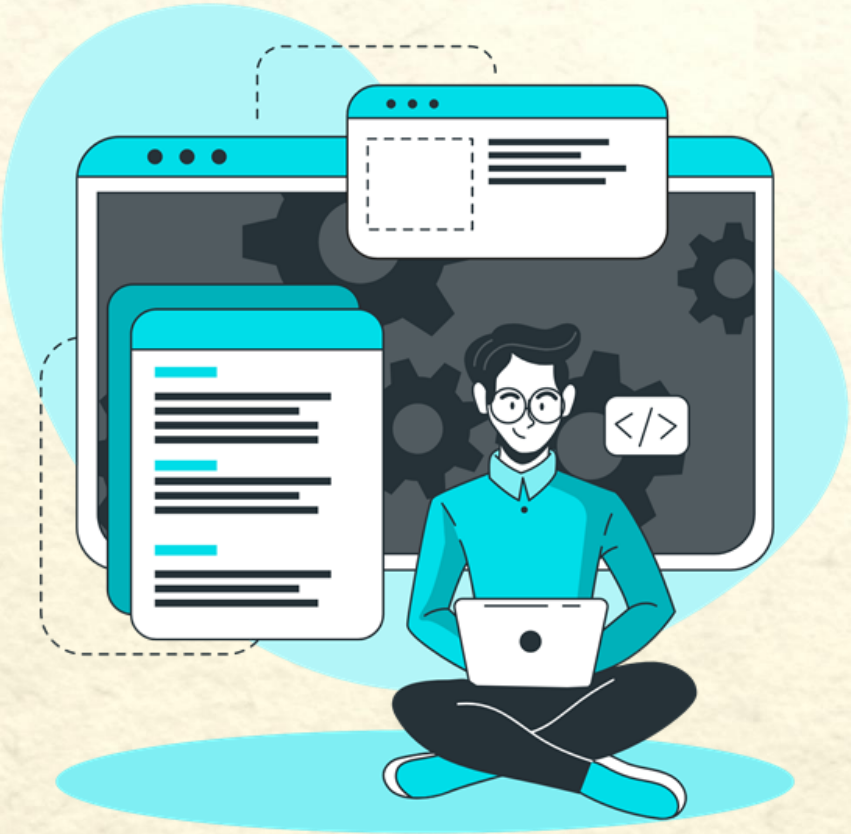
مکتب شریف

اولین بوتکمپ آموزشی - استخدامی ایران

مـــکـــتب 137

سری دوازدهم



شما قرار است یک سیستم مدیریت یادآور بسازید که بتواند انواع مختلف یادآورها را ذخیره، مدیریت و اجرا کند.

هر یادآور اطلاعاتی مثل عنوان، زمان انجام و نوع یادآور دارد.

هر یادآور همچنین باید یک شناسه (ID) اختصاصی داشته باشد که توسط یک generator ساخته می‌شود.

سیستم باید تمام فعالیت‌ها (ساخت، حذف، اجرا و خطاها) را در یک فایل log ذخیره کند.

در سیستم سه نوع یادآور وجود دارد:

یادآور ساده(SimpleReminder)

مثال:

ساعت ۵ زنگ بزن اقای مبارز

یادآور جلسه(MeetingReminder)

ویژگی اضافی:

- نام شرکت‌کننده‌ها(participants)

مثال:

جلسه با تیم — شرکت‌کننده‌ها: علی، سارا

یادآور کار روزانه(DailyRoutineReminder)

ویژگی اضافی:

- تکرار روزانه داشته باشه یا نه True/False :

مثال:

ورزش — تکرار روزانه فعال

همه یادآورها ویژگی‌های مشترک دارند:

ویژگی	توضیح
title	عنوان یادآور
time	زمان انجام یادآور (به صورت رشته ساده: "18:30")
reminder_id	شناسه یکتا که با generator ساخته می‌شود

کلاس ها با @dataclass ساخته می‌شود.

---

generator وظیفه ساخت ID یکتا را دارد

هر بار یک یادآور جدید ساخته می‌شود، IDجدید از generator دریافت می‌شود.

هر نوع یادآور باید متد remind() مخصوص خودش را داشته باشد.

مثال‌ها:

SimpleReminder:

It is time: خرید نان

MeetingReminder:

Meeting Reminder: جلسه با تیم → افراد: سارا، حمید

DailyRoutineReminder:

Daily Routine: ورزش (تکرار روزانه فعال)

وقتی سیستم **execute_all()** را صدا بزند، تمام **remind** ها با پلی‌مورفیسم اجرا می‌شوند.

ReminderManager:

یا همان "مدیریت‌کننده یادآورها"

قابلیت‌ها:

- **add_reminder**
- **remove_reminder**
- **list_reminders**
- **reminder_group** (این بخش امتیازی است)
- **execute_all()** برای اجرای **polymorphic**
- پیدا کردن یادآور با **id**

فایل لاگ:

reminder.log

باید ثبت شود:

→ اضافه شدن یادآور INFO

→ حذف یادآور WARNING

→ اجرای یادآور INFO

→ خطاها مثل زمان خالی یا عنوان خالی ERROR

GitHub :

- یک repo بسازید با توجه به نام پروژه
- حداقل 5 commit معنی‌دار داشته باشید که یکی از آنها اشتباه تایپی داشته باشد و کامیت را تغییر دهید
- README بنویسد که شامل:
  - توضیح پروژه
  - نحوه اجرا
  - نمونه خروجی
- بعد از اینکه نسخه اول پروژه را کامل کردید و Release با برچسب (Tag) زیر را ساختید:

`v1.0.0`

باید یک نسخهٔ جدید منتشر کنید از پروژه که شامل دو قابلیت جدید و خلاقانه باشد.(ایده های خلاقانه نمره اضافی دارند) این قابلیت‌ها باید کاملاً از طرف خود شما طراحی شوند!

- یک Release جدید در GitHub منتشر کنید با tag :

v2.0.0

داخل توضیحات Release بنویسید:

- قابلیت‌های جدید چی هستند
- چرا این‌ها را انتخاب کردید
- چه مشکلی را حل می‌کنند

بخش امتیازی-اجباری... :)

پیاده‌سازی Log Rotation برای فایل‌های لاگ

لاگ‌های پروژه باید به صورت (rotate)پیدا کنند یعنی وقتی یکی از شرایط زیر بر قرار شد خودکار چرخش

فایل فعلی بسته شده و یک فایل جدید ایجاد شود:

- **رسیدن فایل لاگ به یک حجم مشخص مثلاً 100KB**
- **ایجاد یک فایل جدید بر اساس زمان (مثلاً هر روز یا هر ساعت)**

(.انتخاب یکی از دو روش کافی است)

برای پیاده‌سازی این قابلیت باید یکی از Handler های زیر را تحقیق کرده و استفاده کنید:

- **RotatingFileHandler**
- **TimedRotatingFileHandler**

تنظیمات باید به گونه‌ای باشد که:

- **هنگام رسیدن به حد حجم یا زمان، یک فایل لاگ جدید ساخته شود.**
- **فایل‌های قدیمی با شماره یا تاریخ نگهداری شوند.**

همچنین بهتره یاداور ها رو از کاربر بگیرید و دستی وارد نکنید.

نکات:

- مهلت ارسال تمرین تا پایان روز پنجشنبه **13 / 09 / 1404** است
- پاسخ تمرین را در **سامانه مودل** ارسال کنید
- نام فایل ارسالی خود را به این صورت قرار دهید : Name_hw12_maktab137

به عنوان مثال Mahdi_Zolfaghari_hw12_maktab137

- در صورتی که تمرین شامل چند فایل یا فولدر میباشد **حتما** آن را در قالب یک فایل فشرده شده تجمیع کنید.