

# پروژه نهایی

## ساختمان داده‌ها



استاد مهیار حسن‌پور

دانشگاه صنعتی نوشیروانی

ترم 4032

در این پروژه قراره یک پیام‌رسان ساده اما دقیق و مهندسی‌شده طراحی کنیم که تمرکز اصلی اون روی استفاده از مفاهیم درس ساختمان داده‌هاست. این پیام‌رسان شاید در ظاهر ساده باشه، اما پشتش مفاهیم مهمی از دنیای ساختمان داده‌ها قرار داره که قراره با دست خودمون، بدون استفاده از کتابخانه‌های آماده، اون‌ها رو پیاده‌سازی کنیم.

پروژه نه تنها به ما کمک می‌کنه تا با ساختارهایی مثل استک، درخت جستجوی دودویی، لیست پیوندی یک‌طرفه و دیکشنری (هش‌تبل) آشنا بشیم، بلکه به ما یاد می‌ده چطور می‌شه این مفاهیم رو در یک پروژه‌ی واقعی به‌کار گرفت.

هدف از انجام این پروژه:

- آشنایی عملی با ساختمان‌های داده از پایه
- یادگیری کاربرد این ساختارها در یک سیستم واقعی
- طراحی منطقی و ساختارمند یک اپلیکیشن کاربردی
- افزایش مهارت در طراحی الگوریتم و مدیریت داده‌ها
- درک بهتر تفاوت بین «استفاده‌کننده» و «طراح» نرم‌افزار

## نحوه انجام پروژه

این پروژه می‌تونه به صورت فردی یا دو نفره انجام بشه. هر تیم (تک‌نفره یا دونفره) مسئول تمام بخش‌های پروژه خواهد بود؛ از طراحی ساختمان داده گرفته تا ساخت رابط کاربری و ذخیره‌سازی اطلاعات.

## ویژگی‌های کلی پروژه

در ادامه، نگاهی داریم به مهم‌ترین ویژگی‌هایی که این پیام‌رسان باید داشته باشه:

## مدیریت پیام‌ها با استک (Stack)

- پیام‌ها در قالب یک استک نگهداری می‌شن.
- پیام جدید روی استک قرار می‌گیره (عملیات push).
- برای مشاهده آخرین پیام‌ها، از pop استفاده می‌شه.
- هدف: نمایش سریع و مرتب پیام‌های اخیر.

## جستجوی پیام‌ها با درخت جستجوی دودویی (BST)

- هر پیام یک کلید (مثل ID یا زمان ارسال) دارد.
- پیام‌ها به صورت مرتب در BST ذخیره می‌شوند.
- جستجوی سریع بر اساس کلید، با پیچیدگی زمانی مناسب.
- قابلیت جستجو، حذف، و نمایش پیام‌ها به صورت مرتب.

## ریپلای به پیام‌ها با لیست پیوندی یک‌طرفه

- هر پیام می‌تونه پاسخی داشته باشه.
- این پاسخ‌ها به صورت زنجیره‌ای در یک لیست پیوندی ذخیره می‌شوند.
- ساختار یک‌طرفه باعث سادگی و عملکرد بهینه برای نمایش پاسخ‌ها به ترتیب ورود می‌شه.

## دیکشنری (Hash Table) برای ذخیره کاربران

- هر کاربر با یک کلید (مثل نام کاربری یا ID) در دیکشنری نگهداری می‌شه.
- دسترسی سریع به اطلاعات کاربران با استفاده از تابع هش.
- پیاده‌سازی کامل از صفر، شامل مدیریت برخورد (collision resolution).

## ذخیره‌سازی اطلاعات در فایل یا پایگاه داده

- اطلاعات پیام‌ها، کاربران و پاسخ‌ها باید ذخیره و بازیابی بشن.
- انتخاب بین فایل متنی (JSON)، CSV یا پایگاه داده سبک مثل SQLite با تیم پروژه‌ست.
- در زمان اجرای مجدد برنامه، داده‌ها از حافظه بارگذاری می‌شوند.

## رابط گرافیکی برای تعامل کاربر

- کاربر باید بتواند با برنامه ارتباط راحتی داشته باشد.
- فرم‌هایی برای ورود، ثبت‌نام، ارسال پیام، مشاهده لیست پیام‌ها و کاربران، ریلی‌ای و جستجو طراحی می‌شود.
- رابط کاربری می‌تواند با ابزارهایی مثل Tkinter در Python یا JavaFX ساخته بشود.

## پیاده‌سازی از پایه (بدون استفاده از ساختارهای آماده)

- هیچ‌کدام از ساختمان داده‌ها نباید از کلاس‌های آماده زبان برنامه‌نویسی استفاده کنند (مثل `list`, `dict`, `stack` و ... در Python).
- پیاده‌سازی کاملاً دستی انجام می‌شود تا مفاهیم پشت ساختارها بهتر درک بشود.

## زبان و ابزارها

- زبان برنامه‌نویسی: به انتخاب تیم (Python، Java، C++)
- رابط گرافیکی: (GUI)

○ Tkinter در Python

○ JavaFX در Java

○ Qt یا SDL در C++

- ذخیره‌سازی:

○ فایل متنی (JSON/CSV)

○ یا پایگاه داده ساده مثل SQLite

## ساختار کلی پروژه

```
project/
├── data_structures/
│   ├── stack.py
│   ├── bst.py
│   ├── linked_list.py
│   └── hashtable.py
├── models/
│   ├── message.py
│   └── user.py
├── storage/
│   └── storage_handler.py
├── ui/
│   └── main_window.py
├── main.py
└── README.md
```

## توضیح ساختمان داده‌ها

در این بخش دقیق‌تر به ساختمان داده‌هایی که توی پروژه استفاده کردیم می‌پردازیم:

## استک – (Stack) مدیریت پیام‌های اخیر

چی هست؟

استک یه ساختار LIFO هست؛ یعنی آخرین چیزی که وارد می‌شه، اولین چیزیه که خارج می‌شه.

چرا توی پیام‌رسان؟

برای دیدن جدیدترین پیام‌ها به ترتیب ورود، عالیه! مثلاً وقتی یه کاربر چت رو باز می‌کنه، آخرین پیام‌ها رو از بالای استک می‌خونه.

عملیات‌هایی که پیاده‌سازی کردیم:

- `push(message)`
- `pop()`
- `peek()`
- `is_empty()`

## درخت جستجوی دودویی – (BST) جستجوی سریع پیام‌ها

چی هست؟

یه درخت که هر گره‌اش یه مقدار داره، مقادیر کوچکتر سمت چپ، بزرگ‌ترها سمت راست.

چرا توی پیام‌رسان؟

برای جستجوی سریع پیام‌ها مثلاً بر اساس timestamp یا message ID کاربردی.

عملیات:

- `insert(message)`
- `search(id)`
- `delete(id)`
- `traverse()` (به روش in-order برای مرتب‌سازی پیام‌ها)

## لیست پیوندی یک‌طرفه – ریپلای به پیام‌ها

چی هست؟

یه ساختار زنجیره‌ای که فقط از اول به بعد می‌تونی بری. هر گره فقط یه اشاره‌گر داره به گره بعدی.

چرا توی پیام‌رسان؟

برای نمایش پاسخ‌های یه پیام به ترتیب زمانی. مثلاً کاربر به یه پیام جواب می‌ده، بعد یکی دیگه به اون، همه به صورت زنجیره‌وار وصل می‌شن.

عملیات:

- `add_reply(reply)`
- `get_all_replies()`

## دیکشنری – (Hash Table) ذخیره و مدیریت کاربران

چی هست؟

یه ساختار که کلید و مقدار رو نگه می‌داره. سریع و کاربردی برای پیدا کردن اطلاعات بر اساس یک کلید خاص.

چرا توی پیام‌رسان؟

برای نگه‌داشتن لیست کاربران، جستجوی سریع با username یا ID.

عملیات:

- `insert(key, user)`
- `get(key)`
- `delete(key)`
- `hash_function(key)` برای تعیین مکان داده در جدول

ذخیره‌سازی اطلاعات

برنامه باید همه چیز رو ذخیره کنه تا بعد از خروج دوباره بتونه اطلاعات رو بازیابی کنه. ذخیره‌سازی می‌تونه یکی از این دو حالت باشه:

- **فایل متنی (JSON یا CSV):** ساده، قابل مشاهده برای انسان
- **پایگاه داده SQLite:** ساخت یافته، مناسب برای حجم بالاتر

اطلاعات ذخیره‌شونده:

- لیست کاربران
- پیام‌ها
- ارتباط بین پیام‌ها و پاسخ‌ها

توجه شود انتخاب مکانیزم ذخیره‌سازی به عهده خودتان است و در انتخاب آن آزاد هستید.

## رابط کاربری گرافیکی (GUI)

یه برنامه بدون رابط کاربری یعنی فقط برنامه‌نویس می‌تونه باهاش کار کنه! برای همین، رابط گرافیکی ساده و کاربرپسند طراحی می‌شه.

امکانات رابط:

- ثبت‌نام و ورود کاربران
- ارسال پیام
- مشاهده پیام‌ها
- جستجوی پیام‌ها
- ریپلای به پیام‌ها
- دیدن لیست کاربران

امیدواریم این پروژه، فرصتی باشه تا خودتون رو به چالش بکشید و توانایی‌هاتون در طراحی و پیاده‌سازی سیستم‌های واقعی رو محک بزنید. شاید در نگاه اول پیاده‌سازی دستی ساختمان داده‌ها سخت به نظر برسه، اما مطمئن باشید همین مسیر پرچالش، شما رو به یک مهندس نرم‌افزار دقیق، باهوش و خلاق تبدیل می‌کنه.

با دقت، صبر و خلاقیت جلو برید؛ از اشتباه کردن نترسید و از یاد گرفتن لذت ببرید. فراموش نکنید که این پروژه فقط یک نمره نیست، بلکه قدمی بزرگ به سمت حرفه‌ای شدن در دنیای واقعی برنامه‌نویسیه.

برای همتون آرزوی موفقیت داریم – به خودتون باور داشته باشید و با انگیزه تا انتها ادامه بدید!