

## عنوان پروژه: سیستم پرسش و پاسخ

در این پروژه یک سیستم پرسش و پاسخ به همراه کاربران و سطوح دسترسی با مشخصات زیر باید توسعه پیدا کند:

### سوالات:

سوالات به صورت چهار گزینه‌ای و یا تشریحی ذخیره‌سازی می‌شوند. برای مشخص کردن این دو نوع از سوالات دو کلاس FourChoice و Descriptive که فرزند کلاسی به نام Question هستند باید تعریف شوند.

### کلاس Question

این کلاس اعضای داده‌ای محافظت شده زیر وجود دارد:

- question: از نوع string است که متن سؤال می‌باشد.
- createdAt: زمان ثبت سؤال است و متغیری از نوع ساختار DateTime است.
- type: نوع سؤال از جنس رشته که descriptive یا four-choice است.
- user: یک شیء از کلاسی به نام User است. این متغیر کاربری که سؤال را پرسیده ذخیره می‌کند.
- tags: یک آرایه از اشاره‌گرها به کلاسی به نام Tag که مقدار پیش‌فرض آن NULL است.
- isPublished: متغیری دودویی که مشخص می‌کند سؤال منتشر شده است یا خیر. مقدار پیش‌فرض false است.

توابع زیر در این کلاس وجود دارد:

- تابع سازنده که مقادیر question, createdAt, user, type را دریافت می‌کند و به متغیرها انتساب می‌دهد.
- تابع publish که سؤال را منتشر می‌کند.
- تابع unpublish که سؤال را از حالت انتشار خارج می‌کند.
- تابع خالص مجازی print که سؤال و همه مشخصات آن را چاپ می‌کند.
- تابع خالص مجازی printAll که سؤال و نوع و شناسه (ایندکس آرایه) آن را چاپ می‌کند.
- تابع addTag با امضای زیر

```
void addTag(Tag*);
```

که یک tag را به آرایه تگ‌های سوال اضافه می‌کند. اگر این تگ ورودی قبلاً در آرایه تگ‌ها وجود داشته باشد نباید مجدداً وارد شود.

## کلاس Descriptive

این کلاس اعضای داده‌ای خصوصی زیر وجود دارد:

- answer: پاسخ سؤال از نوع string

توابع زیر در این کلاس وجود دارد:

- تابع سازنده که مقادیر question, createdAt, user را دریافت می‌کند و تنها سازنده والد را فراخوانی می‌کند. مقدار نوع را descriptive قرار دهید.
- تابع addAnswer که یک string دریافت می‌کند و پاسخ را ثبت می‌کند.
- پیاده‌سازی تابع print.
- پیاده‌سازی تابع printAll.
- تابع استاتیک create با امضای زیر که یک اشاره‌گر به کلاس خودش را برمی‌گرداند. کاربری می‌تواند چنین سؤالی ایجاد کند که لاگین کرده باشد و سطح دسترسی add-descriptive-question را داشته باشد.  
`Descriptive* create(string question, DateTime createdAt, User user);`  
در این تابع یک شیء (که با فراخوانی تابع سازنده انجام می‌شود) بسازید و آدرس به آن را برگردانید.
- تابع edit با امضای زیر که یک اشاره‌گر به کلاس خودش را برمی‌گرداند. کاربری می‌تواند چنین سؤالی ویرایش کند که لاگین کرده باشد و سطح دسترسی edit-descriptive-question را داشته باشد.  
`Descriptive* edit(string question, DateTime createdAt, User user);`  
در این تابع اعضای داده‌ای آبجکت را با مقادیر ارسال شده آپدیت کنید.

## کلاس FourChoice

این کلاس اعضای داده‌ای خصوصی زیر وجود دارد:

- A: متن گزینه A از نوع string

- B: متن گزینه B از نوع string

- C: متن گزینه C از نوع string

- D: متن گزینه D از نوع string

- answer: از نوع char که یکی از مقادیر 'A', 'B', 'C', 'D' است.

توابع زیر در این کلاس وجود دارد:

- تابع سازنده که مقادیر question, createdAt, user, A, B, C, D, answer را دریافت می‌کند، چهار گزینه و answer را مقداردهی کرده و سازنده والد را فراخوانی می‌کند. مقدار نوع را four-choice قرار

دهید. کاربری می تواند چنین سؤالی ایجاد کند که لاگین کرده باشد و سطح دسترسی add-four-choice-question را داشته باشد.

- پیاده سازی تابع print.
- پیاده سازی تابع printAll.
- تابع استاتیک create با امضای زیر که یک اشاره گر به کلاس خودش را برمی گرداند:  
FourChoice\* create(string question, DateTime createdAt, User user, string A, string B, string C, string D, char answer);  
در این تابع یک شیء (که با فراخوانی تابع سازنده انجام می شود) بسازید و آدرس به آن را برگردانید.
- تابع edit با امضای زیر که یک اشاره گر به کلاس خودش را برمی گرداند:

FourChoice\* edit(string question, DateTime createdAt, User user, string A, string B, string C, string D, char answer);  
در این تابع اعضای داده ای آبجکت را با مقادیر ارسال شده آپدیت کنید. کاربری می تواند چنین سؤالی ویرایش کند که لاگین کرده باشد و سطح دسترسی edit-four-choice-question را داشته باشد.

## کلاس User

این کلاس اعضای خصوصی

- string name
- string username
- string password
- Permission \*permissions[۱۰۰] = {NULL}

را دارد. پارامتر چهارم یک آرایه از اشاره گرها از دسترسی های کاربر است که مقدار اولیه آن NULL است. توابع زیر را پیاده سازی کنید:

- یک تابع سازنده که سه پارامتر را دریافت کرده و با فهرست مقداردهی، مقداردهی می کند.
- تابع استاتیک create که سه پارامتر را دریافت می کند و یک اشاره گر به User برمی گرداند:

User\* create(string, string, string);

این تابع پس از ساخت کاربر، آن را در یک آرایه سراسری از اشاره گرها به کاربران قرار می دهد. توجه کنید کسی می تواند این تابع را فرخوانی کنی که لاگین کرده باشد و دسترسی add-user داشته باشد. در ابتدای برنامه یک کاربر با نام کاربری admin و پسورد ۱۲۳۴۵۶ و تمام دسترسی ها تعریف کنید و با آن لاگین کنید.

- تابع عمومی addPermission با امضای زیر که یک دسترسی را به کاربر اضافه می کند:

void addPermission(Permission \*permission);

و اگر دسترسی تکراری باشد نباید به لیست اضافه شود.

- تابع عمومی print برای چاپ مشخصات کاربر و تمام دسترسی‌های آن بنویسید.
- تابع عضو عمومی checkAuth که پارامترهای username, password دریافت می‌کند و اگر نام کاربری با کلمه عبور تطابق داشته باشد مقدار true وگرنه false برمیگرداند.

Bool checkAuth(string, string);

## کلاس Tag

این کلاس عضو خصوصی

string title

را دارد. برای این کلاس یک سازنده با فهرست مقداردهی، تابع استاتیک create برای ایجاد شیء

Tag\* create(string);

و تابع print برای چاپ آن بنویسید. سپس تگ به آرایه سراسری اضافه شود. تابع printAll برای چاپ همه آنها نیز بنویسید.

## کلاس Permission

این کلاس عضو خصوصی

- string title(string)

را دارد. برای این کلاس یک سازنده با فهرست مقداردهی، تابع استاتیک create برای ایجاد شیء

Permission\* create(string);

و تابع print برای چاپ آن بنویسید. سطح دسترسی‌های

- add-descriptive-question
- add-four-choice-question
- edit-descriptive-question
- edit-four-choice-question
- add-user

را تعریف و در آرایه سراسری از اشاره‌گرها به دسترسی‌ها قرار دهید. یک تابع دستیابی نیز برای برگرداندن عنوان بنویسید.

## کلاس Auth

این کلاس کاربری که لاگین کرده باشد را در خود نگه می‌دارد و یک عضو خصوصی استاتیک

User\* auth

بدین منظور استفاده می‌شود. این کلاس تابع استاتیک login با امضای زیر را دارد

User\* login(string username, string password);

درون این تابع، با یک حلقه روی آرایه‌ای از اشاره‌گرها به تمام کاربران، و فراخوانی تابع `checkAuth` بررسی می‌شود که آیا نام کاربری و کلمه عبور متعلق به کاربری است یا خیر. اگر باشد، آدرس آن کاربر درون متغیر `auth` قرار گرفته و سپس آدرسش نیز برگردانده می‌شود. این کلاس تابع استاتیک `logout` نیز دارد که مقدار `auth` را `NULL` می‌کند.

```
void logout();
```

تابع استاتیک `whoami` نیز وجود دارد که کاربر فعلی لاگین کرده را برمی‌گرداند:

```
User* whoami();
```

آرایه‌های سراسری زیر را بسازید:

```
User* users[١٠٠] = {NULL};
```

```
Question* questions[١٠٠] = {NULL};
```

```
Permission* permissions[١٠٠] = {NULL};
```

```
Tag* tags[١٠٠] = {NULL};
```

و پس از ثبت هر داده‌ای، آن را به آرایه مربوطه اضافه کنید.

## ساختار `DateTime`

این ساختار اعضای `year, month, day, hour, minute, second` از نوع `int` دارد.

# منوی برنامه

## Main Menu

*If user is not logged in:*

- Login
- View Questions

*If user is logged in:*

-----  
*Hello Foo bar!*  
-----

- Question Menu
- Tag Menu
- User Menu

## Question Menu

- List of Descriptive Questions with ID
- Create Descriptive Question
- Edit Descriptive Question
- Print Descriptive Question
- Create FourChoice Question
- Edit FourChoiceQuestion
- List of FourChoice Questions with ID
- Print FourChoice Question

## Tag Menu

- List of Tags with ID
- Create new Tag

## User Menu

- List of Users with ID
- Create new User
- Add Permission to a User

