

تمرین برنامه نویسی شی گرا (برای این تمرین مفاهیم کلاس ها و regex را مطالعه کنید)

در این تمرین قصد داریم که به کمک مفاهیم کلاس ها در پایتون یک اپلیکیشن مدیریت سیستم فروش سایت طراحی کنیم که کاربران بتوانند در آن لاگین کنند و محصولات خود را از سایت های مختلف سفارش دهند. برای این منظور شما باید سه کلاس با نام های `Site` ، `Account` و `Product` طراحی کنید. در این هفته صرفا کلاس های `Site` و `Account` را طراحی میکنیم و در هفته های آتی فانکشنالیتی های بیشتری به این مجموعه اضافه خواهیم کرد. طراحی شما باید به نحوی باشد که به مرور بتوانیم در یک ساختار ماژولار اپلیکیشنمون را بهینه و بهینه تر کنیم

کلاس هایی که در این هفته طراحی میکنید دارای ویژگی ها و متدهای زیر هستند:

کلاس Account:

هر شی این کلاس دارای چهار ویژگی (Attribute) است که در تابع `__init__` مقداردهی میشوند و برای هر شی مقدار آن فرق میکند:

- **یوزرنیم username:** یوزرنیم کاربر باید به فرمت (نام + آندرلاین + نام خانوادگی) ذخیره شود. توجه کنید که نام و نام خانوادگی فقط باید از حروف انگلیسی باشند.
در صورتی که یوزرنیم دریافتی مانند فرمت بالا نبود باید `Exception` با متن ارور `raise invalid username` کنید.
- **پسورد password:** پسورد کاربر باید دارای نکات زیر باشد:
 - طول آن حداقل ۸ کاراکتر باشد.
 - حتما شامل یک حرف بزرگ، یک حرف کوچک، یک عدد باشد.
 - باید حتما به صورت هش (hash) با فرمت `sha256` و `encode utf-8` ذخیره شود. در صورتی که پسورد دریافتی مانند فرمت بالا نبود باید یک `Exception` با متن ارور `raise invalid password` کنید.

مثال:

- اگر پسورد برابر با `Aa123123` بود مشکلی ندارد و صحیح است.
- ولیکن اگر پسورد برابر با `valid aa123123 AA123123 Aa12312` نیست.
- **شماره موبایل phone number:** شماره تلفن کاربر ممکن است به دو فرمت به شما داده شود:

۱- `989xxxxxxxxx+`

۲- `09xxxxxxxxx`

اگر شماره صحیح بود در هر حالتی شما باید آن را به صورت `09xxxxxxxxx` ذخیره کنید،

و در صورتی که شماره موبایل دریافتی مانند فرمت بالا نبود باید یک `Exception` با متن ارور `raise invalid phone number` کنید.

مثال:

- اگر شماره ی موبایل برابر با `09121212121` باشد مشکلی ندارد.
- ولیکن اگر شماره ی موبایل برابر با `08121212121` , `0912121212` , `988121212121+` باشد `valid` نیست.

- **ایمیل Email:** تضمین می شود ایمیل ورودی به فرمت `First_Part@Second_Part.Third_Part` به شما داده می شود.

ایمیل کاربر باید صحیح باشد، ایمیلی را صحیح می نامیم اگر و فقط اگر:

- `First_Part` و `Second_Part` تنها شامل حروف بزرگ و کوچک، اعداد، `_`، `-`، `.` باشد و تضمین می‌شود نیازی به چک کردن محدودیت طولی بخش اول و دوم ندارید.
- `Third_Part` باید تنها شامل حروف بزرگ و کوچک باشد و طولی میان ۲ تا ۵ کاراکتر داشته باشد.

در صورتی که ایمیل دریافتی مانند فرمت بالا نبود باید یک *Exception* با متن ارور `raise invalid email` کنید.

کلاس Site:

هر شی این کلاس دارای سه ویژگی (Attribute) است که در تابع `__init__` مقدار دهی می‌شود و به ازای هر شیء مقدار آن فرق دارد:

1. آدرس سایت `url`

تضمین می‌شود آدرس سایت صحیحی به شما داده می‌شود و نیازی به چک کردن آن ندارید.

2. افراد ثبت نام شده `register_users`

در هنگام ساخت شیء این مقدار را برابر با یک لیست خالی قرار دهید.

3. افراد فعال در سایت `active_users`

در هنگام ساخت شیء این مقدار را برابر با یک لیست خالی قرار دهید.

این کلاس همچنین دارای سه متد زیر است:

• متد `register`:

کلاس سایت باید دارای تابعی به نام `register` باشد. این تابع یک شیء از کلاس `Yوزر` ورودی می‌گیرد و اگر آن `Yوزر` در `register_users` خود سایت بود یک *Exception* با متن ارور `raise user already registered` کنید وگرنه آن را به لیست `register_users` خود سایت بیفزایید و عبارت `register successful` را برگردانید. توجه کنید که لیست `register_users` باید به ترتیب ثبت‌نام کاربران، آن‌ها را در خود ذخیره کند.

• متد `login`:

کلاس سایت باید دارای تابعی به نام `login` باشد. این تابع ورودی‌های متفاوتی می‌گیرد:

۱- ممکن است `email` و `username` و `password` به آن داده شود. در این حالت شما باید چک کنید یوزری در `register_users` سایت وجود دارد که هم یوزرنیم و هم ایمیلی برابر با ورودی داشته باشد و پسوردش برابر با هش شده‌ی پسورد ورودی باشد یا خیر.

۲- ممکن است `username` و `password` به آن داده شود. در این حالت شما باید چک کنید یوزری در `register_users` سایت وجود دارد که یوزرنیمی برابر با ورودی داشته باشد و پسوردش برابر با هش شده‌ی پسورد ورودی باشد یا خیر.

۳- ممکن است `email` و `password` به آن داده شود. در این حالت شما باید چک کنید یوزری در `register_users` سایت وجود دارد که ایمیلی برابر با ورودی داشته باشد و پسوردش برابر با هش شده‌ی پسورد ورودی باشد یا خیر.

- باید با توجه به این که ورودی تابع از کدام نوع است لاگین (*login*) را با اطلاعات ورودی تطبیق دهید.
- اگر کاربری در هر کدام از حالات بالا یافت شد باید آن شیء یوزر را به لیست `active_users` اضافه کنید و عبارت `login successful` را برگردانید.
- اگر در هر کدام از حالات قبل از تطبیق اطلاعات یوزر در لیست `active_users` موجود بود باید عبارت `user already logged in` را برگردانید.
- اگر اطلاعات ورودی برای هیچ کدام از کاربرهای `register_users` نبودند باید عبارت `invalid login` را برگردانید.

توجه کنید که لیست `active_users` باید به ترتیب ورود کاربران، آن ها را در خود ذخیره کند.

● **متد `logout`:**

کلاس سایت باید دارای تابعی به نام `logout` باشد. این تابع یک شیء از کلاس یوزر ورودی می گیرد و اگر آن یوزر در `active_users` خود سایت بود آن را از لیست حذف کرده و عبارت `logout successful` را برگردانید و در صورت عدم وجود آن یوزر در `active_users` سایت عبارت `user is not logged in` را برگردانید.