

این پروژه، اولین تلاش جدی شما برای مدل کردن یک مسئله با استفاده از مفاهیم طراحی شیء‌گرا<sup>۱</sup> است. از این رو توصیه می‌کنیم که پیش از شروع پیاده‌سازی، برای طراحی آن زمان کافی اختصاص دهید.

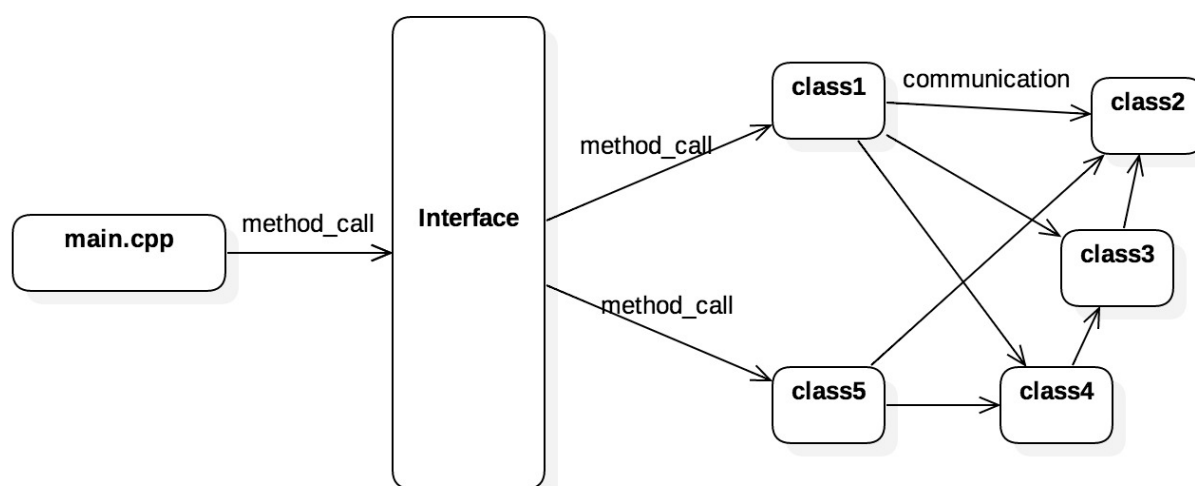
## کاریابی (لینکدین ملی)

قلی که چند ترم از شروع دانشگاهش می‌گذرد، برای ورود به بازار کار به مشکل می‌خورد و نمی‌تواند کاری مطابق با توانایی‌هایش پیدا کند. پس برآن می‌شود تا با کمک دوستانش و شما شبکه‌ای مجازی با عنوان لینکدین ملی (!) راه‌اندازی کند. هر کاربر پس از ثبت نام در این سامانه می‌تواند مواردی از جمله مهارت‌ها، سوابق کاری و تحصیلی خود را وارد کند، برای عناوین شغلی مختلف اقدام کند و مهارت‌های دوستانش را تایید کند.

**توجه:** در این تمرین نیازی به پیاده‌سازی تابع main نیست و فایل‌های شما باید با استفاده از تابع main آماده‌شده توسط دستیاران آموزشی قابل تست باشد. در این پروژه شما تنها باید متدهای به کار رفته در تابع main را پیاده‌سازی کنید. برای مثال به فایل main.cpp که در کنار صورت پروژه قرار گرفته توجه کنید.

## کلاس رابط<sup>۲</sup>

برای اینکه برنامه شما با تابع main مورد نظر ما قابل اجرا باشد، لازم است کلاسی با نام رابط داشته باشید که از آن برای مقداردهی به برنامه شما استفاده می‌کنیم. حالت کلی قرارگیری این کلاس به شکل زیر است. در main یک شیء از کلاس رابط ایجاد شده و برای ارتباط با برنامه شما متدهای این شیء، همراه با ورودی‌ها صدا زده می‌شوند. دقت کنید که این کلاس صرفاً رابط برنامه شما با تابع main است و به هیچ وجه نباید شامل منطق برنامه شما باشد؛ یعنی در پیاده‌سازی این توابع تنها باید توابعی از کلاس‌های دیگر را صدا بزنید و هرگز نباید جزئیات پیاده‌سازی در این توابع دیده شوند.



<sup>۱</sup> Object Oriented

<sup>۲</sup> Interface

● سازنده (Constructor)

Signature	Interface::Interface();
Example	Interface interface;

بخش اول: عملیات های افزودن

● اضافه کردن کاربر (addUser)

هر فرد برای ثبت نام در سامانه ابتدا اطلاعات شخصی خود شامل: نام، نام خانوادگی، ایمیل و یک متن به عنوان بیوگرافی خود را مشخص می کند. بعد از فراخوانی این تابع، فرد مورد نظر به کاربران سامانه ما اضافه خواهد شد. دقت کنید که آدرس ایمیل هر کاربر، رشته ای یکتا در سامانه است و دو کاربر با ایمیل یکسان در سامانه وجود ندارند.

Signature	void Interface::addUser(string firstName, string lastName, string emailAddress, string biography);
Example	addUser("hasan", "kachal", " <a href="mailto:h.kachal@ut.ac.ir">h.kachal@ut.ac.ir</a> ", "I am hasan, 18 years old and studying CS at U of Tehran.");

● اضافه کردن شرکت (addCompany)

هر سازمان برای ثبت شدن در این سامانه تابع زیر را فراخوانی می کند و اطلاعاتی شامل: نام، آدرس و توضیحات را به این تابع پاس می دهد. دقت کنید که نام هر سازمان، رشته ای یکتا در سامانه است و دو سازمان با نام یکسان در سامانه وجود ندارند.

Signature	void Interface::addCompany(string name, string address, string description);
Example	addCompany("University of Tehran", "Enghelab Square, Tehran", "The best university ever.");

● اضافه کردن تجربه (addExperience)

هر کاربر می تواند در یک یا چند سازمان مشغول به کار یا تحصیل باشد. با استفاده از تابع زیر، کاربران می توانند سابقه کار خود را در سامانه ثبت کنند. دقت کنید که نام هر شغل در هر شرکتی عنوانی یکتاست و دو شغل با نام یکسان در سامانه وجود ندارند.

Signature	void Interface::addExperience(string userId, string companyId, string title, string startsAt, string endsAt = NOW);
Example	addExperience("h.kachal@ut.ac.ir", "University of Tehran", "Chief TA", "1/7/1396");

دقت کنید که در این پروژه تاریخ هایی وجود دارد که با فرمت متنی به شما داده می شوند و بعدها عملیاتی بر اساس آن ها اتفاق خواهد افتاد. مقدار NOW نیز، یک عبارت تعریف شده در اینترفیس می باشد و به این معنا است که این عملیات هنوز تمام نشده است.

● اضافه کردن فرصت شغلی (addJobRequest)

برای شرکت‌ها این امکان وجود دارد که یک فرصت شغلی تعریف کنند و با در نظر گرفتن حداقل شرایط برای آن، منتظر دریافت درخواست از کاربران واجد شرایط برای آن باشند.

Signature	<code>void addJobRequest(string companyName, string title, map&lt;string, float&gt; conditions);</code>
Example	<code>addJobRequest("University of Tehran", "TA", conditions);</code>

در اینجا هدف استفاده از `map3`، متناظر کردن نام مهارت مورد نیاز به حداقل میزان اعتبار آن می‌باشد. کاربری که مایل است برای یک فرصت شغلی اقدام کند باید حداقل اعتبار لازم را در تمام مهارت‌های آن شغل داشته باشد.

## بخش دوم: عملیات‌های اختصاص

● اختصاص یک مهارت به کاربر (assignSkill)

هر کاربر باید بتواند مهارت‌های خود را در سیستم وارد کند.

Signature	<code>void assignSkill(string userId, string skillName);</code>
Example	<code>assignSkill("h.kachal@ut.ac.ir", "c++");</code>

● تایید مهارت‌های یک کاربر دیگر (endorseSkill)

این امکان برای کاربران وجود دارد که مهارت‌های یک کاربر دیگر را تایید کنند. در حقیقت تعداد افراد تایید کننده مهارت و میزان اعتبار خود آن افراد در آن مهارت، میزان اعتبار کاربر هدف را در آن مهارت خاص، تعیین می‌کند.

Signature	<code>void endorseSkill(string endorserUserId, string skilledUserId, string skillName);</code>
Example	<code>endorseSkill("ali.gholam@ut.ac.ir", "h.kachal@ut.ac.ir", "c++");</code>

● دنبال کردن افراد (follow)

هر کاربر می‌تواند کاربر دیگری را دنبال کند. این کار به تشکیل شبکه‌ای قوی‌تر برای ارتباط با افراد دیگر کمک می‌کند.

Signature	<code>void follow(string followerId, string followingId);</code>
Example	<code>follow("ali.gholam@ut.ac.ir", "h.kachal@ut.ac.ir");</code>

دقت کنید که دنبال کردن عملیاتی دو طرفه است. یعنی اگر علی، حسن را دنبال کند به صورت پیش فرض بعد از اجرای تابع بالا، حسن هم علی را دنبال می‌کند.

● درخواست برای یک فرصت شغلی (applyForJob)

اگر یک کاربر شرایط یک فرصت شغلی را داشته باشد، می‌تواند برای آن درخواست دهد. پس از دادن درخواست، مشخصات این کاربر در لیست متقاضیان این فرصت شغلی اضافه می‌شود.

Signature	<code>void applyForJob(string userId, string companyId, string jobTitle);</code>
Example	<code>applyForJob("h.kachal@ut.ac.ir", "University of Tehran", "TA");</code>

● انتخاب بهترین متقاضی برای شغل (hireBestApplicant)

<sup>3</sup> <http://www.cplusplus.com/reference/map/map/>

یک شرکت در هر لحظه می‌تواند از بین متقاضیان فرصت شغلی که تا آن لحظه اقدام کرده‌اند، بهترین فرد را انتخاب کرده و استخدام نماید. اگر فردی از جانب سازمانی استخدام شود، باید به صورت خودکار این شغل در لیست تجربیات کاری (با تاریخ شروع startsAt و تاریخ پایان NOW) وی افزوده شود.

Signature	void hireBestApplicant(string companyId, string jobTitle, string startsAt);
Example	hireBestApplicant("University of Tehran", "TA", "1/7/1397");

برای مقایسه‌ی افراد متقاضی، باید مجموع اعتبار آنان در مهارت‌های مورد نیاز شغل ثبت شده مقایسه شود و بالاترین انتخاب شود. اعتبار کاربر در یک مهارت از رابطه زیر به دست می‌آید:

$$skillRate = \sum_{user}^{endorsers} (\sqrt{skillEndorsersCount + 1})$$

### بخش سوم: عملیات‌های گزارش‌گیری

دستورات زیر وظیفه‌ی چاپ اطلاعات را در خروجی استاندارد دارند.

#### ● چاپ اطلاعات یک کاربر (printUserProfile)

Signature	void printUserProfile(string userId);
Example	printUserProfile("h.kachal@ut.ac.ir");

با اجرای این دستور کلیه‌ی اطلاعات کاربر شامل: نام و نام خانوادگی، ایمیل، بایوگرافی و اطلاعاتی نظیر تعداد دوستان و سوابق کاری و مهارت‌ها با فرمت زیر باید چاپ شود. توجه کنید که تجربه‌ها باید ابتدا بر اساس تاریخ شروع و پس از آن بر اساس تاریخ پایان مرتب شده باشند.

```
Name: <firstName> <lastName>
Email: <email>
Biography: <biography>
Network: <numOfFollowers> connections
Experiences:
    // for all experience (1 <= i <= n)
    <i>. <startsAt> - <endsAt> <jobTitle> at <companyName>
Skills:
    // for all skills
    <i>. <skillName> - <skillRate>
```

دقت کنید که skillRate ها را با نهایت دو رقم اعشار نشان دهید و اگر کم‌ارزش‌ترین ارقام بعد از اعشار صفر بودند، نیازی به چاپ رقم ۰ در انتهای عدد نیست.

مثال:

Name: Hasan Kachal

Email: h.kachal@ut.ac.ir

Biography: I am hasan, 18 years old and studying CS at U of Tehran.

Network: 5 connections

Experiences:

1. 1/7/1391 - 1/4/1394 student at Helli
2. 1/7/1394 - NOW student at University of Tehran

Skills:

1. c++ - 4.8
2. python - 4.52
3. java - 3.17

● چاپ اطلاعات یک شرکت (printCompanyProfile)

Signature	void printCompanyProfile(string companyName);
Example	printCompanyProfile("University of Tehran");

با اجرای این دستور اطلاعات یک شرکت شامل: نام، آدرس و توضیحات و شغل‌های فعال در شرکت، لیست فرصت‌های شغلی باید با فرمت زیر چاپ شود. (توجه کنید که شغل‌های فعال، شغل‌هایی هستند که یک کاربر به عنوان تجربه کاری در برنامه اضافه کرده است یا پس از انتخاب بهترین متقاضی انتخاب شده‌اند).  
توجه کنید که شغل‌های فعال در شرکت باید ابتدا بر اساس تاریخ شروع و پس از آن بر اساس تاریخ پایان مرتب شده باشند.

Name: <companyName>

Address: <address>

Description: <description>

Jobs:

```
// for all jobs (1 <= i <= n)
<i>. <startsAt> - <endsAt> <jobTitle> by <userFirstName> <userLastName>
```

Requests:

```
// for all requests (1 <= i <= n)
<i>. <jobTitle> - needed skills: <skillName_1>(<skillRate_1>) , ... ,
<skillName_m>(<skillRate_m>)
```

● جستجوی شغل‌های پیشنهادی برای یک کاربر (printSuggestedJobs)

Signature	void printSuggestedJobs(string userId);
Example	printSuggestedJobs("h.kachal@ut.ac.ir");

با اجرای این دستور باید کلیه فرصت‌های شغلی موجود (اخذ نشده) که این کاربر شرایط اقدام برای آن‌ها را دارد با فرمت زیر چاپ شود.

```
// for all requests (1 <= i <= n)
<i>. <jobTitle> in <companyName> - needed skills: <skillName_1>(<skillRate_1>) , ... ,
<skillName_m>(<skillRate_m>)
```

● جستجوی کاربران پیشنهادی برای یک شغل (printSuggestedUsers)

Signature	void printSuggestedUsers(string companyName, string jobTitle);
Example	printSuggestedUsers("University of Tehran", "TA");

با اجرای این دستور، باید کلیه‌ی کاربرانی که شرایط کافی برای شغل مربوطه را دارند، لیست شوند.

```
// for all qualified users (1 <= i <= n)
<i>.
<user_i_profile>
```

● جستجوی سطح‌های دوستی (امتیازی) (printNetwork)

در این بخش می‌خواهیم به کاربران کمک کنیم تا با دوستان دوستانشان آشنا شوند. به این صورت که با مشخص کردن کاربر و درجه‌ی واسطه‌ی آشنایشان، کاربرانی پیشنهاد می‌شوند که با level واحد واسطه، کاربر ما را می‌شناسند (دنبال می‌کنند). دقت داشته باشید که level هر عدد صحیح مثبتی می‌تواند باشد.

Signature	void printNetwork(string userId, int level);
Example	printNetwork("h.kachal@ut.ac.ir", 3);

فرمت نمایش کاربران آشنا نیز به صورت زیر خواهد بود.

```
// for all qualified users (1 <= i <= n)
<i>.
<user_i_profile>
```

## نکات پایانی

- دقت کنید که این کلاس و متدهای این کلاس تنها رابط می‌باشند و برای سهولت شما و درگیر نشدن با خواندن ورودی است و منطق برنامه نباید در آن‌ها پیاده‌سازی شود. در صورت عدم رعایت این نکته، از شما نمره کسر خواهد شد.
- پروژه‌ی شما باید حتماً Makefile داشته باشد، و کلاس‌ها در فایل‌های مجزا پیاده‌سازی شوند. در غیر این صورت نمره بخش اجرا را از دست خواهید داد.
- توجه داشته باشید که برای هر کاربر userId و emailAddress و برای هر شرکت companyName و companyId یکسان هستند.
- برای راحتی کار با تاریخ‌ها می‌توانید کلاسی با عنوان Date در نظر بگیرید که تمامی عملیات مربوط به تاریخ‌ها را هندل می‌کند.

- دقت کنید که برنامه‌ی شما در صورت وجود خطا در ورودی‌های توابع (مانند عدم وجود کاربر اعلام شده یا عدم وجود شرکت یا فرصت شغلی) دچار مشکل و خطا شود. در صورتی که اطلاعات وجود نداشت، لازم نیست پیامی برای بروز خطا چاپ کنید و تنها دقت کنید که کاری نباید انجام شود. به طور کلی در صورت صحت ورودی‌ها عملیات مربوطه انجام شده و در غیر این صورت، انجام نمی‌شود.
- شما باید کلاس Interface را مطابق با فایل hpp. ای که در اختیارتان قرار می‌گیرد پیاده‌سازی کنید. می‌توانید به این کلاس تابع یا فیلدهای دیگری که می‌خواهید اضافه کنید.

## نحوه‌ی تحویل

فایل‌های برنامه‌ی خود را با نام A4-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که در آن SID شماره دانشجویی شماست.

لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.

- برنامه‌ی شما باید در سیستم عامل لینوکس و با مترجم ++g با استاندارد ++c98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.

- به فرمت و نام فایل‌های خود دقت کنید.

- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.

- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.