



برنامه نویسی پیشرفته: تمرین سری پنجم

تاریخ تحویل: تا دوشنبه ۱/۲۸ ساعت ۲۳:۵۵ (حداکثر تا سه روز بعد از این تاریخ قادر به ارسال تمرینات هستید و به ازای هر روز ۲۵٪ نمره کسر می‌گردد)

توصیه می‌شود درباره تمرینات با همکلاسی‌های خود به صورت گروهی به بحث و تبادل نظر بپردازید اما این به معنای تقلب، کپی کردن و... نمی‌باشد و تمام تمرینات باید توسط شما حل و پیاده سازی شود. تمام تمرینات در یک پوشه با شماره دانشجویی و شماره تمرین و با ساختار به صورت زیر قرار گیرد، در غیر این صورت به عنوان عدم دریافت تلقی می‌شود.

HW_5_9531901

***توجه:** تمام قسمت‌های اپلیکشن در پکیج مانند ir.aut.ceit.app قرار گیرد و باید از پکیج‌های مناسب دیگر (در صورت لزوم) استفاده نمایید و قرار دادن غیرمنطقی تمام کلاس در یک پکیج باعث کسر نمره خواهد شد.

****توجه:** تمام پروژه در قالب یک فایل Zip ارسال شود.

(۱) وضعیت آب و هوا (۱۰۰ نمره)

در این تمرین می‌خواهیم یک اپلیکشن آب و هوایی بسیار ساده ایجاد و برای گرفتن وضعیت دقیق آب و هوا از اطلاعات سایت openweathermap.com استفاده کنیم. نحوه استفاده از اطلاعات این وبسایت بسیار ساده بوده و تنها کافی است مانند نمونه داده شده در کد Weather.java عمل کنید. ابتدا باید در این سایت ثبت نام کرده و وارد آن شوید تا یک کلید خصوصی (key) مخصوص شما ایجاد کند (دقت کنید این کلید منحصر به فرد بوده و نمی‌توانید از کلید دیگران استفاده کنید)

Key

a5263d7b4854e652435cb4876561ac8b

شکل ۱- یک کلید خصوصی نمونه.

حال شما میتوانید درخواستهای مختلفی را به این سایت ارسال و سایت اطلاعات مربوطه را به شما بازگرداند. به عنوان مثال فرض کنید اطلاعات آب و هوایی شهر لندن را بخواهید، این اطلاعات را به راحتی از طریق درخواست زیر دریافت کنید.

<http://samples.openweathermap.org/data/2.5/weather?q=London&appid=a5263d7b4854e652435cb4876561ac8b>

در این درخواست q=London اسم شهر مورد نظر و appid=xxx کلید خصوصی شما است. اگر لینک داده شده را در browser باز کنید اطلاعاتی به شما نمایش داده می شود (برای مشاهده بهتر اطلاعات، جواب حاصل از API ی بالا را در سایت <http://jsonviewer.stack.hu> کپی و پیست کنید) و آنگاه با استفاده از کدی که در فایل Weather.java وجود دارد میتوانید این اطلاعات را دریافت و نمایش دهید.

```
static private String ParseResult(String json) throws
JSONException {

    JSONObject jsonObject = new JSONObject(json);

    //"coord"
    JSONObject JSONObjectCoord =
jsonObject.getJSONObject("coord");
    Double result_lon = JSONObjectCoord.getDouble("lon");
    Double result_lat = JSONObjectCoord.getDouble("lat");

    //"sys"
    JSONObject JSONObjectSys =
jsonObject.getJSONObject("sys");
    String result_country =
JSONObjectSys.getString("country");
    int result_sunrise = JSONObjectSys.getInt("sunrise");
    int result_sunset = JSONObjectSys.getInt("sunset");

    ...// See rest of the code in Weather.java
```

همانطور که در Weather.java مشاهده می کنید برای دسترسی به اطلاعات باید ابتدا در خواستی از طریق HttpURLConnection (در پکیج java.net) ایجاد کنید و سپس از کلاسهای با نام JSONObject, JSONArray استفاده کرده تا به هریک از فیلدهای این اطلاعات دسترسی پیدا کنید. برای استفاده از کلاسهای JSONObject, JSONArray باید فایل کتابخانه gson را در قالب یک فایل jar به پروژه خود اضافه کنید (فایل jar را از [اینجا](#) دانلود کنید).

حال از شما می‌خواهیم یک اپلیکشن آب و هوای ساده ایجاد کنید که قابلیت زیر را داشته باشد:

- اسم یک شهر را با استفاده از Scanner گرفته و اطلاعات مربوط به آب هوای فعلی (Current) را در خروجی به صورت زیر چاپ کند (اطلاعات به صورت جدول زیر نمایش داده شود).

City	Humidity	Description	Wind-Info	Coordination
Your information goes here				

- اسم یک شهر را با استفاده از Scanner گرفته پیش‌بینی (Forecast) آب هوای آن شهر را با استفاده از API های زیر به صورت یک جدول نمایش دهد (با دریافت یک عدد از کاربر، اطلاعات ۵ الی ۱۶ روز اخیر نمایش داده شود)

[http://samples.openweathermap.org/data/2.5/forecast?q=London
&appid=b1b15e88fa797225412429c1c50c122a1](http://samples.openweathermap.org/data/2.5/forecast?q=London&appid=b1b15e88fa797225412429c1c50c122a1)

[https://samples.openweathermap.org/data/2.5/forecast/daily?lat=35&lon=139&cnt=10
&appid=b1b15e88fa797225412429c1c50c122a1](https://samples.openweathermap.org/data/2.5/forecast/daily?lat=35&lon=139&cnt=10&appid=b1b15e88fa797225412429c1c50c122a1)

- اسم یک شهر را گرفته اطلاعات مربوط به آب و هوای گذشته (History) آن نمایش داده شود.
<http://samples.openweathermap.org/data/2.5/history/city?q=London&appid=b1b15e88fa797225412429c1c50c122a1>

- در تمام موارد بالا علاوه بر نام شهر باید بتوان اطلاعات را با استفاده از مختصات جغرافیایی lat, lon و یا idی شهر موردنظر id وارد کرده و اطلاعات خواسته شده را نمایش دهد. پس واضح است API های شما اندکی تغییر کرده و این بار به جای استفاده از q=London باید از id=xxx یا lat=35,lon=139 استفاده نمایید (برای اطلاعات بیشتر درباره نحوه استفاده از API به اینجا مراجعه نمایید).

- یک کلاس OutputFileWriter بنویسید که اطلاعات بدست آمده در هر وب سرویس را به صورت جدولی ذخیره و نام فایل را بر اساس زمان حال (System.currentTimeMillis) ذخیره کنید و اطلاعات کلاسهای متفاوت را در فایل‌های متفاوت نگهداری کنید. به عبارت دیگر در صورتی که کاربر در زمان t1 و t2 دوبار اطلاعات فعلی (Current) آب و هوا را مشاهده کند باید در فولدری با نام Current دو فایل با نامهای t1.txt و t2.txt ایجاد و اطلاعات در آن ذخیره باشد. (برای اسامی فایل t1 و t2 از new Data(System.currentTimeMillis).toString استفاده کنید)

- مختصات lat, lon یک شهر را گرفته و وضعیت فعلی پاکی اب و هوا UV را نمایش دهد.
<http://api.openweathermap.io/air/1.0/uvi/current?lat=55&lon=37&appid=b1b15e88fa797225412429c1c50c122a1>

UV-Index	color	Risk of Harm	Recommended Protection
Your info goes here			

اطلاعات بیشتر درباره UV را در [اینجا](#) بخوانید (از API قدیمی که لینک آن داده شده است استفاده

کنید).

برخی نکات پیاده سازی:

- در این تمرین باید سعی کنید تا تمام مفاهیم OOP را که تاکنون یاد گرفته اید بکار ببرید و صرف نمایش اطلاعات در تابع main با استفاده از سوییچ کیس متوالی نمره ای تعلق نخواهد گرفت.
- یک کلاس با نام ApiWeather بنویسید که تمام مسایل مربوط به URLها را در آن هندل کنید و سایر کلاسها باید برای دسترسی به API ها از این کلاس استفاده کنند.
- مباحث مربوط به نامگذاری، پکیج بندی و عدم وجود duplicate code حتما رعایت شود.
- اطلاعات و متدهای لازم مربوط به پیش‌بینی هوا در کلاس ForecastWeather، وضعیت فعلی آب و هوا در کلاس CurrentWeather، وضعیت UV در کلاس UVWeather، تاریخچه آب و هوا در HistoryWeather قرار گیرد که همگی از یک کلاس با نام BaseWeather ارث بری می‌کنند و کدهای مشترک آنها تنها در کلاس BaseWeather قرار می‌گیرد.
- هرکدام از کلاسها باید وظایف مربوط به خود را داشته باشند و وظایف مشترک باید به نحوه درست هندل شوند.
- مسایل مربوط به خواندن و نوشتن فقط باید در کلاس InputFileReader و OutputFileWriter انجام شود و کلاسهای دیگر حق استفاده از InputStreamReader، OutputStreamWriter یا BufferedOutputStream و کلاسهای مشابه را ندارند.
- نوشتن جاوا داک ضروری است.
- کلاس Scanner به هیچ عنوان نباید در داخل کلاسهای اصلی استفاده شود و فقط باید یک رشته (اسم شهر یا آیدی) از کاربر دریافت و از طریق سازنده یا یک تابع به کلاس مربوطه ارسال کند.
- اطلاعات مختلف باید به صورت یک منو به کاربر نمایش داده شود و کاربر با انتخاب شماره آن گزینه، اطلاعات شهر (آیدی یا مختصات جغرافیایی) را وارد کرده و سپس اطلاعات مرتبط به وی نمایش داده شود و تا هنگامی که کاربر گزینه خروج را نزده است بتواند با برنامه کار کند.