



# تمرین سری اول

دکتر سعید پارسا

الناز رضایی

اسفند ۱۴۰۰

## سوال ۱ :

انتلر را نصب کنید. مراحل نصب را به صورت خلاصه توضیح داده و از خروجی موفق دستورات زیر grun ، antlr۴ در ترمینال اسکرین شات تهیه کنید.

## پاسخ ۱:

روش plugin :

در ابتدا از منوی بالای صفحه Pycharm، گزینه setting را انتخاب کرده و از بخش setting، روی گزینه plugins کلیک می‌کنیم. سپس در قسمت search این بخش، ANTLR را سرچ کرده و ANTLR v۴ را نصب می‌کنیم. سپس از بخش Project همین قسمت، setting بخش Python-In-terpreter، روی + کلیک کرده و antlr۴-python۳-runtime را نصب می‌کنیم.

روش نصب روی سرور:

۱. ابتدا JDK را بر روی سیستم خود نصب می‌کنیم.
۲. antlr.bat grun.bat و antlr-۸.۴-complete.jar را دانلود کرده و در مسیر C:\Javlib قرار می‌دهیم.
۳. ANTLR را در variables system وارد کرده و داخل کلاسی با نام classpath نگهداری می‌کنیم.
۴. سپس در cmd دستورهای خواسته شده را اجرا می‌کنیم که نتیجه خروجی آن به شرح زیر است:

```
C:\Users\user>antlr4
C:\Users\user>java -jar "C:\Javlib\antlr-4.8-complete.jar"
ANTLR Parser Generator Version 4.8
-o _____ specify output directory where all output is generated
-lib _____ specify location of grammars, tokens files
-atn _____ generate rule augmented transition network diagrams
-encoding _____ specify grammar file encoding; e.g., euc-jp
-message-format _____ specify output style for messages in antlr, gnu, vs2005
-long-messages _____ show exception details when available for errors and warnings
-listener _____ generate parse tree listener (default)
-no-listener _____ don't generate parse tree listener
-visitor _____ generate parse tree visitor
-no-visitor _____ don't generate parse tree visitor (default)
-package _____ specify a package/namespace for the generated code
-depend _____ generate file dependencies
-D<option>=value _____ set/override a grammar-level option
-llerror _____ treat warnings as errors
-XdbgST _____ launch StringTemplate visualizer on generated code
-XdbgSTWait _____ wait for STVir to close before continuing
-Xforce-atn _____ use the ATN simulator for all predictions
-Xlog _____ dump lots of logging info to antlr-timestamp.log
-Xexact-output-dir _____ all output goes into -o dir regardless of paths/package

C:\Users\user>grun

C:\Users\user>java org.antlr.v4.gui.TestRig
java org.antlr.v4.gui.TestRig GrammarName startRuleName
[-tokens] [-tree] [-gui] [-ps file.ps] [-encoding encodingname]
[-trace] [-diagnostics] [-SLL]
[input-filename(s)]
Use startRuleName='tokens' if GrammarName is a lexer grammar.
Omitting input-filename makes rig read from stdin.

C:\Users\user>
```

## سوال ۲ :

ا بعد از نصب کردن پالگین انتلر ، شکل درخت را برای قطعه کد جاوای زیر را در گزارش تمرین نمایش بدهید.

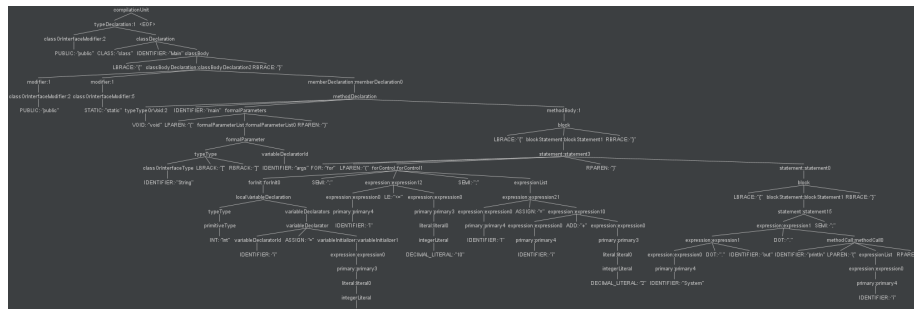
```
for (int i = 0; i <= 10; i = i + 2) {  
    System.out.println(i);  
}
```

## پاسخ ۲:

در ابتدای هر پروژه از بخش setting، بخش project گزینه interpreter python علامت + بالای صفحه را زده و antlr۴ را سرچ کرده و antlr۴-python۳-runtime را دانلود می‌نماییم.

در ابتدا روی فایل‌های g۴. کلیک راست کرده و گزینه ANTLR configure را انتخاب کرده و در بخش Python۳ Language، سپس دوباره روی فایل‌ها کلیک راست کرده و گزینه recognizer ANTLR generate را انتخاب می‌کنیم.

در فایل JavaParserLabeled روی compilationUnit کلیک راست کرده و گزینه Rule test compilationUnit را انتخاب کرده و در انتهای صفحه، در بخش preview antl کد خود را وارد کرده و در سمت راست آن بخش tree parse می‌توان درخت آن را مشاهده کرد.



### سوال ۳ :

با استفاده از ابزار انتلر و زبان پایتون یک برنامه بنویسید که ورودی آن مسیر پروژه جاوا باشد و خروجی آن اسم تابع های پروژه جاوا باشد. (تذکر : دقت کنید که تنها با استفاده از ابزار انتلر خروجی بالل مورد پذیرش خواهد بود و در غیر این صورت هیچ نمره ای به شما تعلق نخواهد گرفت.)

### پاسخ ۳:

در بخش metric.py، تابع هایی با نام enterMethodDeclaration و enterPackageDecla-ration تعریف می کنیم تا به ترتیب نام method ها و package مربوط به فایل های java که در پوشه testProject قرار گرفته اند را تشخیص دهد.

سپس در بخش main برنامه دو حلقه for تو در تو برای walk کردن فایل های جاوا قرار می دهیم و در انتها خروجی تابع های بالا را در یک دیکشنری به نام packagesMethods می ریزیم و در خروجی چاپ می کنیم.

همچنین باید دقت داشت تا مسیر پروژه هم در for باید تعیین کرد تا اگر چند فایل ورودی داشتیم، همه را walk کند.

```
Compiler result :
[{'package_name': 'a.a.a', 'methods': ['m1', 'm2', 'm3', 'm4']}, {'package_name': 'a.a.a', 'methods': ['m1', 'm2', 'm3', 'm4']}, {'package_name': 'b.b.b', 'methods': ['m1', 'm2']}
Process finished with exit code 0
```

## سوال ۴ :

استفاده از انتلر و زبان پایتون یک برنامه بنویسید که ورودی آن مسیر پروژه جاوا باشد و خروجی آن تعداد های attribute هر کلاس باشد. یک آرایه از زوج مرتب ها.

## پاسخ ۴:

برای این سوال هم تا حدودی مشابه سوال قبل عمل کرده اما با یک تفاوت که یک دیکشنری با نام attributes تعریف می کنیم و برای هر کلاس مقدار اولیه آن را صفر قرار می دهیم. سپس با استفاده از enterFieldDeclaration method به هر کلاس که رسیدیم، تعداد attribute هایش را محاسبه می کنیم. بخش main این تمرین هم مشابه تمرین قبل می باشد.

```
compiler result :  
[{'package_name': 'a.a.a', 'attributes': {'A': 2}}, {'package_name': 'b.b.b', 'attributes': {'B': 2}}, {'package_name': 'c.c.c', 'attributes': {'C': 2}}]
```

## سوال ۵ :

یک FSA برای تشخیص یک لینک صحیح ارائه کنید. سپس در یک فایل گرامر انتلر آن را تست کنید های (url) زیر را در Antlr Preview تست کنید. اسکرین شات هایی که گرامر لینک های زیر را پذیرفته است ارائه کنید.

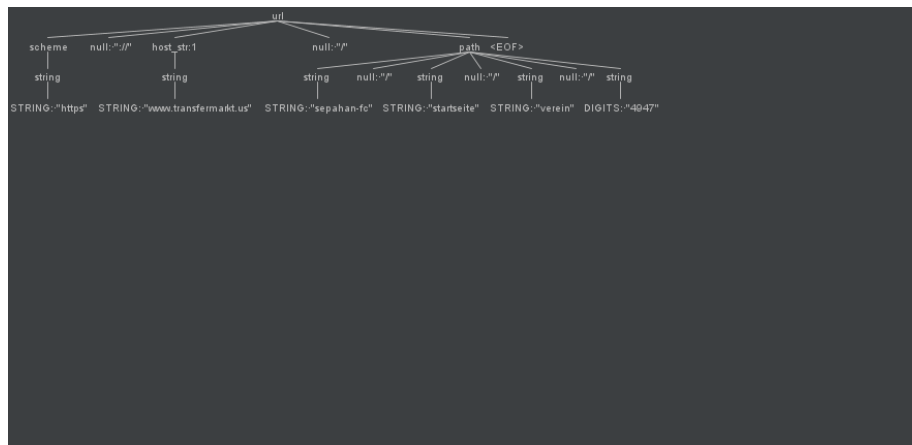
<https://www.transfermarkt.us/sepahan-fc/startseite/verein/4947>

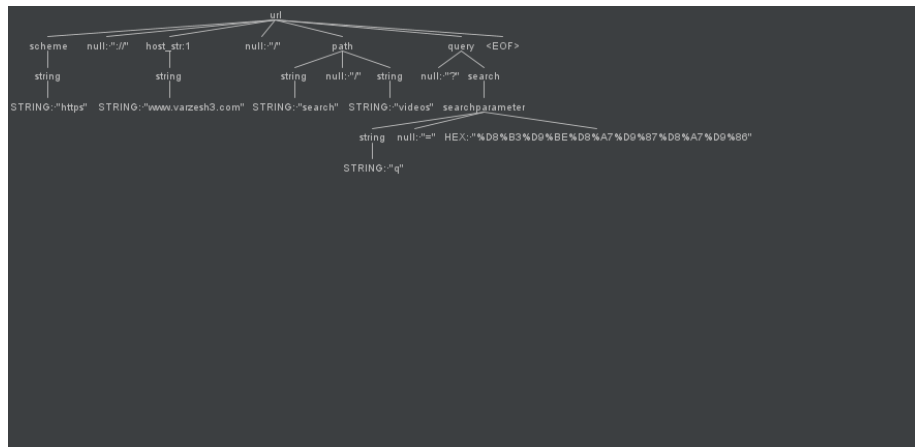
<https://www.varzesh3.com/search/videos?q=%D8%B3%D9>

<https://www.varzesh3.com/search/videos?q=%D8%B3%D9%BE%D8%A7%D9%87%D8%A7%D9%86>

## پاسخ ۵ :

گرامر مربوطه را در فایل Url.g۴ نوشته و روی همین فایل کلیک راست کرده و rule test را می زنیم و در قسمت url preview، antlr ها را وارد کرده و در سمت راستش خروجی را می توانیم مشاهده کنیم.





## سوال ۶ :

عملیات سوال ۵ را برای رشته Connection String انجام دهید و از رشته های صحیح اسکرین شات تهیه کنید.

## پاسخ ۶:

گرامر مربوطه را با بهره گیری از لینک قرار داده شده در سوال در فایل connectStrings نوشته و مانند بخش قبل url ها را امتحان می کنیم.

