کامپایلر

فاز اول پروژه

فاطمه کمانی و سیده صدیقه مکی

۹۷۰۱۲۲۶۸۱۰۰۸ - ۹۷۰۱۲۲۶۸۰۰۳۳

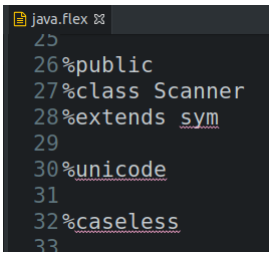
برای انجام فاز اول پروژه ابتدا باید اسکنر تولید کنیم پس jflex و cup را نصب میکنیم .

پس از نصب jflex در پوشه examples فایلی به نام java.flex وجود دارد که اسکنر جاوا است در این پروژه از این فایل استفاده شده و تغییرات گفته شده در صورت پروژه بر این فایل اعمال شده .

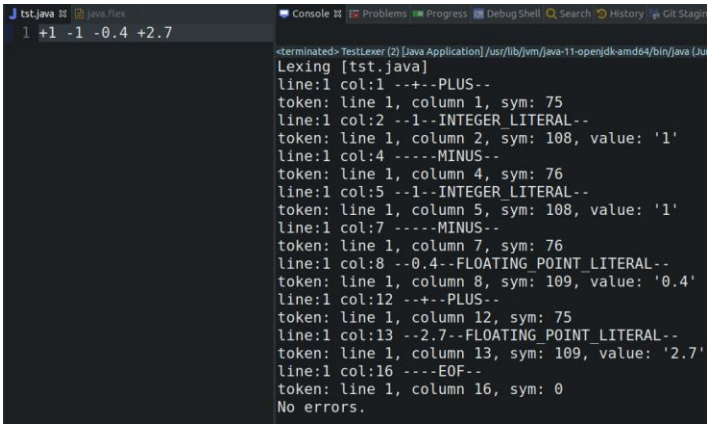
در صورت پروژه به مواردی که اسکنر باید تشخیص دهد اشاره شده که در ادامه آنها را بررسی می‌کنیم

non case sensitive بودن:

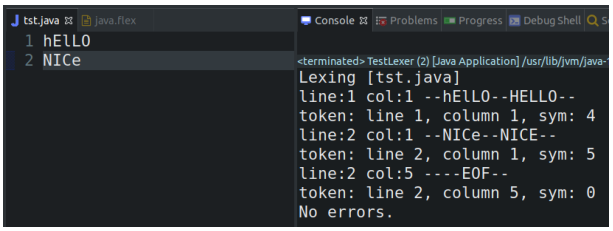
با استفاده از آپشن caseless این خواسته بر آورده شد.



۱- اعداد مثبت، منفی و اعشاری:



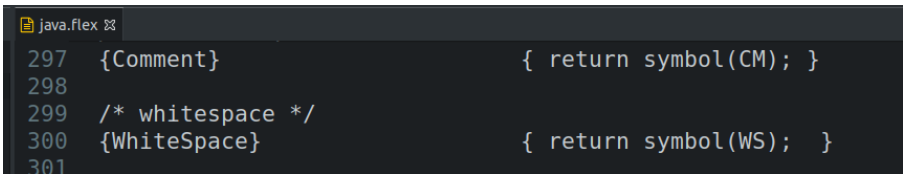
۲- کلمات nice و hello

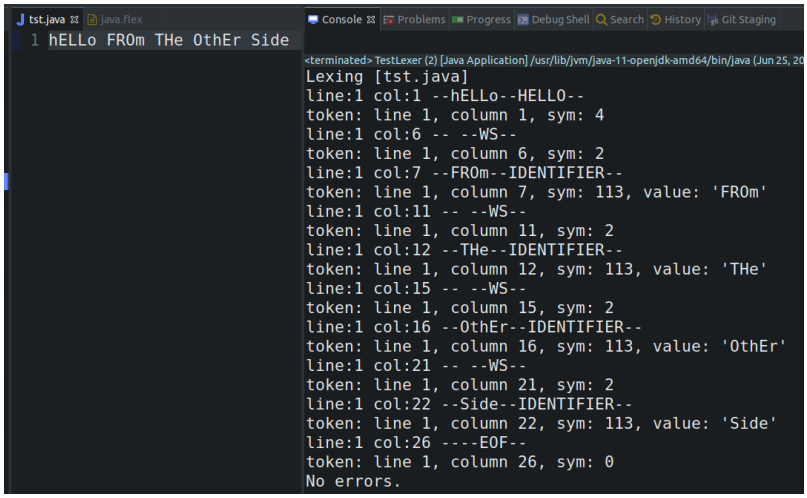


۳- فضای خالی یا White Space (توکن WS):

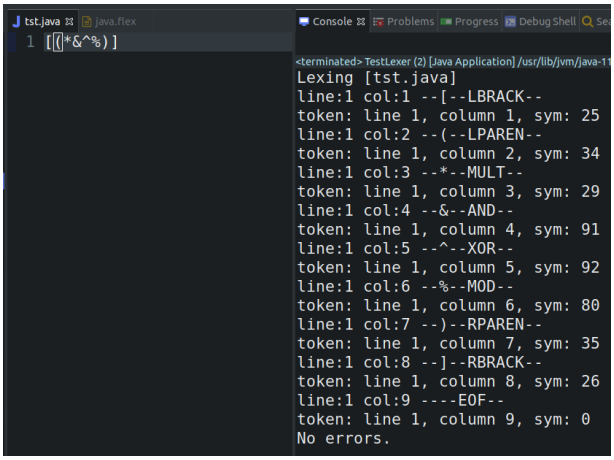
در فایل اصلی java.flex توکن های مربوط به comment و whitespace ایگنور میشدند و شناسایی نمیشدند . اما در صورت پروژه از ما خواسته شده بود که آنها را شناسایی کنیم .

پس برای آنها نیز return میگذاریم تا شناسایی شوند:

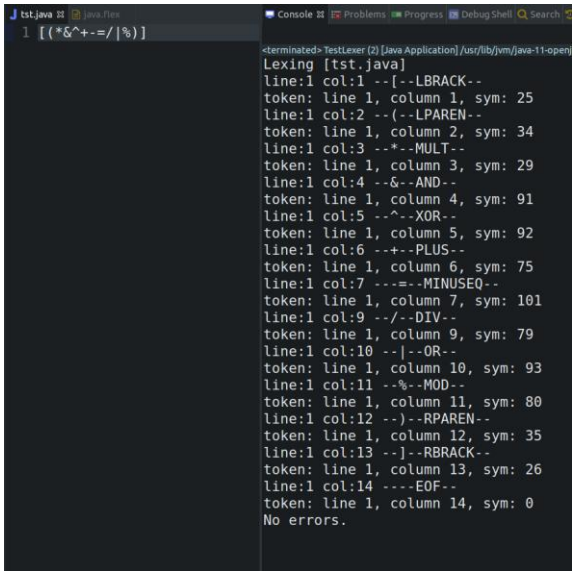




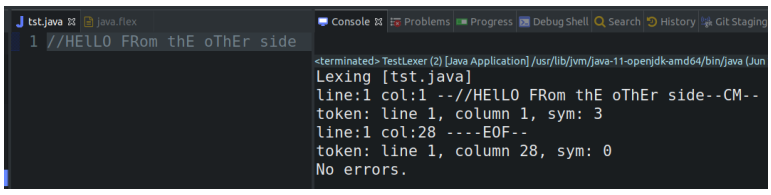
۴- علامت های پشتیبانی شده:



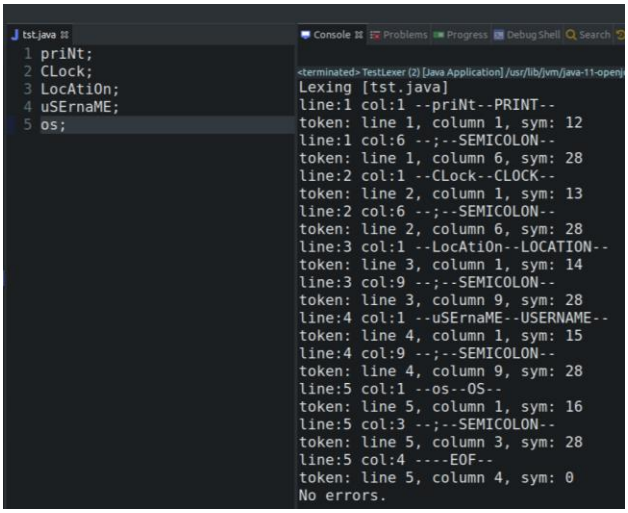
۵- عملگر ها:



۶- کامنت (توکن CM) :



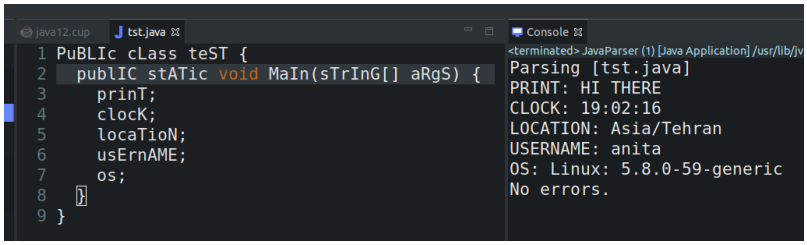
۷- وجود ۵ کلمه کلیدی:



```
tst.java
1 priNt;
2 CLock;
3 LocAtiOn;
4 uSErnaME;
5 os;

Console  Problems  Progress  Debug Shell  Search
<terminated> TestLexer (2) [Java Application] /usr/lib/jvm/java-11-openj
Lexing [tst.java]
line:1 col:1 --priNt--PRINT--
token: line 1, column 1, sym: 12
line:1 col:6 --;--SEMICOLON--
token: line 1, column 6, sym: 28
line:2 col:1 --CLock--CLOCK--
token: line 2, column 1, sym: 13
line:2 col:6 --;--SEMICOLON--
token: line 2, column 6, sym: 28
line:3 col:1 --LocAtiOn--LOCATION--
token: line 3, column 1, sym: 14
line:3 col:9 --;--SEMICOLON--
token: line 3, column 9, sym: 28
line:4 col:1 --uSErnaME--USERNAME--
token: line 4, column 1, sym: 15
line:4 col:9 --;--SEMICOLON--
token: line 4, column 9, sym: 28
line:5 col:1 --os--OS--
token: line 5, column 1, sym: 16
line:5 col:3 --;--SEMICOLON--
token: line 5, column 3, sym: 28
line:5 col:4 ----EOF--
token: line 5, column 4, sym: 0
No errors.
```

کامپایل کلمات کلیدی:

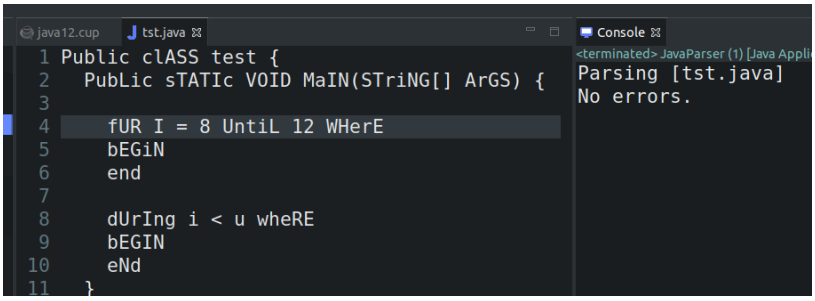


```
java12.cup  tst.java
1 PuBLIc cLass teST {
2   publIC stATic void MaIn(sTrInG[] aRgS) {
3     prinT;
4     clocK;
5     locaTioN;
6     usErnAME;
7     os;
8   }
9 }

Console
<terminated> JavaParser (1) [Java Application] /usr/lib/jv
Parsing [tst.java]
PRINT: HI THERE
CLOCK: 19:02:16
LOCATION: Asia/Tehran
USERNAME: anita
OS: Linux: 5.8.0-59-generic
No errors.
```

۸- وجود Identifier : به مراتب در عکس های بالا نشان داده شد.

بخش for و while: در این بخش دو گرامر fur و during اضافه شد.



```
java12.cup  tst.java
1  Public clASS test {
2    PubLic sTATIc VOID MaIN(STriNG[] ArGS) {
3
4      fUR I = 8 UntiL 12 WHerE
5      bEGiN
6      end
7
8      dUrIng i < u wheRE
9      bEGIN
10     eNd
11   }

Console
<terminated> JavaParser (1) [Java Appli
Parsing [tst.java]
No errors.
```

مشاهده می شود که کامپایل بدون خطا انجام شد.

تصویر از محیط JFlex :

```
java.flex
184
185  "hello"                    { return symbol(HELLO); }
186  "nice"                     { return symbol(NICE); }
187  "where"                    { return symbol(WHERE); }
188  "print"                    { return symbol(PRINT); }
189  "clock"                    { return symbol(CLOCK); }
190  "location"                 { return symbol(LOCATION); }
191  "username"                 { return symbol(USERNAME); }
192  "os"                       { return symbol(OS); }
193
194  "during"                   { return symbol(DURING); }
195  "fur"                      { return symbol(FUR); }
196  "where"                    { return symbol(WHERE); }
197  "begin"                    { return symbol(BEGIN); }
198  "end"                      { return symbol(END); }
199  "until"                    { return symbol(UNTIL); }
```

پس همانطور که مشاهده شد اسکنر قابلیت تشخیص تمامی موارد گفته شده در صورت پروژه را دارد.

تصویر از محیط java۱۲.cup:

در پوشه examples فایلی به نام java۱۲.cup نیز وجود دارد که گرامر های زبان جاوا هست . ابتدا ترمینال هایی که به java.flex اضافه کردیم را به آن اضافه میکنیم سپس برای گرامر های جدیدی که می‌خواهیم بنویسیم نان ترمینال های مورد نظر را اضافه میکنیم .(با استفاده از cup از java۱۲.cup در مسیر گفته شده sym.java و parser.java ایجاد میشود و این فایل ها را نیز باید به پروژه خود اضافه کنیم و برای اجرای فایل ها نیاز داریم که از jar file های موجود در فایل cup استفاده کنیم و آنها را نیز پروژه خود اضافه کنیم.)

تعریف ترمینال ها:

```
java12.cup
79////////////////
80terminal WS; //white space token
81terminal CM; //comment token
82terminal HELLO; //hello token
83terminal NICE; //nice token
84///////////////
85terminal FUR;//FUR token used for the requested for function
86terminal DURING;//DURING token used for the requested while function
87terminal WHERE; //where token
88terminal BEGIN; //BEGIN token
89terminal END; //END token
90terminal UNTIL;//UNTIL token
91//
92//
93terminal PRINT;//print_stmt// a keyword to print "HI THERE"
94terminal CLOCK;//clock_stmt// a keyword to print clock
95terminal LOCATION;//location_stmt// a keyword to print location
96terminal USERNAME;//username_stmt// a keyword to print username
97terminal OS;//os_stmt// a keyword to print os name and virsion
98///////////////
99
```

تعریف نان ترمینال ها:

```
java12.cup
100
101
102/////////////////
103non terminal my_statement;// a statemnt used for declaring project things
104non terminal keyword_statement;// a statement used for keywords
105non terminal print_stmt; // a statement used for PRINT keyword
106non terminal clock_stmt;  // a statement used for CLOCK keyword
107non terminal location_stmt;  // a statement used for LOCATION keyword
108non terminal username_stmt;  // a statement used for USERNAME keyword
109non terminal os_stmt;  // a statement used for OS keyword
110//
111//
112non terminal fur_stmt; // a statement used for fur token
113non terminal during_stmt; // a statement used for during token
114///////////////////
115
```

اضافه کردن نان ترمینال my_statement به گرامر statement :

```
java12.cup
617 statement ::=   statement_without_trailing_substatement
618     |   labeled_statement
619     |   if_then_statement
620     |   if_then_else_statement
621     |   while_statement
622     |   for_statement
623     |   my_statement
624     ;
```

گرامر ها my_statement و keyword_statement :

```
java12.cup
972
973
974/////////
975my_statement ::=
976         keyword_statement  SEMICOLON
977     |   during_stmt
978     |   fur_stmt
979     ;
980
981//////////////////
982/////////
983keyword_statement ::=
984         print_stmt
985     |   clock_stmt
986     |   location_stmt
987     |   username_stmt
988     |   os_stmt
989     ;
```

گرامر های keyword ها و گرامر fur و during : برای قسمت دوم پروژه که باید گرامر های جدید تعریف میشد ، چون گرامر های for و while از قبل وجود دارد، به جای for و while از کلمات fur و during استفاده شد و گرامر های گفته شده نوشته شد.