

پروژه درس نظریه زبان ها و ماشین ها

نیم سال دوم ۹۸-۹۹

گرامرنویسی با انتلر

-
- موعده تحویل پروژه ساعت ۸:۰۰ صبح روز **شنبه ۳ خرداد ۱۳۹۹** بوده و تحویل حضوری آن متعاقبا اعلام می گردد.
 - ارسال پروژه تنها از طریق لینک در اپ باکس مشخص شده قابل قبول است.
 - انجام پروژه به صورت **انفرادی** می باشد.
 - در صورت مشاهده هرگونه تخلف، نمره پروژه فرد یا افراد متخلف -۰- خواهد بود.

شرح پروژه:

در این پروژه هدف آشنایی با نرم افزار Antlr و پیاده سازی ساختار کلی یک زبان ساده و رسم درخت برنامه های نوشته شده برای این زبان است. ساختار این زبان در ادامه توضیح داده شده است.

ساختار کلی زبان:

۱. در ابتدای برنامه تعدادی کتابخانه یا اعضای آن ها استفاده می شود.
۲. هر برنامه دارای یک یا چند class است.
۳. هر class می تواند شامل تعریف چند متغیر و تابع باشد.
۴. تعریف حوزه در این زبان { و } است.
۵. بعد از دستورات می تواند علامت ; (semicolon) قرار بگیرد.
۶. قوانین نام گذاری مشابه زبان جاوا است با این تفاوت که هر اسم حداقل ۲ کاراکتر دارد.
۷. عبارات **Bold** شده کلمات کلیدی (با حروف کوچک) هستند.

- استفاده از کتابخانه و یا اجزای آن ها (توابع یا متغیرهای گلوبال):

```
lib1 = require <library-1>;  
func1 = from <library-2> require <function1>;  
...
```

- تعریف class:

بخش های extends و implements می توانند در معرفی یک کلاس باشند یا نباشند.

```
class <class-name> extends <parent-class-name> implements <interface-name-1>,  
<interface-name2>, ... {  
  
    <statement-1>  
  
    <statement-2>  
  
    ...  
  
}
```

- تعریف متغیر (و انتساب):

```
let/const <variable-type> <variable-name>, <variable-name> (= <value>), ...;  
let/const <variable-type> <variable-name> = <variable-name> = ... = <value>;  
let/const <variable-type>[] <variable-name>, <variable-name> (= <value>), ...;  
let/const <variable-type>[] <variable-name> = <variable-name> = ... = <value>;
```

این زبان از نوع داده صحیح (int)، اعشاری (float)، ثابت‌ها (constants) شامل: true، false و iota و مقدار null پشتیبانی می‌کند.

- تعریف تابع:

```
<func-name> = (<func-args>) => {  
    <function-Code>  
    return <value> | <variable-name> (در صورت نیاز)  
}
```

<function-code> می‌تواند تعریف متغیر، if، for، while و switch case باشد.

- دستور for به دو صورت بازه‌ای و پیمایشی تعریف می‌شود:

○ بازه‌ای:

```
for (<variable-name> in <variable-range> step <int>) {  
    <function-code>  
}
```

که مقدار <variable-range> به صورت زیر مشخص می‌شود:

<int>:<int> (به عنوان مثال 1:5)

○ پیمایشی:

```
for (auto <variable-name> in <iterator-name>) {  
    <function-code>  
}
```

- دستورات if، while و switch case مشابه زبان جاوا تعریف می‌شود.

○ بخش شرط در دستورات فوق از اهمیت بالایی برخوردار بوده و لازم است اولویت عملگرها در

آن رعایت شود. (اولویت عملگرها در ادامه توضیح داده می‌شود).

- دستور انتساب (برای متغیر از قبل تعریف شده):

```
<variable-Name> = <value>;  
<variable-Name> = <variable-name>;  
<variable-Name> = <variable-name>[<variable-range>;]
```

در صورتی که مقدار سمت راست انتساب عدد باشد می‌تواند به صورت نماد علمی وارد شود. (برای مثال:
(1.037e-21

- کامنت‌گذاری:

به دلخواه کاراکتری را برای کامنت‌گذاری تعریف کنید. خطوطی که با این کاراکتر شروع شوند یا در بین دو تا از این کاراکترها باشند، نباید توسط برنامه‌ی شما بررسی شده و در درخت ترسیم شوند.

- عملگرها:

لازم است برنامه شما عملگرهای زیر را تشخیص دهد:

عملگرهای حسابی:

+, -, *, **, /, //, %

عملگرهای انتساب:

=, +=, -=, *=, /=, %=, **=, //=

عملگرهای بیتی:

>>, <<, ~, ^, |, &

عملگرهای منطقی:

and, or, not

اولویت عملگرها به ترتیب:

1. ()
2. **
3. ~
4. - + (مثبت و منفی)
5. * / // %
6. - + (جمع و تفریق)
7. ### (عملگر یکانی یا تک‌عملوند)
8. >> <<
9. & ^ |
10. == != <>
11. < <= > >=
12. = **= /= // = *= %= -= +=
13. not and or

توضیحات اضافی:

- ❖ برنامه Antlr همراه پروژه آپلود شده است. در صورت هرگونه مشکل می‌توانید آن را از بخش plugins در NetBeans 7.4، IntelliJ IDEA و یا Visual Studio Code نصب کنید.
- ❖ هرگونه قسمت مبهم در این زبان می‌تواند توسط شما مشخص و تعریف شود.
- ❖ فایل ارسالی شما می‌بایست فقط گرامر زبانی باشد که در صورت پروژه ذکر شده است. (فایل .g4).
- ❖ در هنگام تحویل، تسلط کامل به گرامر فرستاده شده و توانایی تغییر آن در سطح پروژه لازم است.

موفق باشید

تیم حل تمرین نظریه زبان‌ها و ماشین‌ها

