



به نام خدا



فاز اول پروژه کامپایلرها و زبان‌های برنامه‌نویسی

بهار 1400

مهلت تحویل : 23 فروردین

در فاز یکم پروژه، شما باید به کمک ابزار ANTLR4 و زبان برنامه‌نویسی جاوا، برای زبان Jepeto (که سند آن در اختیار شما قرار گرفته است) تحلیلگر لغوی و نحوی بنویسید.

تحلیل گر لغوی¹ :

در این بخش از پروژه، باید به کمک ابزار antlr4، تمامی Token های مورد نظر در زبان Jepeto را مشخص و پیاده سازی کنید .

تحلیل گر نحوی² :

در این بخش از پروژه، ابتدا با نوشتن قواعد نحوی صحیح، گرامر زبان Jepeto را به کمک antlr4 پیاده سازی می‌کنید سپس با اعمال ورودی های مورد نظر و با توجه به درخت Parse³ گرامر خود را تست و اطلاعات خواسته شده را چاپ می‌کنید. بهتر است برای قواعد خود نام‌های مناسب انتخاب کنید تا فهم آنها راحت تر باشد. هم چنین توجه داشته باشید که گرامر شما نباید مشکل چپ گردی و ابهام داشته باشد .

¹ Lexer

² Parser

³ درختی که بر اساس آن قواعد گرامر بررسی میشوند

در این فاز نیازی نیست که هیچ گونه قاعده معنایی⁴ را بررسی و پیاده سازی کنید . برای مثال وجود دو آرگومان هم‌نام، استفاده از متغیری که در آرگومان های توابع نیست ، شرطی بودن عبارت شرطی if، ناهماهنگی تایپ ها و . . . همگی از مواردی هستند که در فاز های بعدی بررسی می شوند و در این فاز نیازی به رسیدگی به آنها نیست.

مواردی که در پیاده سازی گرامر خود باید رعایت کنید به شرح زیر است:

- (1) برنامه باید دارای دقیقا یک main باشد.
- (2) بدنه توابع اگر تنها یک Statement است می تواند بدون تعریف block باشد در غیر این صورت باید به صورت block باشد.
- (3) هنگام فراخوانی توابع، آرگومان ها یا همگی به صورت key-value پاس داده می شوند و یا همگی باید به صورت بدون key آیند. (رجوع به قسمت 6 سند زبان)
- (4) بدنه توابع گمنام (anonymous functions) باید به صورت block باشد.
- (5) main آرگومان نمی پذیرد.
- (6) در بدنه main از ساختار شرطی نمی توان استفاده کرد.
- (7) بدنه main تنها شامل یک function Call یا یک Print می باشد.
- (8) هر تابع از هر مسیری که اجرا شود، باید حداقل یک گزاره return داشته باشد.
- (9) عبارت جلوی گزاره return نمی تواند خالی باشد.
- (10) قسمت شرط if باید یک expression باشد و نمی توان statement در آن قرار داد .
- (11) تابع print با بدنه خالی نداریم.
- (12) همانطور که قبلا گفته شد نیازی به بررسی هیچ گونه ناهماهنگی در تایپ ها ندارید . برای مثال یک لیست با عضو هایی که تایپ ها متفاوت دارند ، باید توسط برنامه شما پذیرفته شود.
- (13) اولویت عملگرها باید رعایت شود.

⁴ Semantic rules

(3) خروجی

پس از نوشتن تحلیل گر های لغوی و نحوی , باید به کمک Action هایی که با زبان java می نویسد و به گرامر خود اضافه می کنید خروجی های زیر را بر اساس پیمایش Pre-order درخت parse چاپ کنید :

1) در صورت مشاهده بخش main، نام آن را به صورت زیر (قبل از مشاهده ی دستورات داخل آن) چاپ کنید :

Main

2) هنگام رسیدن به تعریف هر تابع، نام تابع را به صورت زیر (قبل از مشاهده ی دستورات داخل آن) چاپ کنید :

FunctionDec : #name_of_function

3) هنگام رسیدن به تعریف آرگومان های یک تابع، نام آرگومان را به صورت زیر چاپ کنید :

ArgumentDec : #name_of_argument

4) در صورت مشاهده دستورات if و else، به صورت زیر آنها را چاپ کنید :

Conditional : if

Conditional : else

5) در صورت مشاهده دستور return آن را به صورت زیر چاپ کنید :

Return

6) در صورت مشاهده دستور print آن را به صورت زیر چاپ کنید :

Built-in : print

7) در صورت مشاهده فراخوانی یک تابع به عنوان یک گزاره (نه در expression) ، آن را به صورت زیر چاپ کنید:

FunctionCall

8) در صورت مشاهده تابع گمنام، آن را به صورت زیر (قبل از مشاهده دستورات داخل آن) چاپ کنید :

Anonymous Function

9) در صورت مشاهده فیلد size آن را به صورت زیر چاپ کنید :

Size

10) عبارت ها را در درخت عبارت به صورت Post-order پیمایش کنید. یعنی برای یک عملگر دوتایی ابتدا عملگر های اپرند اول و بعد عملگر های اپرند دوم و در نهایت خود آن عملگر چاپ شود .(عملگر یگانه مانند دوگانه ای است که یک فرزند دارد)

در صورت مشاهده یک عملگر، تنها خود عملگر را به صورت زیر چاپ کنید. عملگر های [] و () نباید چاپ شوند.

Operator : OperatorSymbol

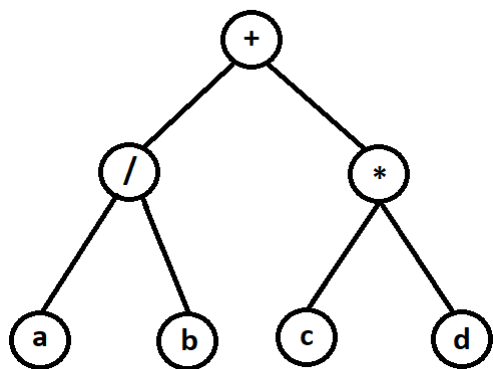
به عنوان مثال برای عبارت $a / b + c * d$ عملگرها را به صورت زیر چاپ کنید :

Operator : /

Operator : *

Operator : +

به درخت عبارت آن توجه کنید :



همچنین دقت کنید که عبارتهای قبل از : همگی کلیدواژه هستند و آنها را عینا چاپ کنید. تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خط های خالی و فاصله و ... نیز خودداری کنید.

نمونه ای از یک قطعه کد و خروجی آن در ادامه آمده است.

```
func f1 (a1, a2, a3, a4, a5) : {  
    f2(a6 = a5, a7 = 0);  
    print(a1 + a2 * (1 - 2 + 3) or true );  
    print(a4(5));  
    return f3("PLC") + 1400;  
    f4();  
    print(f5()[0]);  
    return true;  
}  
  
func f2(a6 , a7) : {  
    if a7 > a6.size - 1 :  
        return void;  
    print(a6[a7]);  
    f2 (a6, a7 + 1);  
    return void;  
}  
  
func f3(a9) :  
  
    if a9 is "PLC":
```

```

        return 1;
    else :
        return 2;
func f4() :
    return ()->{print("I'm reading an example code"); return void;};
func f5():
    return [1,2,3];

main : print(f1(1, "hi", true, (a8)->{return a8 + 1 * 2 + 3;}, [11 , false,
"code"]));

```

```

FunctionDec : f1
ArgumentDec : a1
ArgumentDec : a2
ArgumentDec : a3
ArgumentDec : a4
ArgumentDec : a5
FunctionCall
Built-in : print
Operator : -
Operator : +
Operator : *
Operator : +
Operator : or
Built-in : print
Return
Operator : +
FunctionCall
Built-in : print
Return
FunctionDec : f2
ArgumentDec : a6
ArgumentDec : a7
Conditional : if
Size
Operator : -
Operator : >
Return
Built-in : print

```

```
FunctionCall
Operator : +
Return
FunctionDec : f3
ArgumentDec : a9
Conditional : if
Operator : is
Return
Conditional : else
Return
FunctionDec : f4
Return
Anonymous Function
Built-in : print
Return
FunctionDec : f5
Return
Main
Built-in : print
Anonymous Function
ArgumentDec : a8
Return
Operator : *
Operator : +
Operator : +
```

نکات مهم:

- کد خود را به صورت یک فایل Jepeto.g4 آپلود کنید و نام اولین قانون زبان را jepeto بگذارید.
- در صورت کشف هرگونه تقلب، نمره 100- لحاظ میشود.
- دقت کنید که خروجیهای شما به صورت خودکار تست می شوند ؛ پس نحوه چاپ خروجی شما باید عیناً مطابق موارد ذکر شده در بالا باشد. علاوه بر آن، درخت parse شما نیز بررسی میشود.

- بهتر است سوالات خود را در فروم یا گروه درس مطرح نمایید تا دوستانتان نیز از آنها استفاده کنند؛ در غیر این صورت به مسئولان پروژه ایمیل بزنید:

nazaninyousefian@ut.ac.ir نازنین یوسفیان

arash3908@gmail.com آرش رسولی