

به نام خدا

فاز دوم پروژه نهایی کامپایلرها و زبان‌های برنامه‌نویسی

جمع آوری نام‌ها

بهار ۹۸

در فاز اول پروژه ، تحلیل گره لغوی و نحوی زبان Toorla را پیاده سازی کردید و با استفاده از وارد کردن قوانین معنایی در تحلیلگر نحوی ، برنامه ی داده شده را از text به AST متناظر تبدیل کردید ؛ حال بعد از تحلیل لغوی و نحوی برنامه ، به تحلیل معنایی آن میرسیم که برخی از مواردی که با استفاده از گرامر مستقل از متن قابل بررسی نبودند ، بررسی خواهند شد ، اما قبل از آن شما باید اطلاعاتی از برنامه ی داده شده به دست آورید و در جدول علایم ذخیره کنید که عبارتند از :

۱. کلاس‌ها.
۲. فیلد‌ها.
۳. متد‌ها.
۴. متغیرهای محلی (از جمله پارامترهای متد‌ها که متغیر محلی محسوب می شوند) .
۵. Scope های برنامه .

در این فاز باید به ازای متغیر محلی که در یک متد تعریف شده باید یک index نسبت داده شود . دقت کنید که باید برای indexing به این شکل عمل کنید که index اولین پارامتر یک متد باید حتما از ۱ شروع شود و سپس تا آخرین پارامتر به صورت auto increment به آنها index نسبت دهید . برای سایر متغیرهای محلی می توانید یا به صورت auto increment یا هر روش دیگر به آنها index نسبت دهید ، فقط توجه داشته باشید که این index ها چه برای پارامترها چه برای متغیرهای محلی یک متد (نه تمام متد ها) باید یکتا باشد و اینکه نباید index شان منفی باشد و باید از index آخرین پارامتر متد بزرگتر باشد ، یعنی در طول یک متد هیچ دو متغیر محلی (از جمله پارامترها) نباید index یکسانی داشته باشند (حتی اگر در Scope متفاوتی باشند) یعنی به طور مثال برای برنامه ی صفحه ی بعد ، index متغیرها برای متد main کلاس Eval به صورتی است که در ادامه توضیح داده خواهد شد :

```

class Eval :

    function main( a : string , t: int[] , y: int ) returns
    bool:
        var c = 2;
        begin
            var a = true;
            var b = false;
        end
        var b = 3;
        return false;
    end

end

```

در متد main کلاس Eval باید حتما index پارامتر های a و t و y به ترتیب ۱ ، ۲ و ۳ باشد ، اما index متغیر های محلی c و b در Scope اصلی متد و متغیر های a و b در Scope درونی متد ، میتوانند مقدار دلخواه بیشتر از ۳ داشته باشند (دقت داشته باشید که متغیر های محلی دیده شده در یک Scope نیز نباید index یکسانی حتی با متغیر بیرون Scope خود داشته باشند یعنی به طور مثال index متغیر b در Scope درونی باید متفاوت از متغیر b در خارج Scope باشد). اما برای این فاز index تمام متغیر ها را با روش **auto increment** مقدار دهی کنید تا تست کردن کدتان ساده تر شود . در غیر این صورت ممکن است نمره ی تست ها را از دست بدهید . یعنی به ازای کد بالا index متغیر های c و a و b درونی و b بیرونی به ترتیب ۴ تا ۷ خواهند شد .

در این فاز در هنگام جمع آوری اطلاعات برنامه داده شده به شما ممکن است به خطاهایی برخورد کنید که باید آنها را به فرمتی که در ادامه توضیح داده خواهد شد بدهید (دقت کنید که در صورت برخورد به هر کدام از خطاهای زیر ، اجرای کامپایلر شما به هیچ عنوان نباید خاتمه پیدا کند و باید با رفع موقتی خطای آن به تحلیل کد ادامه دهید) :

۱. عدم وجود کلاس در برنامه ی داده شده :

در صورت برخورد به این مورد باید به شکل زیر خطا دهید (در صورتی که از کد فاز ۱ آپلود شده استفاده نمی کنید) :

Error:There must be at least one class in a Toorla code

۲. تعریف دوباره ی یک کلاس موجود در برنامه:

در صورت برخورد به تعریف کلاسی که قبلا در برنامه دیده شده است ، باید به شکل زیر خطا دهید (فرض کنید اسم کلاس Test باشد):

Error:Line:#lineNumber:Redefinition of Class Test

میتوانید برای رفع خطای آن نام کلاس دوباره تعریف شده را تغییر دهید ، مثلا نام آن را به Temp_Test_i تغییر دهید که i یک عدد دلخواه است.

۳. تعریف دوباره ی متد و فیلد :

در صورت برخورد به تعریف متد یا فیلدی که قبلا در کلاس دربرگیرنده یا کلاس پدر آن دیده شده است ، باید به شکل زیر خطا دهید (فرض کنید اسم متد method و اسم فیلد هم field1 باشد):

Error:Line:#lineNumber:Redefinition of Method (Field) method (field1)

برای رفع خطای آن نیز می توانید همانند روشی که برای رفع خطا در حالات قبلی گفته شد عمل کنید .

۴. تعریف دوباره ی یک متغیر محلی در Scope جاری :

در صورت برخورد به تعریف متغیری که قبلا در همان Scope متغیری با همان نام تعریف شده است باید به شکل زیر خطا دهید (فرض کنید اسم متغیر محلی localVar باشد) :

Error:Line:#lineNumber:Redefinition of Local Variable localVar in current scope

برای رفع خطای آن نیز می توانید همانند روشی که برای رفع خطا در حالات قبلی گفته شد عمل کنید .

۵. تعریف فیلد length به عنوان فیلد یک کلاس :

در صورت برخورد به فیلدی به نام length در یک کلاس باید به شکل زیر خطا دهید :

Error:Line:#lineNumber:Definition of length as field of a class

برای رفع خطای آن نیز می توانید همانند روشی که برای رفع خطا در حالات قبلی گفته شد عمل کنید .
در صورتی که در این فاز به هیچ کدام از موارد بالا برخورد نکردید ، برنامه داده شده را به همان فرمتی که را در فاز ۱ چاپ میکردید چاپ کنید با این تفاوت که به تعریف هر متغیر محلی که رسیدید (از جمله پارامترهای متدها) به جای چاپ کردن اسم آن ، باید اسم آن را به همراه index اش چاپ کنید ، یعنی به طور مثال برای تکه کد صفحه ی بعد برنامه ی شما باید چنین خروجی بدهد (identifier های bold شده تعریف یک متغیر می باشند) :

```

function main( a : int , b : string[] , c: US ) returns int :
    var a1 = 2 , b1 = 4 , c1 = 5;
    if( a == 2 ) begin
        return a;
    end
end

```

```

(( entry class (Identifier,Main) ((ACCESS_MODIFIER_PUBLIC) method (Identifier,main)( (Identifier,a_1) :
(ArrayType,(IntType))) ( (Identifier,b_2) : (ArrayType,(StringType))) ( (Identifier,c_3) : (UserDefined,US))
(IntType) (
(var (Identifier,a1_4) (IntValue,2))
(var (Identifier,b1_5) (IntValue,4))
(var (Identifier,c1_6) (IntValue,5))
(if (== (Identifier,a) (IntValue,2)) (
(return (Identifier,a))
)
(Skip)
)
)
)
)
)
)
)
)
)
)

```

بسیار دقت داشته باشید که ترتیب دادن خطاها در این فاز از اهمیت بسیار بالایی برخوردار است و خطاها بر حسب شماره ی خطوط و اگر شماره ی خطوط مساوی بود باید بر حسب پیمایش preorder درخت بدست آمده از برنامه مرتب باشد .

ابزار مدیریت پروژه ی maven :

در این فاز از پروژه ، شما می توانید با استفاده از ابزار maven ، پروژه ی خود را پیاده سازی کنید . استفاده از این ابزار نمره ی امتیازی خواهد داشت . هدف استفاده از این ابزار ، استاندارد و ساده سازی کردن test ، build کردن و مدیریت گزارش دهی پروژه (از قبیل وابستگی های پروژه ی شما به کلاس ها و کد های مختلف) و بهبود بخشیدن سرعت توسعه ی آن است که برای پروژه های با زبان Java کاربرد دارد .

برای آشنایی با ابزار maven می توانید به [اینجا](#) مراجعه کنید . در این فاز از پروژه ، کد داده شده به شما در واقع یک پروژه ی maven است و شما فقط کافیسیت ساختار فایل همان کد را رعایت کنید و فایل pom.xml موجود در پروژه را که در واقع وابستگی های پروژه ی maven شما و برخی موارد دیگر از جمله plugin های مورد نیاز برای پروژه ی شما را توصیف می کند را تغییر ندهید . در این صورت می توانید با استفاده از دو دستور زیر پروژه ی خود را در هر سیستم عاملی که روی آن کار می کنید کامپایل و اجرا کنید (دقت کنید که این دو دستور باید حتما در فولدری اجرا شوند که فایل pom.xml در آنجا قرار دارد) :

1. Compile Command : mvn compile

2. Running a Java Class Using maven : mvn -q exec:java -Dexec.mainClass='runnable class' -Dexec.args='java_class_arguments'

اگر از q- در دستور دوم استفاده کنید فقط خروجی حاصل از اجرای پروژه ی شما را نمایش خواهد داد و در غیر این صورت علاوه بر خروجی پروژه تان ، برخی log های maven نیز در کنسول به نمایش آورده خواهند شد. اگر هم نمی خواهید از maven استفاده کنید فقط کافیسیت فایل pom.xml را از کد داده شده حذف نمایید و با همان روش قبلی کارتان را انجام دهید .

نکات مهم :

- اگر پیاده سازی پروژه ی خود را با ابزار maven انجام می دهید ، شما باید کل فولدری که شامل فایل pom.xml می باشد را در قالب یک فایل زیپ به فرمت SID1_SID2.zip آپلود نمایید ، در غیر این صورت فایل های ToorlaCompiler.java به همراه Toorla.g4 و پکیج toorla از کد داده شده که فرض می شود کد هایی که زده اید در این پکیج هستند را آپلود نمایید . بسیار دقت داشته باشید که اگر با استفاده از ابزار maven پروژه تان را پیاده سازی می کنید ، پروژه ی شما باید در این فولدر با استفاده از دو دستور گفته شده بی هیچ تغییری در ساختار فایل پروژه تان بعد از extract کردن آن کامپایل و اجرا شود .
- در این فاز ، کد فاز ۱ به همراه کد جدول علائم در اختیار شما قرار داده شده است که می توانید از آنها استفاده کنید ، البته می توانید از کد فاز قبل خودتان استفاده کنید ، اما در این صورت مسئولیت عواقب احتمالی آن به عهده ی خودتان خواهد بود .
- همانطور که در سند زبان Toorla گفته شده ، شما در طول هیچ کدام از فاز ها نباید method overriding را هندل کنید و این موضوع نمره اضافی برای پروژه ی شماست . اگر خواستید این موضوع را پیاده سازی کنید باید به صورت جدا از تمام فاز های پروژه آن را تحویل دهید .
- مواردی هستند که در این گزارش اشتباهی گفته شده اند که خودتان باید آنها را پیدا و به شکل درست آنها را پیاده سازی کنید و این موارد در تست ها بررسی خواهند شد .
- دقت کنید که تست های شما در این فاز به صورت خودکار اجرا خواهند شد . پس بسیار به ترتیب و فرمت خطا دادن و چاپ کردن خطاهای برنامه داده شده دقت داشته باشید تا نمره ی تست ها را از دست ندهید .

- چنانچه سوالی در مورد پروژه داشتید در فروم درس مطرح کنید تا سایر دانشجویان نیز استفاده کنند ، چنانچه ابهامی داشتید نیز می توانید به صورت حضوری مراجعه کنید و یا ایمیل بزنید و بپرسید .

موفق باشید .