



پروژه‌ی درس طراحی کامپیوتر

فاز ۲ پروژه: ساختاریاب

آقای بهرامی

نسخه ۱

موعد تحویل: ۲۵ خرداد

## ۱ مقدمه

در این فاز پروژه قرار است ساختاریاب را پیاده‌سازی کنید و به واژه‌یابی که در فاز قبلی پروژه پیاده‌سازی کرده‌اید متصلش کنید (البته در صورتی که قصد انجام پروژه با استفاده از LARK را دارید، کارهایی که در فاز قبلی کردید در ادامه به کار نمی‌آیند ولی خود LARK بسیاری از بخش‌های واژه‌یاب را درون خودش دارد. آن تعدادی را هم که ندارد می‌توانید با استفاده از Regex به آن معرفی کنید). توجه کنید که پس از پیاده‌سازی این بخش، از آنجایی که هنوز به کدساز نرسیده‌ایم، نمی‌توان کدی را اجرا کرد و از صحت آن مطمئن شد. به همین دلیل تنها از شما می‌خواهیم که با گرفتن یک کد ورودی به زبان Decaf به ما بگویید که آیا خطای نحوی<sup>۱</sup> دارد یا نه.

## ۲ انواع خطاهای زمان کامپایل

گفتیم که باید در این بخش مشخص کنید که برنامه خطای نحوی دارد یا نه. یعنی چه؟ بگذارید نگاهی به انواع خطاهای زمان کامپایل بیاندازیم:

- خطای واژه‌ای<sup>۲</sup>  
شامل غلط‌های املایی در نام شناسه‌ها، کلیدواژه‌ها و عملگرها است. این خطاها عموماً توسط واژه‌یاب کشف می‌شوند.
- خطاهای نحوی<sup>۳</sup>  
خطاهایی از قبیل جا افتادن؛ و یا درست نبودن پرانتزگذاری و ... که در واقع رعایت نکردن syntax برنامه‌اند از این قبیل خطاها اند. این گونه خطاها عموماً توسط ساختاریاب کشف می‌شود.
- خطاهای معنایی<sup>۴</sup>  
خطاهایی نظیر استفاده از متغیر تعریف نشده یا عدم همخوانی نوع در هنگام انتساب و ... از این دست خطاها هستند. این موارد عموماً توسط کدساز کشف می‌شوند.
- خطاهای منطقی<sup>۴</sup>  
به این معنی است که منطقی که برنامه انجام می‌دهد با چیزی که برنامه‌نویس انتظارش را داشته متفاوت است. ولی برنامه درست کامپایل می‌شود و اجرا می‌شود. همچنین حلقه‌های نامتناهی و کدهای غیرقابل دسترس از این دست خطاها هستند. اکثر خطاهای منطقی به هیچ عنوان قابل کشف نیستند.

منبع این بخش اینجا است. می‌توانید برای مطالعه‌ی بیشتر به آن مراجعه کنید.

## ۳ ورودی و خروجی

برنامه‌ی شما در ورودی چیزی را دریافت می‌کند که انتظار می‌رود یک کد به زبان Decaf باشد. ولی ممکن است یک دنباله‌ی کاملاً بی‌معنی از کاراکترها باشد. در خروجی، اگر کد ورودی دارای خطای نحوی بود،

syntax-error<sup>۱</sup>  
lexical-error<sup>۲</sup>  
semantical-error<sup>۳</sup>  
logical-error<sup>۴</sup>

عبارت YES و در غیر این صورت اگر به هر طریقی با استفاده از گرامر داده شده در مستند توصیف زبان Decaf قابل تولید بود عبارت NO را چاپ کنید.

برای مثلاً به برنامه‌های زیر توجه کنید:

```
public class Main { // error 1: program should start with 'class', not
    identifier 'public'
    static void main() {
        int x = 1;
        int y;
        y = 1;
        z[] = 2; // error 2: missing index for '[]'
    }
}
```

همانطور که در کامنت‌ها مشاهده می‌کنید، این برنامه دارای دو خطای نحوی می‌باشد. پس در خروجی باید YES چاپ شود. کد زیر نیز خطای نحوی دارد و باید پیغام YES در خروجی چاپ شود:

```
class Main {
    int x,y; // error 1: this grammar is not supported

    static void main() {
        x = 0xf;
        y = 0xff;
        ;
        Print(z,); // error 2: missing parameter after ','
        return 0;
    }
}
```

## ۴ چند نکته

- برای هرکدام از روش‌های پیاده‌سازی، یک bash script به عنوان قالب به شما داده می‌شود. حتماً دقت کنید که پروژه‌تان با استفاده از این script قابل اجرا روی هر ماشینی باشد. چرا که در هنگام تحویل، تنها از این طریق برنامه‌تان تست خواهد شد.
- دقت کنید که تمام بخش‌های پروژه باید توسط خود شما پیاده‌سازی شوند.
- در صورتی که از منابعی استفاده می‌کنید، حتماً آنها را ذکر کنید.
- در صورت مشاهده‌ی تقلب مطابق با سیاست‌های درس برخورد خواهد شد.
- سعی کنید پروژه را زودتر شروع کنید. با این که پروژه در سه فاز تقسیم شده، ولی اگر پیاده‌سازی را به روزهای نزدیک به ددلاین بیاندازید دچار مشکل خواهید شد.