



پروژه‌ی درس طراحی کامپیوتر

فاز ۱ پروژه: واژه‌یاب

نسخه ۱.۱

موعد تحویل: ۲ آبان

## ۱ مقدمه

پیش از این که به سراغ مطالعه‌ی این مستند بروید، توصیه می‌کنیم که حتما مستند مربوط به معرفی زبان Decaf را مطالعه کنید.

در طول این پروژه که شامل ۳ فاز خواهد بود، شما به ترتیب قرار است واژه‌یاب<sup>۱</sup>، ساختاریاب<sup>۲</sup> و نهایتاً کدساز<sup>۳</sup> را پیاده‌سازی کنید. در واقع پس از انجام فاز آخر کامپایلر شما کامل می‌شود. همچنین این کار را باید با استفاده از یکی از ۴ روش زیر انجام دهید:

- استفاده از Flex برای واژه‌یاب و سپس استفاده از Bison برای ساختاریاب و کدساز. در این حالت، ابتدا با استفاده از Flex واژه‌یاب را پیاده‌سازی می‌کنید و پس از آن گرامر (۱) LARL را به Bison می‌دهید و آن را به واژه‌یاب‌تان وصل می‌کنید. سپس کد ساز را پیاده‌سازی می‌کنید.
- استفاده از Java CUP، در جاوا. این ابزار با گرفتن گرامر (1) LALR یک پارسر می‌سازد و این امکان را به شما می‌دهد که هنگام پارس کردن گرامر توابع دلخواه خود را در بخش‌هایی که می‌خواهید اجرا کنید و به این وسیله می‌توانید درخت پارس و کدساز را پیاده‌سازی کنید.
- استفاده از ابزار pGen برای تولید درخت پارس و سپس استفاده از پایتون برای پیاده‌سازی باقی کامپایلر. ابزار pGen با گرفتن گراف نحو، درخت پارس مربوط به گرامر را خروجی می‌دهد. در این حالت برای واژه‌یاب می‌توانید از Regex ها استفاده کنید.
- استفاده از کتابخانه‌ی Lark در پایتون (دقت کنید که در صورت انتخاب این روش، تنها می‌توانید از حالت (۱) LALR موجود در آن برای ساختاریاب استفاده کنید). این ابزار با گرفتن گرامر (۱) LALR و کمی تنظیمات، کار کدساز و ساختاریاب را انجام می‌دهد و ساختاریاب را به کدساز وصل می‌کند.

در کلاس‌های حل تمرین بیشتر با این ابزارها آشنا خواهید شد.

## ۲ کلیت این فاز

پس از این که ابزاری که می‌خواهید با آن پروژه را پیاده‌سازی کنید انتخاب کردید، باید به پیاده‌سازی واژه‌یاب بپردازید. در این بخش به عنوان ورودی یک stream از کاراکترها می‌گیرید و باید آنرا به نشانه‌ها<sup>۴</sup> بشکنید. دقت کنید که هیچگونه عمل ساختاریابی‌ای در این بخش انجام نمی‌شود و هر نشانه مستقل از نشانه‌های قبلی (به صورت بدون حافظه) پیدا می‌شود. واژه‌یاب، فضاهای سفید<sup>۵</sup> که خارج از ثوابت رشته‌ای و کاراکتری‌اند را در نظر نمی‌گیرد و از آنها می‌گذرد. و باید کلیدواژه‌ها، ثوابت integer و double و رشته‌ای و کاراکتری و بولین، عملگرها و شناسه‌ها<sup>۶</sup> را تشخیص دهد. همچنین برای کلماتی که مشکلات واژه‌ای دارند خطا دهد.

scanner<sup>۱</sup>  
parser<sup>۲</sup>  
codegen<sup>۳</sup>  
tokens<sup>۴</sup>  
whitespaces<sup>۵</sup>  
identifiers<sup>۶</sup>

### ۳ ورودی‌ها و خروجی‌ها

همانطور که گفته شد در ورودی دنباله‌ای از کاراکترها به شما داده می‌شود. برای مثال ورودی زیر را در نظر بگیرید:

```
class Program { void main () {} }
```

خروجی واژه‌یاب برای این برنامه مطابق زیر خواهد بود:

```
class
T_ID Program
{
void
T_ID main
(
)
{
}
}
```

حال به مثال کمی پیچیده‌تر زیر توجه کنید:

```
{-123-a35,id3a,++;}[| |==!=()&&]<><=>==
a[24]="7"; n!=if;
false,-if>true32;
forpar
```

همانطور که مشاهده می‌کنید در اینجا واژه‌یاب باید نشانه‌های به هم چسبیده‌ای را از هم تشخیص دهد.

خروجی آن چیزی شبیه به این خواهد بود:

```
{
-
T_INTLITERAL 123
-
T_ID a35
,
T_ID id3a
,
+
*
;
}
[
| |
==
=
!=
(
)
&&
]
<
>
```

```

<=
>=
=
ID a
[
T_INTLITERAL 24
]
=
T_STRINGLITERAL "7"
;
T_ID n
!=
if
;
T_BOOLEANLITERAL false
,
-
if
;
T_ID true32
;
T_ID forpar

```

دقت کنید که واژه یاب حتی اگر به یک نشانه رسیده باشد، تا جایی که با اضافه شدن کاراکتر بعدی به نشانه‌ی فعلی کماکان نشانه‌ای معتبر داشته باشیم، ادامه می‌دهد.

همچنین همانطور که در مثال‌ها مشخص است، برای Literal ها و شناسه‌ها، علاوه بر نوع نشانه‌ی تشخیص داده شده، باید خود نشانه را نیز ذکر کنید. مثلاً برای شناسه‌ی true32 در خروجی مقدار زیر چاپ شده:

```
T_ID true32
```

### ۱.۳ انواع نشانه‌ها

برای عمل‌گرها و کلیدواژه‌ها کافی است خودشان را خروجی دهید. برای شناسه‌ها، نوع نشانه T\_ID است. برای ثوابت integer و double و رشته و کاراکتر و boolean نیز نشانه‌ها از این قرار هستند:

```

integer: T_INTLITERAL
double: T_DOUBLELITERAL
string: T_STRINGLITERAL
bool: T_BOOLEANLITERAL

```

### ۴ چند نکته

- برای این فاز باید پاسخ خود را که یک فایل که به زبان‌های C++، Java، یا python نوشته شده است را در سامانه کوئرا آپلود کنید. نمره‌دهی در آنجا صورت می‌گیرد. توجه داشته باشید که ممکن است در طی زمان تست‌های پیشرفته‌تری به مجموعه تست‌ها اضافه شوند برای همین، از نمره‌ی

داده شده در کوئرا غافل نشوید تا در صورت وجود هرگونه ایراد بتوانید به سرعت کد خود را اصلاح کنید.

- برای هرکدام از روش‌های پیاده‌سازی، یک script bash به عنوان قالب به شما داده می‌شود. حتما دقت کنید که پروژه‌تان با استفاده از این script قابل اجرا روی هر ماشینی باشد. از این فایل برای تحویل فازهای بعدی استفاده خواهد شد. این فایل به شما نشان می‌دهد که چه تعداد از تست‌ها را به درستی پاسخ داده‌اید. پیشنهاد می‌کنیم از همین ابتدا سعی کنید که پروژه را توسط این script اجرا کنید تا در فازهای بعد کار راحت‌تری داشته باشید.
- دقت کنید که تمام بخش‌های پروژه باید توسط خود شما پیاده‌سازی شوند.
- در صورتی که از منابعی استفاده می‌کنید، حتما 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.