



پروژه فاز اول (اسکنر) درس کامپایلر

استاد : دکتر رضا اکبری

تهیه کننده : سید محمد روزگار

3	مقدمه :
4	توضیحات :
5	پیاده سازی :
7	ورودی و خروجی پروژه :
7	تحويل پروژه :
7	زمان تحويل و ارائه فاز اول پروژه :
8	نمره دهی و ارزیابی فاز اول پروژه :

مقدمه :

این حجم زیاد مستندات تنها به منزله عدم گمراهی شما می باشد و به معنای حجیم بودن پروژه نخواهد بود .

پروژه درس کامپایلر از دو فاز تشکیل شده است :

1- تحلیل گر لغوی (Lexical Analyzer)

2- تحلیل گر نحوی (parser)

در این فاز قصد داریم توضیحاتی در مورد فاز اول پروژه یعنی تحلیل گر لغوی بدهیم و فاز بعدی در آینده تعریف خواهد شد .

در فاز اول پروژه قصد داریم یک کامپایلر با قابلیت های کم برای زبان جاوا (java) طراحی کنیم . برای این پروژه 20 ورودی در اختیار شما قرار می گیرد که باید پروژه خود را با این ورودی ها تست کنید . (مسلماً تمام ورودی ها ، یک ورودی صحیح نیستند !)

این پروژه به صورت انفرادی است و در نهایت هر فرد موظف است پروژه خود را در سامانه مودل در محل مشخص شده آپلود نماید .

برای این پروژه خود نیاز است در گیتهاب (GitHub) یک مخزن (repository) به نام **compiler-1401** بسازید و قدم به قدم انجام پروژه را در آنجا کامیت (commit) کنید .

هر گونه ابهام و مشکلی که در مورد پروژه داشتید می توانید از طریق شخصی واتساپ بیان کنید .

✓ شماره TA : 09178507015 (روزگار)

در صورت مشابهت بیش از حد پروژه دو نفر ، از دو فرد نمره کسر خواهد شد لذا کامیت های شما در گیتهاب می تواند معیار خوبی برای نمایش میزان فعالیت شخصی شما باشد .

✓ تمامی اطلاعات تکمیلی و همچنین ورودی هایی که نیاز است با آنها کدی که نوشتید را تست کنید به مرور زمان در [این آدرس](#) قرار خواهد گرفت .

توضیحات :

در این پروژه نیاز است قسمتی را جهت خواندن ورودی باشد پیاده کنید که زمانی که ورودی را به صورت فایل را متنی به آن می دهید ، بتواند توکن ها را تشخیص دهد .

در صورتی که توکن تشخیص داده شد ، نیاز است آن توکن به همراه مقدار آن را به جدول نماد بفرستید و در نهایت جدول نماد خود را چاپ کنید .

✓ توجه داشته باشید در این فاز جهت سادگی نیازی نیست خطا ها را تشخیص دهید لذا اگر خطای خاصی که حس کردید بهتر است در این فاز تشخیص دهید را میتوانید گزارش کنید تا برای فاز دوم پروژه که شامل مازول تشخیص خطا است ، راحت تر باشید .

شما می توانید بر حسب اختیار پروژه خود را به صورت دستی (کد نویسی زیاد) یا به صورت خودکار بنویسید . (پیشنهاد من استفاده از ابزار های خودکار است چون خیلی کد کمتری نیازی هست بنویسید و همچنین کار خود را برای فاز بعدی ساده تر می کنید !!!)

✓ در روش دستی می توانید از ساختار زبان های برنامه نویسی نظیر شروط if-else و یا switch-case که خصوصیات یک دیاگرام انتقال را در اختیارتان قرار می دهد، استفاده کنید.

✓ در روش خودکار می توانید از نرم افزار های Lex نظیر **Flex** (تعبیه شده برای زبان C و C++) و **jFlex** (تعبیه شده برای زبان java) و سایر ابزار های مشابه به اختیار خودتان استفاده کنید. (بهتر است هر ابزاری که استفاده می کنید یک نیم نگاهی به ابزار پارسری که با این اسکندر سازگار است ، بیاندازید تا در آینده به مشکلی بر نخورید .) (دو ابزاری که در بالا معرفی شد ، به دلیل وجود پارسر های قدرتمند سازگار با آن ها ، در آینده به مشکلی بر نخواهید خورد .)

✓ شما می توانید ترکیبی از روش های دستی و خودکار را استفاده کنید . (چیزی که در نهایت مهم است منطق کد شما است و خروجی که چاپ می کند .)

پیاده سازی :

در کامپایلر شما نیاز است موارد زیر در نظر گرفته شود :

- تعریف متغییر به همراه نوع آنها
- تعریف حلقه های `for` و `while`
- دستور شرطی `if-else`
- عملگر انتساب (`=`)
- کامنت تک خطی و چند خطی
- عملگر های ریاضی
 - جمع (+) و تفریق (-)
 - ضرب (*) و تقسیم (/)
 - باقی مانده تقسیم یا همان `mod (%)`
 - عملگر های منطقی `AND` و `OR (&& و ||)`
- عملگر های مقایسه
 - مساوی و نامساوی (`==` و `!=`)
 - کوچکتر ، کوچکتر یا مساوی ، بزرگتر و بزرگتر یا مساوی (`<` ، `<=` ، `>` و `>=`)
- عملگر های حسابی افزایش و کاهش (`++` و `--`)
- دستور `switch-case`

کلمات کلیدی زبان نیاز است که موارد زیر را در نظر داشته باشد :

"for", "while", "if", "else", "switch", "case", "break", "else if", "void",
"default", "continue", "return", "long", "int", "String", "char",
"boolean", "double", "float"

✓ عبارت های true/false ، رشته های تشخیص داده شده ، مقادیر کارکتر و اعداد (صحیح یا اعشاری) را می توان به عنوان literal در نظر گرفت .

✓ در این فاز جهت سادگی نیازی نیست بررسی کنید که آیا تطابقی بین نوع متغیر تعریف شده و مقدار انتساب داده شده به آن وجود دارد یا نه .

▪ مثلاً اگر عبارتی چون `int a = 'c';` وجود داشت نیازی نیست خطایی را گزارش کنید .

✓ در صورتی که نتوانستید نوع متغیر و یا مقدار آن را تشخیص دهید می توانید خط تیره و یا عبارت undefined را در جدول نماد چاپ کنید .

نمونه ای از جدول نماد برای کد زیر :

```
void main(){  
    int a;  
    b = 12;  
    //this is comment  
    int c = "ali";  
}
```

id	name	type	value
1	a	int	undefined
2	b	undefined	12
3	c	int	"ali"

ورودی و خروجی پروژه :

▪ ورودی :

○ فایل یا متن حاوی دستورات برنامه

▪ خروجی :

○ لیست توکن ها به ترتیبی که شناسایی کردید

○ جدول نماد حاوی شناسه ها و مشخصات آنها

✓ نیاز است یک ماژول جهت خواندن ورودی نیز تعبیه کنید که ورودی را بخواند و به اسکنر بدهد . (البته اگر از ابزار های خودکار استفاده کنید ، معمولاً این ماژول را درون خود دارد و تنها نیاز است آن را فعال کنید .)

تحويل پروژه :

همانطور که در مقدمه گفته شد نیاز است پروژه خود را در گیتهاب قرار دهید و فایل نهایی را در سامانه مودل دانشگاه آپلود نمایید .

دقت کنید جهت جلوگیری از اشتراک گذاری کد ، مخزن یا repository گیتهاب شما بایستی به صورت خصوصی یا private تعریف شده باشد و تنها آیدی TA را در قسمت collaborators اضافه نمایید .

✓ آیدی TA در گیتهاب : **rouzegarCompiler** (روزگار)

زمان تحويل و ارائه فاز اول پروژه :

مهلت تحويل : 3 اردیبهشت 1401 ساعت 23:59

زمان ارائه پروژه : متعاقباً اعلام خواهد شد .

نمره دهی و ارزیابی فاز اول پروژه :

40%	ارائه شفاهی
10%	رعایت نکات گفته شده
30%	انجام پروژه
20%	استفاده از گیت و گیتهاب
5%	کامنت گذاری منظم
5%	ساخت فایل Readme مناسب برای پروژه
100% + 10%	مجموع

✓ در فایل Readme.md که برای مخزن گیتهاب خود ایجاد می کنید ، نیاز است که توضیحاتی در مورد پروژه خود و نحوه ی اجرای پروژه نوشته شده ، بدهید . (راهنمایی : برای نوشتن این فایل به زبان نشانه گذاری markdown نیاز دارید ، لذا اگر در مورد نحوه نوشتن این فایل به مشکلی برخوردید می توانید به [این آدرس](#) مراجعه کنید).