



## طراحی کامپایلر و زبان‌های برنامه‌نویسی

تمرین کامپیوتری ۱

دستیاران آموزشی: شایان کاشفی، علی امامزاده

بهار ۱۴۰۲

### ۱. مقدمه

در این تمرین، شما باید به کمک ابزار ANTLR و زبان برنامه‌نویسی جاوا، برای زبان LogicPL (که سند آن را در اختیار دارید) تحلیلگر لغوی و نحوی بنویسید و در انتها خروجی مورد نظر را با استفاده از اکشن‌های جاوا ای که می‌نویسید تولید کنید. اهداف کلی این فاز شامل موارد زیر خواهد بود:

1. تحلیلگر لغوی: برای این بخش از تمرین باید به کمک ابزار ANTLR، تمامی token های مورد نظر خود را مشخص کرده و پیاده‌سازی کنید. برای اینکار باید در فایل گرامر خود توکن های مورد نیاز زبان را با استفاده از عبارات منظم تعریف کنید.

2. تحلیلگر نحوی: در این بخش از تمرین، ابتدا با نوشتن قواعد نحوی صحیح، گرامر زبان LogicPL را به کمک ANTLR پیاده‌سازی می‌کنید. بهتر است برای هر کدام از قواعد گرامر خود نام مناسب انتخاب کنید و سپس با اعمال ورودی های مناسب و بررسی درخت parse گرامر خود را تست کنید. توجه داشته باشید که در این فاز نیازی به پیاده‌سازی قواعد معنایی برای بررسی ارورها نیست. برای مثال، مواردی مانند وجود دو متغیر هم‌نام، ارجاع به متغیری که تعریف نشده، بررسی RValue و LValue بودن عملوندها، و ... همگی از موارد مربوط به تحلیل معنایی هستند که در فازهای بعدی بررسی می‌شوند. البته انتظار می‌رود موارد مربوط به تحلیل لغوی و نحوی موجود در سند زبان را بصورت کامل پیاده‌سازی کنید و این موارد در تستهای خودکار نیز مورد بررسی قرار می‌گیرند. گرامر شما در این فاز باید به نحوی باشد که اولویت عملگرها در آن رعایت شده باشد و همچنین فاقد چپگردی یا ابهام باشد

## ۲. فرمت خروجی

در ادامه، پس از نوشتن کد تحلیلگر لغوی و نحوی، باید به کمک action هایی که به زبان جاوا می نویسد و به گرامر خود اضافه می کنید node های درخت parse را به صورت pre-order چاپ کنید:

1. 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
```

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

```
ArgumentDec: #name_of_argument
```

3. با رسیدن به تعریف هر متغیر (آرایه یا متغیر عادی)، نام آن را به صورت زیر چاپ کنید:

```
VarDec: #name_of_variable
```

4. در صورت مشاهده ی دستور return آن را به صورت زیر چاپ کنید:

```
Return
```

5. در صورت مشاهده دستور print آن را به صورت زیر چاپ کنید:

```
Built-in: print
```

6. در صورت رسیدن به بدنه ی اصلی کد (قسمت main) آن را به صورت زیر چاپ کنید:

```
MainBody
```

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

```
FunctionCall
```

8. در صورت مشاهده ی دستور for، آن را به صورت زیر چاپ کنید:

```
Loop: for
```

9. در صورت رسیدن به یک predicate نام آن را به صورت زیر چاپ کنید:

Predicate: #name\_of\_predicate

10. در صورت رسیدن به یک عبارت implication آن را به صورت زیر چاپ کنید:

Implication

عبارت‌ها را در درخت عبارت به صورت **post-order** پیمایش کنید. یعنی برای یک عملگر دوتایی ابتدا **عملگرهای عملوند اول**، سپس عملگرهای عملوند دوم و در نهایت خود آن عملگر چاپ شود (در درخت عبارت، عملگر تک عملندی به صورت یک عملگر دو عملوندی در نظر گرفته شود که یک فرزند دارد).

در صورت مشاهده‌ی یک عملگر، تنها خود عملگر را به صورت زیر چاپ کنید. **عملگرهای [ ] ( )** نباید چاپ شوند.

Operator : #operator\_symbol

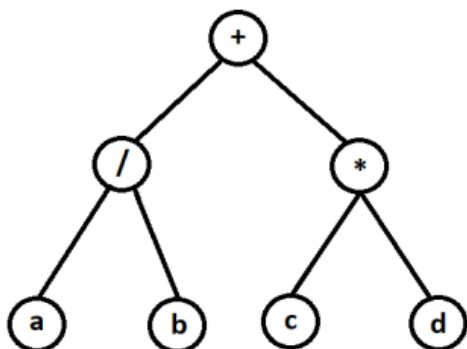
به عنوان مثال برای عبارت  $a / b + c * d$  عملگرها را به صورت زیر چاپ کنید :

Operator : /

Operator : \*

Operator : +

همچنین، درخت عبارت parse این عبارت به صورت زیر خواهد بود.



**نکته:** تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خط‌های خالی و فاصله و ... نیز خودداری کنید.

**نکته:** جهت بررسی صحت کارکرد پارسر خود می‌توانید پس از کامل کردن فایل grammar.g4 برای ساخت پارسر توسط ANTLER رو فایل کلیک راست کرده و گزینه Generate ANTLER Recognizer را انتخاب کنید. پس از تولید شدن کدهای پارسر با run کردن سمپل روی کد، خروجی مورد نظر را تولید کنید.

#### ۴. نکات نهایی

- می‌توانید پروژه را به صورت گروهی انجام دهید.
- پس از آپلود فایل پروژه ، یک جلسه تحویل پروژه خواهید داشت که متعاقبا اعلام خواهد شد.
- دقت کنید که خروجی های شما به صورت خودکار تست می‌شوند. لذا لازم است اطمینان حاصل کنید که خروجی‌های شما دقیقا همانند خروجی‌های خواسته شده باشد. در صورت عدم توجه به این نکته، متاسفانه بررسی دستی کد امکان پذیر نخواهد بود.
- می‌توانید سوالات خود را از طریق ایمیل یا در گروه درس مطرح نمایید.
- توجه کنید که در صورت احراز تقلب به هر نحو ( دسترسی به کد دیگران، استفاده از کدهای ترم‌های قبل، استفاده از کدهای موجود بصورت آنلاین و ...) ، مطابق قوانین درس با آن برخورد خواهد شد.
- کد خود را به صورت یک فایل LogicPL.g4 آپلود کنید و نام اولین قانون زبان را LogicPL بگذارید. توجه کنید که با وجود ارزیابی خودکار در صورت عدم توجه به این موضوع و از دست دادن نمره هیچ عذری پذیرفته نخواهد بود.
- آپلود فایل نهایی پروژه فقط توسط یکی از اعضای گروه صورت پذیرد.