



## به نام خدا

تمرین کامپیوتری اول درس طراحی کامپیوتر

بهار ۱۴۰۳

مهلت تحویل: ۲۶ فروردین ۱۴۰۳



## فهرست مطالب

1	فهرست مطالب
2	تحلیل‌گر لغوی
2	تحلیل‌گر نحوی
2	Action-ها
2	پیمایش درخت parse
4	پیمایش عبارت‌ها
5	نکات مهم

## تحلیل‌گر لغوی

برای این بخش از پروژه، باید به کمک ابزار antlr4، تمامی token های مورد نظر خود را مشخص کرده و پیاده‌سازی کنید.

## تحلیل‌گر نحوی

- در این بخش از پروژه، ابتدا با نوشتن قواعد نحوی صحیح، گرامر زبان FunctionCraft را به کمک antlr4 پیاده‌سازی می‌کنید.
- بهتر است برای هر قاعده گرامری نام مناسبی انتخاب کنید و سپس با اعمال ورودی‌های مناسب و تشکیل درخت parse، گرامر خود را تست نمایید.
- توجه داشته باشید که در این مرحله، نیازی نیست هیچگونه قاعده معنایی را پیاده‌سازی کنید. برای مثال مواردی از قبیل وجود دو متغیر هم نام، ارجاع به متغیری که که تعریف نشده، بررسی نوع عملوندهای عملگرها همگی از مواردی هستند که مربوط به تحلیل معنایی زبان هستند و در این تمرین نیازی نیست برای آنها اقدامی صورت گیرد.
- لازم به ذکر است که گرامر شما نباید شامل چپ‌گردی یا ابهام باشد.

## action-ها

### پیمایش درخت parse

- پس از نوشتن تحلیل‌گر لغوی و نحوی، باید با استفاده از action-هایی که با زبان جاوا بنویسید و به گرامر خود اضافه می‌کنید، خروجی‌های زیر بر اساس پیمایش pre-order در درخت parse چاپ شوند:
- با رسیدن به تابع main، خروجی به صورت زیر چاپ شود:

MAIN

- با رسیدن به توابع تعریف شده توسط کاربر، خروجی به صورت زیر چاپ شود:
- FuncDec: "function name"
- با رسیدن به تعاریف pattern matching، خروجی به صورت زیر چاپ شود:
- PatternDec: "pattern name"

- با رسیدن به return statement خروجی زیر را چاپ کنید:

RETURN

- با رسیدن به گزاره‌های شرطی، خروجی‌های متناسب با keyword های به کار رفته به صورت زیر چاپ شوند:

Decision: IF

Decision: ELSE IF

Decision: ELSE

- با رسیدن به توابع پیش فرض زبان، خروجی‌های مخصوص هر کدام به صورت زیر چاپ شود:

Built-In: PUTS

Built-In: LEN

Built-In: PUSH

Built-In: MATCH

Built-In: CHOP

Built-In: CHOMP

- با رسیدن به Loop Statement-ها خروجی‌های مخصوص هر کدام به صورت زیر چاپ شود:

Loop: DO

Loop: FOR

- در صورت مشاهده دستورهای break، break if، next، next if خروجی‌های زیر مخصوص هر کدام چاپ شود:

Control: NEXT

Control: BREAK

- با رسیدن به Lambda function ها خروجی زیر چاپ شود:

Structure: LAMBDA

- با رسیدن به گزاره‌های assignment، خروجی به صورت زیر چاپ شود:

Assignment: "variable name"

- در صورت مشاهده فراخوانی توابع (در هر مکانی از کد) خروجی زیر چاپ شود:

Function Call

## پیمایش عبارت‌ها

- عبارت‌ها بر اساس پیمایش post-order در درخت عبارت، به صورت زیر چاپ شوند:
- در صورت مشاهده هر یک از عملگرهای ذکر شده در documentation زبان، خروجی زیر چاپ شود:

Operator: "operator name"

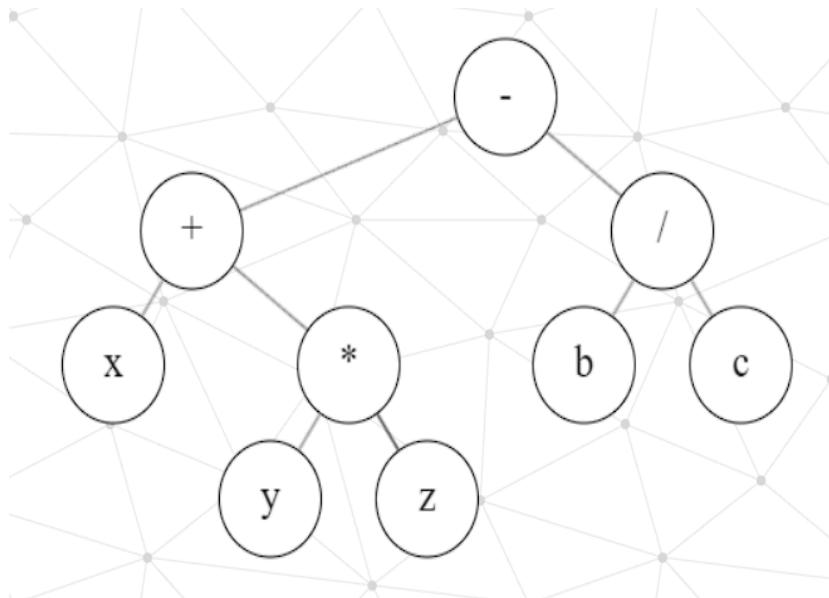
برای مثال، برای عبارت  $x + y * z - b / c$  عملگرها بدین صورت چاپ میشوند :

Operator: \*

Operator: +

Operator: /

Operator: -



دقت کنید که عملگرهای [], (), ., , و ; نباید چاپ شوند.

همچنین دقت کنید که عبارتهای قبل از : همگی کلید واژه هستند و باید آنها را عیناً چاپ کنید. تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خطهای خالی، فاصله و غیره نیز خودداری کنید.

## نکات مهم

- تمامی فایلها و کدهای خود را در یک فایل فشرده به صورت studentID1\_studentID2.zip آپلود نمایید.
- در صورت کشف هرگونه تقلب، نمره صفر لحاظ میشود.
- دقت کنید که خروجیها به صورت خودکار تست میشوند؛ پس نحوه چاپ خروجی باید عیناً مطابق موارد ذکر شده در بالا باشد. علاوه بر آن، درخت parse ساخته شده نیز مورد بررسی قرار میگیرد.
- بهتر است سوالات خود را در فروم درس یا در گروه اسکایپ مطرح نمایید تا دوستانتان نیز از آنها استفاده کنند؛ در غیر این صورت به مسئولان پروژه ایمیل بزنید.