



طراحی کامپایلر و زبان‌های برنامه‌نویسی

تمرین کامپیوتری ۲

دستیاران آموزشی: شایان کاشفی، علی امامزاده

بهار ۱۴۰۲

در این تمرین، شما باید با اضافه کردن action های مناسب به گرامر فاز قبل، درخت AST را ساخته و با استفاده از ویزیتوری که مینویسید node های درخت AST را به فرمت خواسته شده چاپ کنید و خطاهای معنایی را نیز در کد تشخیص بدهید.

1. پس از نوشتن تحلیلگر لغوی و نحوی در فاز قبل، باید به کمک action هایی که به زبان جاوا مینویسید و به گرامر خود اضافه میکنید node های درخت AST را بسازید. تمامی کلاس‌های node های AST در پوشه ast در اختیار شما قرار گرفته است و شما کافی است با استفاده از action ها در گرامر این node ها را بسازید و مقادیر لازم را به آنها بدهید.

2. سپس با استفاده از ویزیتور AstPrinter باید پیام مربوط به هر node را با استفاده از تابع messagePrinter که در کلاس AstPrinter قرار دارد چاپ کنید. لازم است دقت کنید که در اینجا ترتیب پرینت‌های انجام گرفته باید مطابق ترتیب preorder در پیمایش درخت AST باشد. خروجی‌های شما باید مانند فرمت زیر شامل شماره خط (مطابق شماره line های ارائه شده برای هر node در درخت AST) باشد:

```
line <node_line>:<node.toString()>
```

برای مثال برای program خواهیم داشت:

```
line 0:Program
```

3. سپس برای بررسی خطاهای معنایی ابتدا باید symbol table ها را ساخته و کامل کنید. سپس در ویزیتورهای مربوط به آن به بررسی خطاهای معنایی بپردازید. قسمت هایی که شما باید کد آن را بنویسید به صورت TODO مشخص شده‌اند.

لیست خطاها

برای چاپ خطاها کلاس‌هایی در اختیار شما قرار داده شده است که کافی است با رسیدن به هر خطا، کلاس مربوط به آن خطا را به لیست خطاهای ویزیتور مربوط به آن اضافه کنید. فرمت کلی خطاها به صورت زیر است:

```
Line <LineNumber> : <ErrorMessage>
```

مرحله‌ی name analysis (و چاپ خطاهای مربوط به آن):

1. عدم وجود دو متغیر با نام یکسان در یک scope:

در صورت خطا در این قسمت باید خطای مربوطه اعلام گردد و یک نام موقت برای متغیر کنونی در نظر گرفته شود. برای اسم دادن دوباره به متغیر، از الگویی استفاده کنید که برخوردی با نام متغیر از پیش موجود در برنامه، نداشته باشد.

```
ErrorMessage: Redefinition of variable
```

2. عدم وجود دو تابع با نام یکسان در برنامه:

```
ErrorMessage: Redefinition of function
```

توجه کنید همانطور که در صورت زبان توضیح داده شده است، در صورت استفاده از یک نام به عنوان آرگومان predicate، ممکن است در ادامه متغیری به همان نام تعریف شود. ممکن است یک نام به عنوان آرگومان predicate به کار رفته باشد اما متغیری با آن نام در برنامه تعریف نشده باشد.

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 خروجی‌های 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، که روش دوم بدلیل امکان استفاده دانشجویان دیگر از موارد مطرح شده پیشنهاد میشود.
- توجه کنید که در صورت احراز ثقلب 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.
- تمام کد خود را به صورت یک فایل زیپ به نام <sid2>-<sid1>-PLC-Phase2 در بخش قرار داده شده در صفحه درس آپلود کنید
- آپلود فایل نهایی پروژه فقط توسط یکی از اعضای گروه صورت پذیرد.
- موارد خواسته شده در این فاز را بر روی کدی که در سایت برایتان آپلود شده تکمیل کنید.
- برای بخش ویزیتورها کافی است که بخش هایی که ToDo دارند را کامل کنید.
- ابتدا چاپ‌های مربوط به درخت AST انجام میشوند و در ادامه خطاهای معنایی چاپ میشوند.