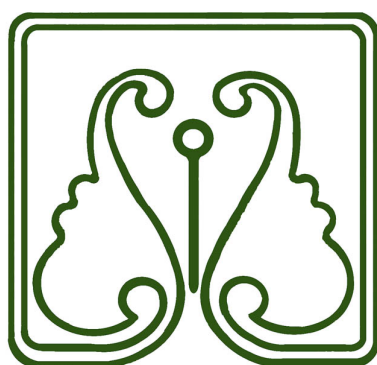


پروژه اصول طراحی کامپایلر: فاز دوم



دانشگاه سهند

نام استاد: دکتر فرید فیضی

ترم ۱۴۰۲۲

در فاز دوم پروژه، شما باید یک تحلیل‌گر نحوی (Parser) بنویسید که ویژگی‌های مورد نظر برای این تحلیل‌گر در ادامه بررسی خواهد شد.

در این مرحله می‌توانید از تحلیل‌گر لغوی‌ای که در فاز قبلی طراحی کردید استفاده کنید و در صورت نیاز در آن تغییرات لازم را برای استفاده در تحلیل‌گر نحوی ایجاد کنید.

نحوه‌هایی که تحلیل‌گر باید آن‌ها را پشتیبانی کند:

- تعریف متغیر با نوع‌های مختلف (int, float, double, string)
- ساختارهای حلقه (for, while)
- ساختارهای شرطی (if, else if, else)
- ساختار توابع
- ساختار class
- عملیات‌های ریاضی: تحلیل‌گر نحوی باید قادر به انجام محاسبات ریاضی با رعایت الویت محاسباتی باشد.

مواردی که در پیاده‌سازی گرامر خود باید رعایت کنید، به شرح زیر است:

۱. برنامه باید دقیقا دارای یک تابع main باشد.
۲. تابع main آرگومان نمی‌پذیرد.
۳. در هر scope (برای ساختارهای شرطی و حلقه) اگر فقط یک گزاره وجود داشته باشد، می‌توان از begin و end استفاده نکرد؛ در غیر این صورت حتما scope باید با این دو کلیدواژه مشخص شود.
۴. هنگام تعریف متغیر، مقداردهی اولیه آن الزامی نیست.

خروجی

پس از نوشتن تحلیل‌گر نحوی، باید به کمک action‌هایی که می‌نویسید خروجی‌های زیر را چاپ کنید:

۱. هنگام رسیدن به هر class باید نام آن را چاپ کنید.

۲. هنگام رسیدن به هر تابع باید موارد زیر را چاپ کنید:

- نوع خروجی تابع
- نام تابع
- نوع و نام پارامترها

۳. با رسیدن به تعریف هر متغیر، باید نوع و نام متغیر را چاپ کنید. همچنین اگر متغیر مقداردهی اولیه شده است، باید آن مقدار را نیز چاپ کنید.

۴. در صورت مشاهده ساختار شرطی کافی است صرفاً عبارتی مانند زیر چاپ کنید:

Conditional: if

Conditional: else if

Conditional: else

۵. در صورت مشاهده ساختار حلقه، کافیست صرفاً نوع آن (while, for) را چاپ کنید.

۶. در صورت رسیدن به یک فراخوانی تابع، نام تابع به همراه مقادیر پاس داده شده به عنوان پارامتر را چاپ کنید. (امتیازی: در صورت عدم وجود تابع یا مغایرت نوع پارامترها، خطا چاپ شود.)

۷. در صورت مشاهده عملیات ریاضی، باید مقدار نهایی آن را چاپ کنید

بخش امتیازی

۱. ساخت درخت AST در action های مربوطه

۲. جلوگیری از تعریف کلاس یا فانکشن با نام های مشابه

۳. جلوگیری از تعریف متغیر با نام های مشابه در یک scope واحد

۴. عدم وجود وابستگی حلقوی: اگر دو کلاس داشته باشیم و در کلاس اول، متغیری از جنس کلاس دوم و در کلاس دوم نیز متغیری از جنس کلاس اول داشته باشیم، وابستگی حلقوی وجود دارد و باید هشدار چاپ شود.

```

1  class MyClass begin
2      int int_var;
3      int int_var2 = 10;
4
5      void some_function(int x, float y, double z) begin
6
7          end
8  end
9
10 void free_function( int a1, int a2) begin
11
12 end
13
14 void main() begin
15     float float_var = 10.0f;
16     int j = 20 + 2 * 3 - 10 + 15 / 5; // = 19
17
18     free_function(5, 10);
19
20     for( int i = 0; i < 10; i++)
21     begin
22         free_function(10, j);
23     end
24
25     int i = 0;
26     while(i < 10) begin
27         print("Hello world!");
28         i++;
29     end
30
31     if (i < 5) begin
32         print("If!");
33     end
34     else if (j < 5) // Without begin/end
35         print("Else if!");
36     else begin
37         print("Else!");
38     end
39 end

```

نکات مهم:

- یک فایل pdf به نام report به عنوان گزارش کار از عملکرد پروژه شامل توضیحات مورد نیاز درباره کد را به همراه تصاویری از مراحل اجرا در فایل فشرده پروژه قرار دهید.
- برای دریافت کامل نمره امتیازی، پیاده سازی تمام بخش های امتیازی الزامی نیست.
- تاریخ و زمان ارائه ی پروژه، متعاقبا اعلام خواهد شد.