



به نام خدا

فاز سوم پروژه کامپایلرها و زبان‌های برنامه‌نویسی

پاییز 1400

مهلت تحویل : 3 دی

در این فاز پروژه، قرار است تحلیل معنایی زبان C-- را که در فاز قبلی آغاز کردید به اتمام برسانید. بدین منظور شما قوانین بررسی نوع را پیاده سازی خواهید کرد. برای تعیین قوانین از اطلاعات بیان شده در ادامه استفاده کنید.

نوع

همانطور که میدانید هر نوع زیرنوع خودش است (رابطه انعکاسی میان نوع‌ها وجود دارد). به عنوان مثال $\text{int} <: \text{int}$ است.

Overloading

در زبان C-- تنها عملگر $==$ و تابع `display` به صورت `overloaded` تعریف شده‌اند. در عبارت $e1 == e2$ باید نوع `e1` و `e2` یکسان باشد. این عملگر روی تمام نوع‌ها بجز لیست تعریف شده‌اند. در فراخوانی `display(e)` عبارت `e` میتواند از نوع `bool` یا `int` یا لیست باشد.

عملگرهای دیگر را طبق جدول زیر بررسی کنید:

عملگرها	نوع عملوند(ها)
ضرب(*)، تقسیم(/)، جمع(+)، تفریق(-)، منفی تک عملوندی(-)، بزرگتر(>)، کوچکتر(<)	int
عطف منطقی(&)، فصل منطقی()، نقیض منطقی(~)	bool

تخصیص

در عبارت $e1 = e2$ باید نوع $e1$ و $e2$ برابر باشد و همچنین $e1$ یک Lvalue باشد.
(Lvalue عبارتی است که به یک مکانی از حافظه اشاره می کند).

لیست خطاها

در این فاز لازم است در صورتی که در بررسی های نام خطایی وجود نداشت، موارد زیر را بررسی کنید و در صورت خطا، پیام مناسب را به فرمتی که برای هر خطا گفته شده چاپ کنید و به بررسی کد تا انتهای فایل ادامه دهید. برای اینکه خطا در یک قسمت روی تمام قسمت های دیگر اثر نگذارد، فرض میکنیم که تمامی متغیرهایی که در نوع خود خطا دارند از یک نوع پیش فرض به نام NoType هستند (NoType زیرنوع تمامی نوع های دیگر است). نوع عباراتی که در نوع خود خطا دارند نیز از همان نوع NoType است.

فرمت کلی هر یک از خطاهایی که باید نمایش دهید به شکل زیر است:

Line <LineNumber>: <ErrorMessage>

مثال:

Line 21: List index is not an integer

که ErrorMessage برای هر یک از موارد زیر تعریف می شود :

1. عدم ارجاع به متغیری که تعریف نشده است

ErrorMessage: Variable <VariableName> is not declared

2. عدم استفاده از نوع ساختمانی که تعریف نشده است

ErrorMessage: Struct <StructName> is not declared

3. وجود متغیر در ساختمان هنگام دسترسی

ErrorMessage: There is no member named <VariableName> in struct <StructName>

4. مجاز بودن نوع هنگام استفاده از عملگرها

در عملگرهای دوتایی، در صورتی که یک عملوند از نوع NoType باشد، این خطا باید برای عملوند دیگر بررسی شود.

ErrorMessage: Unsupported operand type for operator <OperatorName>

5. استفاده از نوع bool در شرط ساختار تصمیم گیری و ساختار تکرار

نوع شرط در عبارت if و while باید از نوع bool باشد.

ErrorMessage: Condition must be bool

6. عدم تخصیص به عملوند rvalue

ErrorMessage: Left side of the assignment must be a valid lvalue

7. قابل فراخوانی بودن در هنگام فراخوانی

فراخوانی باید روی اشاره گر تابع و یا تابع ها صورت گیرد.

ErrorMessage: Calling a non-callable

8. مجاز بودن نوع ورودی تابع display

ErrorMessage: Unsupported type for display

9. عدم استفاده از مقدار بازگشتی تابعی که نوع بازگشتی آن void تعریف شده است

ErrorMessage: Cannot use return value of a function having void return type

10. بررسی نوع مقدار بازگشتی در هنگام استفاده از دستور return با نوع بازگشتی تابع

نوع مقدار بازگشتی باید زیرنوع نوع بازگشتی تابع باشد. نوع بازگشتی تابع getter باید زیرنوع نوع متغیر تعریف شده در ساختمان باشد.

ErrorMessage: Return value does not match with function return type

11. تطابق آرگومان های توابع با تعریف آن ها در هنگام فراخوانی

نوع آرگومان های متناظر باید زیرنوع نوع تعریف شده برای آن در تعریف تابع باشد.

ErrorMessage: Args in function call do not match with definition

12. عدم دسترسی به اعضای متغیری که از نوع ساختمان نباشد

ErrorItemMessage: Access on a non-struct

13. عدم دسترسی به عناصر متغیری که از نوع لیست نباشد

ErrorItemMessage: Access by index on a non-list

14. integer بودن نوع اندیس در هنگام دسترسی به عناصر لیست

ErrorItemMessage: List index is not an integer

15. عدم استفاده از تابع سایز بر متغیری که از نوع لیست نباشد

ErrorItemMessage: Get size of a non-list

16. عدم استفاده از تابع append بر متغیری که از نوع لیست نباشد

ErrorItemMessage: Append to a non-list

17. عدم اضافه کردن عنصر غیر هم نوع با لیست به آن

عنصری که با تابع append به لیست اضافه می شود، باشد زیرنوع اعضای لیست باشد.

ErrorItemMessage: New element type does not match with list type

18. عدم وجود گزاره return در توابع setter و بخش main

ErrorItemMessage: Cannot use return statement in this scope

19. عدم تعریف متغیر جدید در توابع getter و setter

ErrorItemMessage: Cannot define a new variable in this scope

20. وجود گزاره return برای توابعی که نوع بازگشتی آنها غیر از void است (امتیازی)

ErrorItemMessage: Missing return statement for function <FunctionName>

نکته : در صورت وجود بعضی خطاها در یک متغیر، خطاهای مرتبط نباید بررسی شوند، مانند:

- در صورت وجود خطای قسمت 7، خطاهای قسمت های 9 و 11 برای آن فراخوانی بررسی نمیشوند.
- در صورت وجود خطای قسمت 16، خطای قسمت 17 بررسی نمی شود.
- در صورت وجود خطای قسمت 12، خطای قسمت 3 بررسی نمی شود.

همچنین در حالتی که پس از پیمایش کامل کد ورودی هیچ خطایی نداشته باشد، برنامه شما باید خروجی زیر را چاپ کند:

Compilation successful

نکات مهم:

- در این فاز شما باید کدهای داده شده را تکمیل کنید. خطاها و visitor ها به شما داده شده و باید از آنها استفاده کنید.
- تنها دو فایل TypeChecker و ExpressionTypeChecker را به صورت یک فایل P3_<studentID1>_<studentID2>.zip آپلود کنید. توجه شود که تنها یک نفر از هر گروه باید پروژه را آپلود کند؛ در غیر این صورت از شما نمره کسر خواهد شد.
- در صورت کشف هر گونه تقلب، نمره 0 لحاظ میشود.
- دقت کنید که خروجی های شما به صورت خودکار تست میشوند؛ پس نحوه چاپ خروجی شما باید عیناً مطابق موارد ذکر شده در بالا باشد. تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خط خالی و فاصله و ... نیز خودداری کنید.
- بهتر است سوالات خود را در گروه درس مطرح نمایید تا دوستانتان نیز از آنها استفاده کنند؛ در غیر این صورت به مسئولان پروژه ایمیل بزنید:

آرش رسولی arash3908@gmail.com

نازنین یوسفیان nazaninyousefian79@gmail.com