



به نام خدا

فاز سوم پروژه کامپایلرها و زبان‌های برنامه‌نویسی

بهار 1401

مهلت تحویل : 4 خرداد

در این فاز پروژه، قرار است تحلیل معنایی زبان SimpleLOOP را که در فاز قبلی آغاز کردید به اتمام برسانید. بدین منظور شما قوانین بررسی نوع¹ را پیاده سازی خواهید کرد. برای تعیین قوانین بررسی نوع، از اطلاعات بیان شده در ادامه استفاده کنید.

نوع²

همانطور که میدانید هر نوع، زیرنوع³ خودش است (رابطه انعکاسی⁴ میان نوع ها وجود دارد). به عنوان مثال $int <: int$ است. همچنین رابطه‌ی وراثت یک رابطه‌ی زیرنوع میان کلاس‌ها ایجاد میکند. برای مثال اگر B از A ارث‌بری کرده باشد (B فرزند A باشد)، آنگاه $A <: B$ برقرار است.

Overloading

در زبان SimpleLOOP تنها عملگر `==` و تابع `print` به صورت overloaded تعریف شده‌اند.

در عبارت $e1 == e2$ باید نوع `e1` و `e2` یکسان باشد. این عملگر روی تمام نوع‌ها به غیر از آرایه و مجموعه تعریف شده است. در فراخوانی `print(e)` عبارت `e` میتواند از نوع `bool`، `int`، آرایه یا `set` باشد.

¹ Type checking

² Type

³ Subtype

⁴ Reflexive

عملگرهای دیگر را طبق جدول زیر بررسی کنید:

عملگرها	نوع عملوند(ها)
ضرب (*)، تقسیم (/)، جمع (+)، تفریق (-)، منفی تک عملوندی (-)، بزرگتر (>)، تک عملوندی پسوندی (-- و ++)، کوچکتر (<)	int
عطف منطقی (&&)، فصل منطقی ()، نقیض منطقی (!)	bool
عملگر شرطی (ternary)	عملوند اول از نوع bool باشد. نوع دو عملوند دیگر یکسان باشد.

فراخوانی متدها

چنانکه گفته شد، از توابع موجود در زبان SimpleLOOP فقط تابع print به صورت overloaded تعریف شده است. در سایر توابع زبان SimpleLOOP، هیچ overloading یا overriding وجود ندارد. بنابراین هنگام فراخوانی یک متد یک کلاس، آن متد باید در scope آن کلاس تعریف شده باشد. تعداد آرگومان‌های فراخوانی متد باید با تعداد پارامترهای موجود در تعریف متد یکسان باشد. نوع آرگومان‌های فراخوانی متد باید زیرنوعی از پارامترهای موجود در تعریف متد باشد.

تخصیص

تخصیص عبارت $e:T$ فقط به متغیری از نوع S امکان پذیر است که $T < S$ باشد.

self

self همواره نوع کلاسی که در scope آن فراخوانی شده را دارد.

لیست خطاها

در این فاز لازم است در صورتی که در بررسی نام⁵ 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: Array 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

⁵ Name analysis

2. عدم ارجاع به کلاسی که تعریف نشده است:

ErrorItemMessage: Class <ClassName> is not declared

3. وجود فیلد و متد در کلاس هنگام فراخوانی:

ErrorItemMessage: There is no member named <VariableName> in class <ClassName>

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.

ErrorItemMessage: Unsupported operand type for operator <OperatorName>

5. استفاده از نوع bool در شرط ساختار تصمیم گیری

نوع شرط در عبارت if باید از نوع bool باشد.

ErrorItemMessage: Condition must be bool

6. عدم تخصیص به عملوند rvalue

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

7. **lvalue** بودن عملوندهای ++ و - -

ErrorItemMessage: Lvalue is required as the operand of <OperatorName>

8. قابل فراخوانی بودن در هنگام فراخوانی

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

ErrorItemMessage: Calling a non-callable

9. مجاز بودن نوع ورودی تابع **print**

ErrorItemMessage: Unsupported type for print

10. عدم تعریف آرایه خالی

تعریف آرایه به صورتی که طول آن 0 باشد، مجاز نیست (مانند `a[int[0]]`).

ErrorItemMessage: Cannot have empty array

11. عدم استفاده از مقدار بازگشتی متدی که نوع بازگشتی آن **void** تعریف شده است

ErrorItemMessage: Cannot use return value of a method having void return type

12. بررسی نوع مقدار بازگشتی در هنگام استفاده از دستور **return** با نوع بازگشتی متد

نوع مقدار بازگشتی باید زیرنوع نوع بازگشتی تابع باشد.

ErrorItemMessage: Return value does not match with method return type

13. تطابق نوع آرگومان‌های متدها (**initialize**) با نوع تعریف شده

در این قسمت و قسمت بعد، نوع آرگومان‌های متناظر باید زیرنوع نوع تعریف شده برای آن در تعریف تابع باشد.

ErrorItemMessage: Args in method call do not match with definition

14. تطابق نوع آرگومان‌های **initialize** با نوع تعریف شده

ErrorItemMessage: Constructor args do not match with definition of class <ClassName>

15. آرایه بودن نوع پیمایش‌شونده در حلقه **each**

نوع متغیر قبل از **each** باید آرایه باشد.

ErrorItemMessage: Each cannot iterate over a non-array

16. **Integer** بودن مقدارهای داخل **range** در حلقه **each**

ErrorItemMessage: Each range expression is not an integer

17. تطابق نوع متغیر پیمایش کننده با نوع اعضای پیمایش شونده در حلقه **each**

ErrorItemMessage: Each variable <VarName> is not same type as range input

18. عدم دسترسی به اعضای متغیری که از نوع کلاس نباشد

ErrorItemMessage: Access on a non-class

19. عدم دسترسی به عناصر متغیری که از نوع آرایه نباشد

ErrorItemMessage: Access by index on a non-array

20. **integer** بودن نوع اندیس در هنگام دسترسی به عناصر آرایه

ErrorItemMessage: Array index is not an integer

21. **Set** بودن عبارت ورودی **merge**

عبارت ورودی **merge** یا باید از نوع **Set** باشد یا به شکل **comma seperated integers** باشد.

ErrorItemMessage: Merge on non-Set

22. Set بودن عبارت ورودی new

عبارت داخل new باید به شکل comma seperated integers باشد.

ErrorItemMessage: New on non-Set

23. Integer بودن عبارت داخل add

ErrorItemMessage: Set add method input is not an integer

24. Integer بودن عبارت داخل include

ErrorItemMessage: Set include method input is not an integer

25. وجود کلاس Main

ErrorItemMessage: Main class not found

26. عدم ارث‌بری کلاس Main از کلاسی دیگر

ErrorItemMessage: Main class cannot inherit any class

27. عدم ارث‌بری کلاس‌ها از کلاس Main

ErrorItemMessage: Classes cannot extend from Main class

28. وجود initialize در کلاس Main

ErrorItemMessage: No constructor found for Main class

29. عدم تعریف آرگومان برای initialize در کلاس Main

ErrorItemMessage: Main constructor cannot have arguments

30. وجود دستور return در متد با مقدار بازگشتی غیر void (امتیازی)

ErrorItemMessage: Method <MethodName> needs return statement

31. عدم وجود گزاره return در متد با نوع بازگشتی void (امتیازی)

ErrorItemMessage: Method <MethodName> should not have a return

statement

32. عدم وجود گزاره‌های دسترسی ناپذیر (امتیازی)

این خطا برای اولین گزاره‌ی دسترسی ناپذیر چاپ می‌شود. گزاره‌ی دسترسی ناپذیر گزاره‌ای است که بعد از return می‌آید.

ErrorItemMessage: Statements are unreachable

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

- در صورت وجود خطای قسمت 8، خطای قسمت 11 و 13 بررسی نمی شود.
- در صورت وجود خطای قسمت 15، خطای قسمت 16 و 17 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 zip. <studentID2>_<studentID1>_P3_ آپلود کنید. توجه شود که تنها یک نفر از هرگروه باید پروژه را آپلود کند؛ در غیر این صورت از شما نمره کسر خواهد شد.
- در صورت کشف هر گونه تقلب، نمره -100 لحاظ میشود.
- دقت کنید که خروجی ها به صورت خودکار تست میشوند؛ پس نحوه چاپ خروجی باید عیناً مطابق موارد ذکر شده در بالا باشد. تنها موارد خواسته شده را در فایل خروجی نمایش دهید و از قرار دادن خطهای خالی و فاصله و ... نیز خودداری کنید.
- بهتر است سوالات خود را در فروم یا گروه درس مطرح نمایید تا دوستانتان نیز از آنها استفاده کنند؛ در غیر این صورت به مسئولان پروژه ایمیل بزنید:

kianoosharshi@gmail.com

کیانوش عرشی

Amirferyg@gmail.com

امیر فراهانی