



به نام خدا



فاز سوم پروژه کامپایلرها و زبان‌های برنامه‌نویسی

پاییز ۹۹

مهلت تحویل: ۷ دی

در این فاز پروژه، قرار است تحلیل معنایی زبان Sophia را که در فاز قبلی آغاز کردید به اتمام برسانید. بدین منظور شما قوانین بررسی نوع^۱ را پیاده‌سازی خواهید کرد. برای تعیین قوانین از اطلاعات بیان‌شده در ادامه استفاده کنید.

نوع

همانطور که می‌دانید هر نوع زیرنوع^۲ خودش است (رابطه انعکاسی میان نوع‌ها وجود دارد). به عنوان مثال $\text{int} < \text{int}$ 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 $B < A$ برقرار است.

Overloading

در زبان Sophia تنها عملگرهای $=$ و $!=$ و تابع `print` به صورت `overloaded` تعریف شده‌اند. در عبارات $e1 == e2$ و $e1 != e2$ باید نوع `e1` و `e2` یکسان باشد. این دو عملگر روی تمام نوع‌ها بجز لیست تعریف شده‌اند. توجه داشته باشید در صورتی که یک عملوند آنها از نوع کلاس یا اشاره‌گر تابع باشد، عملوند دیگر می‌تواند `null` نیز باشد.

در فراخوانی `print(e)` عبارت `e` می‌تواند از نوع `int`، `string` و یا `bool` باشد.

عملگرهای دیگر را طبق جدول زیر بررسی کنید:

^۱ Type check

^۲ Subtype

نوع عملوند(ها)	عملگرها
int	ضرب (*)، تقسیم (/)، باقی مانده (%)، جمع (+)، تفریق (-)، منفی تک عملوندی (-)، تک عملوندی پیشوندی و پسوندی (-- و ++)، بزرگتر (>)، کوچکتر (<)
bool	عطف منطقی (&&)، فصل منطقی ()، نقیض منطقی (!)

فراخوانی متدها

با توجه به اینکه در زبان Sophia، overloading و overriding برای تعریف متدها نداریم، هنگام فراخوانی یک متد یک کلاس، باید آن کلاس متد فراخوانی شده را داشته باشد و تعداد آرگومان‌های فراخوانی با تعریف یکسان و نوع آنها زیرنوعی از آرگومان‌های تعریف شده باشد.

تخصیص

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

This

this همواره نوع کلاسی که در حوزه‌ی آن فراخوانی شده است را خواهد داشت.

لیست خطاها

در این فاز لازم است در صورتی که در بررسی‌های نام^۳ 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 قسمت‌های 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 نوع‌های 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:22:Main class is not found

که ErrorMessage برای هر یک از موارد زیر تعریف می‌شود:

۱. عدم ارجاع به متغیری که تعریف نشده است

ErrorMessage: Variable <VariableName> is not declared

۲. عدم ارجاع به کلاسی که تعریف نشده است

ErrorMessage: Class <ClassName> is not declared

۳. وجود فیلد و متد کلاس در هنگام فراخوانی

توجه داشته باشید ممکن است فیلد یا متد در خود کلاس موجود نباشد و در اجداد آن موجود باشد.

ErrorMessage: There is no member named <MemberName>
in class <ClassName>

۴. 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>

۵. استفاده از نوع bool در شرط ساختار تصمیم‌گیری و ساختار تکرار

نوع شرط در عبارت if و عبارت وسط for باید از نوع bool باشد.

ErrorMessage: Condition must be bool

۶. عدم تخصیص به عملوند rvalue

ErrorMessage: Left side of assignment must be a valid lvalue

۷. lvalue بودن عملوندهای ++ و - - (چه به صورت پیشوندی چه پسوندی)

به عنوان مثال ++22 قابل قبول نیست.

ErrorMessage: Lvalue required as <OperatorName> operand

۸. قابل فراخوانی بودن در هنگام فراخوانی

فراخوانی باید روی اشاره‌گر تابع و یا متدهای کلاس‌ها صورت گیرد.

ErrorMessage: Calling a non-callable

۹. استفاده از **break** و **continue** داخل حلقه‌ی تکرار

ErrorMessage: Break statement not within loop

یا

ErrorMessage: Continue statement not within loop

۱۰. مجاز بودن نوع ورودی تابع **print**

ErrorMessage: Unsupported type for print

۱۱. عدم تعریف لیست خالی

تعریف لیست به صورتی که طول آن ۰ باشد، مجاز نیست (مانند `list(0 # int)`).

ErrorMessage: Cannot have empty list

۱۲. عدم استفاده از عبارت به جای مقدار ثابت **integer** به عنوان اندیس عناصر لیستی که

نوع همه‌ی اعضای آن باهم یکسان نیست

ErrorMessage: Expression cannot be used as index of multi-type list

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

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

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

نوع مقدار بازگشتی باید زیرنوع نوع بازگشتی تابع باشد.

ErrorMessage: Return value does not match with method return type

۱۵. تطابق آرگومان‌های متدها (بجز **constructor**) با تعریف آنها در هنگام فراخوانی

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

ErrorMessage: Args in method call do not match with definition

۱۶. تطابق آرگومان‌های **constructor** با تعریف آن در هنگام فراخوانی

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

۱۷. تطابق اسم **constructor** با اسم کلاس

ErrorMessage: Constructor name does not match with class name

۱۸. عدم وجود دو شناسه‌ی یکسان برای عناصر لیست

در صورتی که چندین عنصر شناسه‌ی یکسانی داشتند این خطا باید تنها یک بار چاپ شود (به ازای هر لیست این خطا تنها یک بار چاپ می‌شود). توجه داشته باشید در صورتی که عنصری از لیست خود یک لیست باشد این بررسی باید برای آن عنصر هم صورت گیرد (و در صورت وجود خطا، چاپ شود).

ErrorMessage: List elements cannot have the same id

۱۹. لیست بودن نوع پیمایش شونده در حلقه‌ی **foreach**

نوع سمت راست **in** باید از نوع لیست باشد.

ErrorMessage: Foreach cannot iterate over a non-list

۲۰. یکسان بودن نوع عناصر لیست در حلقه‌ی **foreach**

توجه داشته باشید که در هنگام بروز این خطا پیمایش شونده لیستی است که نوع همه اعضای آن باهم یکسان نیست.

ErrorMessage: Foreach list elements do not have the same type

۲۱. یکسان بودن نوع پیمایش کننده با نوع عناصر لیست در حلقه‌ی **foreach**

نوع سمت چپ **in** باید با نوع عناصر لیست سمت راست آن برابر باشد. در صورتی که خطای قسمت قبل رخ داده باشد، باید نوع پیمایش کننده با عنصر اول لیست مقایسه شود.

ErrorMessage: Foreach variable <VariableName> and list elements types do not match

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

ErrorMessage: Access by index on a non-list

۲۳. **integer** بودن نوع اندیس در هنگام دسترسی به عناصر لیست

ErrorMessage: List index is not an integer

۲۴. وجود عنصر لیست با شناسه‌ی استفاده شده

ErrorMessage: There is no element named <Name> in the list

۲۵. وجود کلاس **Main**

ErrorItemMessage: Main class is not found

۲۶. عدم ارث‌بری کلاس **Main** از کلاس دیگر

ErrorItemMessage: Main class cannot extend from any class

۲۷. عدم ارث‌بری کلاس‌ها از کلاس **Main**

ErrorItemMessage: Classes cannot extend from Main class

۲۸. وجود **constructor** در کلاس **Main**

ErrorItemMessage: No constructor found for Main class

۲۹. عدم تعریف آرگومان در **constructor** کلاس **Main**

ErrorItemMessage: Main constructor cannot have arguments

۳۰. عدم دسترسی به اعضای متغیری که از نوع کلاس یا لیست نباشد

ErrorItemMessage: Object or list member access on an expression that is neither a list nor an object

۳۱. وجود گزاره **return** در تابعی که نوع بازگشتی آن **void** نباشد (امتیازی)

ErrorItemMessage: Method <MethodName> needs return statement

۳۲. عدم وجود گزاره‌های دسترس‌ناپذیر (امتیازی)

این خطا برای گزاره‌هایی چاپ می‌شود که در روند اجرای برنامه دسترس‌ناپذیرند که شامل گزاره‌هایی است که در یک حلقه بعد از break و continue می‌آیند و گزاره‌هایی که بعد از دستور return می‌آیند. در صورتی که چند گزاره‌ی دسترس‌ناپذیر پشت سر هم آمده باشند، خطا تنها برای اولین گزاره چاپ می‌شود.

ErrorItemMessage: Statements are unreachable

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

- در صورت وجود خطای قسمت ۸، خطاهای قسمت‌های ۱۳ و ۱۵ برای آن فراخوانی بررسی نمی‌شوند.
- در صورت وجود خطای قسمت ۱۹، خطاهای قسمت‌های ۲۰ و ۲۱ برای آن پیمایش‌شونده بررسی نمی‌شوند.

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 visitorها به شما داده شده و باید از آنها استفاده کنید. کلیه فایل‌ها را به صورت یک فایل **P3_<studentID1>_<studentID2>.zip** آپلود کنید (کدهای جدول علائم و گرامر و AST و visitor را هم در آن قرار دهید). توجه شود که تنها یک نفر از هرگروه باید پروژه را آپلود کند. در صورت عدم رعایت این موارد از شما نمره کسر خواهد شد.
- در صورت کشف هر گونه تقلب، نمره ۱۰۰- لحاظ می‌شود.

■ دقت کنید که خروجی‌های 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 خط‌های خالی و فاصله و ... نیز خودداری کنید. ترتیب چاپ مهم نیست و تنها کافیت تمام خط‌های موجود چاپ شده باشند و خطای اضافی چاپ نشده باشد.

■ در صورتی که قبل از کلاس‌های رفع اشکال سوالی برایتان پیش آمد، می‌توانید زودتر آنها را بپرسید تا ابهامات شما در ابتدای زمان پروژه برطرف شود و به ساعات پایانی موکول نشود. 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:

امیر پورمحمدعلی amir.pma1378@gmail.com

مبینا شاه‌بنده shbmobina@gmail.com