

فاز سوم پروژه طراحی و پیاده‌سازی کامپایلر

گذر^۱ دوم و تحلیل‌گر معنایی

پاییز و زمستان ۹۶

تحویل: ساعت ۲۳:۵۹ شنبه، ۲۵ آذر

در این مرحله از پروژه قرار است تحلیل‌گر معنایی که در فاز قبل نوشته‌اید را تکمیل کنید. در اصل در این فاز قرار است موارد مربوط به قسمت تحلیل‌گر معنایی گذر دوم را پیاده‌سازی کنید. توجه نمایید که اگر خطایی در گذر اول یافت شود کامپایلر نباید وارد گذر دوم شود و بررسی‌های مربوط به گذر دوم انجام نمی‌شود.

در این گذر لازم است موارد زیر را بررسی کنید و در صورت خطا پیام مناسب به همراه شماره‌ی خط را اعلام نمایید و به بررسی کد تا انتهای فایل ادامه دهید:

۱. عدم ارجاع به متغیری که تعریف نشده است: در صورت خطا فرض شود که متغیر وجود دارد و نوع آن NoType است. البته امکان ارجاع به متغیر سراسری اکتور که در ادامه تعریف می‌شود وجود دارد.
۲. بررسی گونه‌ها^۲ هنگام استفاده از عملگرها: در صورت خطا با استفاده از گونه‌ی NoType که در کلاس درس مطرح شده بررسی کد ادامه یابد.
۳. بررسی گونه‌ها در ساختارهای تصمیم‌گیری و تکرار
۴. بررسی وجود اکتوری که به آن پیام فرستاده می‌شود.
۵. بررسی وجود گیرنده و گونه‌های ورودی‌های آن در اکتور مقصد هنگام ارسال پیام (با استفاده از نام اکتور یا کلمه‌ی کلیدی self)^۳
۶. ملاحظات مربوط به توابع پیش‌فرض read و write (بررسی گونه‌های آرگومان‌ها و ...)
۷. عدم تخصیص به عملوند rvalue: در صورت خطا فرض شود که سمت چپ lvalue است و مشکلی در تخصیص وجود ندارد تا بررسی کد ادامه یابد.
۸. عدم استفاده از کلمه‌ی کلیدی sender در گیرنده‌ی init بدون آرگومان

توجه نمایید که لیست بالا برخی از مواردی هست که در سند زبان Atalk بیان شده‌اند و شما باید همه‌ی موارد داخل سند (به جز قسمت امتیازی که در ادامه مطرح می‌شود) را پیاده‌سازی کنید.

^۱ Pass

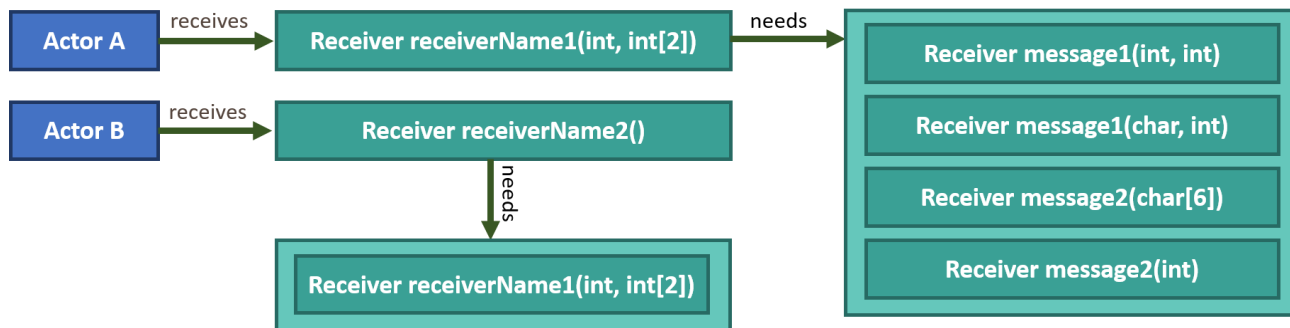
^۲ Type Checking

^۳ برای بررسی گونه‌ها هنگام استفاده از کلمه‌ی کلیدی sender به قسمت امتیازی مراجعه نمایید.

بررسی وجود گیرنده‌های مناسب هنگام ارسال پیام به فرستنده (sender):^۴ برای این موضوع نیاز است تا در گذر اول برای هر گیرنده مجموعه‌ی گیرنده‌های لازم آن مشخص شوند. به مثال زیر توجه کنید:

قطعه کد ۱	
1	actor A<5>
2	receiver receiverName1(int x, int[2] y)
3	sender << message1(x, x + y[0])
4	sender << message1('2', x)
5	sender << message2("123456")
6	sender << message2(x <> y)
7	self << receiverName1(x, y)
8	B << receiverName2("234");
9	end
10	end
11	actor B<4>
12	receiver receiverName2()
13	sender << receiverName1(2, {2, 4})
14	end
15	end

در انتهای گذر اول باید موارد زیر از کد بالا بدست آمده باشند:



در واقع می‌توان تصور کرد که هر گیرنده یک لیست از گیرنده‌هایی دارد که لازم است فرستنده‌ی پیام، آن‌ها را داشته باشد.

سپس در گذر دوم باید با استفاده از اطلاعات بالا و یک الگوریتم ساده در زمان کامپایل وجود گیرنده‌های مناسب هنگام ارسال پیام به فرستنده را بررسی نماییم.

توصیه می‌شود در صورتی که مایل هستید قسمت امتیازی را پیاده‌سازی کنید حتماً با طراحان پروژه مشورت نمایید.

^۴ به قسمت عملگر ارسال (گفتن) در سند زبان مراجعه نمایید.

- تست‌های مورد بررسی در این فاز خطای نحوی^۵ ندارند.
- کدهای خود را به صورت یک فایل studentID1_studentID2.zip آپلود کنید.
- سوالات خود را در فروم درس مطرح نمایید تا دوستانتان نیز از آنها استفاده کنند.
- دقت کنید که خروجی‌های شما به صورت خودکار تست نمی‌شوند.