

# Lab3

---

201250203 张若皓

## 实验思路

---

在上次实验的基础上，通过构建符号表进行处理。

总体结构与老师上课所写的结构相同，但做出了一些改变：

实际上可能会有些差别，因为我首先用的Listener进行实验（结果大失败），所以BaseScope类中会有一些没使用到的方法（都在Listener中使用到），实际上我完成实验是在SymbolTableVisitor中完成。

1. Type类：Type接口，Type类下面根据实验手册进行了分类，分为了BasicTypeSymbol、ArrayType and FunctionType
2. Symbol类：Symbol接口，Class分为FunctionSymbol、BasicTypeSymbol、VariableSymbol以及 BaseSymbol（作为基类，不直接使用）
3. Scope类：Scope接口，Class分为LocalScope、FunctionSymbol（继承了BaseScope类）、GlobalScope和 BaseScope（作为基类，不直接使用）

Symbol类中会有如下属性：Type, lineno, columnno（皆为ArrayList，用来记录定义和使用时行号和列号）

Scope类中会有如下属性：一个Map（用来充当符号表），父亲Scope。

Type类中不同Type会有不同属性，将在精巧构思里阐述。

使用了两个visitor，一个是SymbolTableVisitor，将泛类定义为了Symbol，用来创建符号表，另一个是Lab2中创建的myVisitor（用来输出ParseTree）。

在每一次定义变量和函数时，除了将Symbol加入到当前Scope中的符号表中，还会将其加入到一个总的大符号表（不关心作用域），因为不同作用域的变量会有不同的Symbol对象，其lineno和columnno（在每个Symbol被使用的时候进行添加）也会不同。

剩下的实验思路都非常零碎，简而言之，就是对于Parser中的每一个语句进行结构上的处理，找到能够检查出错误的最小单位进行处理。（比如我其实可以在函数定义的时候就将函数参数加入，但是这样会导致整个的结构非常臃肿）但是例如加法语句，只有在加法语句将左右两边都进行访问，然后再处理。

## 精巧设计

---

在ArrayType中设计了两个int变量，一个为dimension，用来记录这个数组型的维度，另一个为accessDim（用来记录访问（使用）这个数组型的时候的维度）。这样做可以很容易解决多维数组间互相赋值的问题，也可以大大简化判断数组型是否为int型（accessDim为0）。

还有就是在适当时候（例如加法语句）直接返回BasicTypeSymbol可以简化一些繁琐的过程（可能，仔细思考后感觉不太行，反而可能会更麻烦）。

此外，将lineno和columnno（数组）作为属性加入了Symbol的属性中，可以简化重命名的过程。

## 你遇到的困难及解决办法(遇到的奇怪bug)

---

遇到的困难太多了。不知道从何谈起。（挑几个记得住的）

第一个困难是，使用Listener不知道如何传递数据，解决办法是直接Visitor写。

第二个困难是，visitor需要super.visitxxxContext完成对剩下语句的遍历，有时会需要它（不然有些语句无法覆盖到），但是有时不需要（会导致同一个错误遍历两次）。最后每一个语句都慢慢判断。

第三个困难是，修改了gitignore文件（替换成群里的以后），在src内创建文件夹后提交后不提示fail to make，但是全部都是0分。最后还是选择摊在src文件夹中。

第四个困难是，用例太少了，导致很多时候都在空想。根本不知道从哪里开始debug，必须要想到的用例符合oj上的用例才可以。

最后一个奇怪bug，hardtest02会有一个多余输出（导致99）。但是不知道是什么原因，但是问过老师这样的话成绩算满分。所以就不想纠结了。