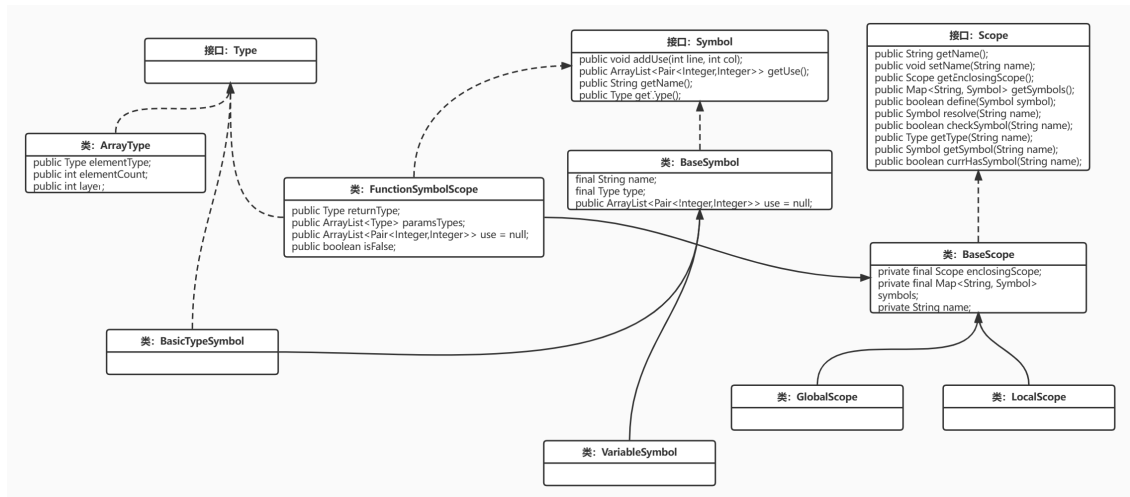


编译原理 lab3 实验报告

李维璇 201250058

实验思路

- 首先设计符号表，类图如下：



设计得比较冗余。FunctionSymbolScope 其实不用实现 Type。BasicTypeSymbol 本来是为了储存数组类型时更方便，但后来发现数组只要存int和维度就行了，所以不必继承 Symbol。感觉也不必将 Scope 作为接口，只用 BaseScope 就可以了。

- 接着，我使用 Listener 来访问每种需要的语法单元。在单个语法单元中，我也对每种情况进行标签标记以示区分。
- 对于重命名，我为每个 Symbol 维护了一个 ArrayList 来储存这个 symbol 在哪行哪列使用过。并且访问途中，如果正好访问到给定的行数列数所对应的变量，就将这个变量所代表的的 Symbol 的 ArrayList 指针保存到 Main 中。这样打印的时候直接看这个 list 就行了。

精巧的设计

- 具体来说说重命名的实现。
- 首先我为每个 Symbol 维护了一个 ArrayList 来储存这个 symbol 在哪行哪列使用过。这个 list 是声明变量时初始化的。以后每次访问到这个变量（无论是 assign 或是作为 lval 或是作为函数的实参），都将行号列号存储在对应的 list 中。
- 当访问到 args 中指定的行号列号所对应的变量时，代表 hit 中了我们需要的变量，就将这个变量的 ArrayList 的指针保存到 Main 中。
- 若没有错，就打印语法树。我使用 Visitor 打印。打印 terminal 时，若这个 token 的行号列号在 Main 中保存的 list 中，就打印新的名字。
- 实现的代码比较短小，且只访问两次语法树，效率较高。

遇到的困难

- 我一度卡在 1213 无法通过，且 normaltest6 只有 26 分。后来询问助教发现我将 array 加在了 globalScope 中，导致声明全局变量 array 时会报 3 错误。这也是一开始没有想清楚符号表的设计导致的。我想单独加一个 array 的 type，但这并不是基础数据类型，不应该加在 globalScope 中。