PA2 实验报告

姓名：杨雅儒 班级：无73 学号：2017011071

**主要工作**：

首先对主要新增代码进行了阅读，然后将PA1-A的部分代码复制进来，之后便开始进行修改。

主要进行的修改如下：

**特性1**

Tree.java：在Visitor类中加入visitScopy空方法，并在Scopy节点上加入accept方法。

BuildSym.java：重载visitScopy方法。

TypeCheck.java：重载visitScopy方法，在其中对于两个参数进行了逻辑判断，并使用了BadScopyArgError和BadScopySrcError。

Parser.y：将之前对于OCStmt的处理进行了更改，IDENTIFIER不仅作为String存储，而新生成一个Ident节点存储。更改了OCStmt传入的loc为SCOPY的loc。

**特性2**

Class.java：增加isSealed布尔型变量，表示是否为sealed类，并新增public方法 getSealed()。

BuildSym.java：在visitTopLevel处新增了对继承sealed的类的处理，使用BadSealedInherError报错。

**特性3**

Tree.java：在Visitor类中加入visitGuardedIf和visitIfSubStmt空方法，并在GuardedIf节点和IfSubStmt节点上加入accept方法

BuildSym.java：重载visitGuardedIf方法和visitIfSubStmt方法，直接使用accept进行递归调用。

TypeCheck.java：重载visitGuardedIf方法和visitIfSubStmt方法，并使用checkTestExpr进行bool类型判定。

**特性4**

BuildSym.java：重载visitIdent方法，判断其isVar为true时进行变量声明，参照visitVarDef方法实现；重载visitAssign方法，通过；重载visitBinary方法。

TypeCheck.java：修改checkBinaryOp方法，使其处理了有任一边为UNKNOWN的情况；修改visitAssign方法，使得左边为UNKNOWN而右边已知时进行处理

BaseType.java：新增UNKNOWN，用于表示未知类型的类型。

Symbol.java：新增setType()方法用于改变Type。

Tree.java：在Expr抽象类中新增isLeft判断是否在赋值语句左侧，若不在则不可以进行var变量定义。

**特性5**

**（1）**

TypeCheck.java：在两个switch语句中都加入对DMOD的处理，使用到了BadArrElementError以及BadNewArrayLength。增加了对TypeCheck.java中的visitIdent的处理，使得在ident为VAR且不在赋值语句左边时不进行向上类型检查和lookup，否则会被之前定义的变量类型所替代。

**（2）**

Tree.java：在Visitor类中加入visitDefaultArrayRef空方法，并在DefaultArrayRef节点上加入accept方法。

BuildSym.java：重载visitDefaultArrayRef方法。

TypeCheck.java：重载visitDefaultArrayRef方法，使用BadArrOperArgError和BadArrIndexError以及BadDefError。增加checkCanBeArrElement(Type type)方法用于检查type能否作为数组元素。

**（3）**

Tree.java：在Visitor类中加入visitForeach空方法，并在Foreach节点上加入accept方法，并加入LocalScope类型的associatedScope变量用以存储作用域;；增加AttachedStmtBlock节点，为方便而使其从Block节点继承，并增加相应常量，以处理作用域问题；

BuildSym.java：重载visitForeach方法;重载visitAttachedStmtBlock方法。

TypeCheck.java：重载visitForeach方法，使用BadArrOperArgError，BadForeachTypeError等，加入break的处理；重载visitAttachedStmtBlock方法。

Parser.y：更改ForeachStmt和BoundVariable中的处理，使得BoundVariable整体作为VarDef进行处理，这也表示需要对应在visitVarDef和visitTypeIdent中加入关于VAR的处理，并且在Tree.java的Foreach类中更改其成员变量为VarDef。加入AttachedStmtBlock，以处理作用域问题。增加AttachedStmt以处理单句的情况。

SemValue.java：新增Tree.AttachedStmtBlock类型的变量attBlock。

**部分调试记录**：

1. 注意在特性4中，由于BuildSym并不会访问到Assign语句，所以更无法访问到Var赋值语句，所以需要将中间缺失的visit都补上。有visitAssign和visitBinary

2. 依然在特性4中，对于q4-5中出现一定问题，就是当var出现在右侧时，似乎不能当作定义变量了，但觉得应该是测例的问题...因为在q4-1中a没有报出未定义的错误...所以最后改为了var在右边时不报出在定义时出现的错误

3. 一定要注意区分兼容和相同，在foreach中会出现这样的问题。

4. 最开始在Parser.y的AttachedStmtBlock中对于不加大括号的情况直接写了Stmt，但是Stmt可以变为StmtBlock，这样将会导致和自己写的带大括号的情况发生冲突进而错误。最终在Parser.y中加了AttachedStmt表示除StmtBlock之外的Stmt。