stage-2

学号：2021012806

姓名：尤梓锐

step5

实验内容

在namer当中，首先增加了对declaration的处理。首先检查在当前ctx环境中能否检索到该变量的定义，如果已经有了，说明重复定义，raise redefine的error。如果没有重复定义，则在该ctx环境中增加该变量的声明，为该declaration创建一个VarSymbol，将该declaration的symbol特征与该VarSymbol绑定。如果有初始化语句，进一步访问初始化语句

```
-        raise NotImplementedError
+        if bool(ctx.lookup(decl.ident.value)):
+            raise DecafRedefinedVarError(decl.ident.value)
+        else:
+            ctx.declare(VarSymbol(decl.ident.value, decl.var_t.type))
+
+        decl.setattr("symbol", ctx.get(decl.ident.value))
+
+        if bool(decl.init_expr):
+            decl.init_expr.accept(self, ctx)
```

```
+class DecafRedefinedVarError(Exception):
+    def __init__(self, name: str) -> None:
+        super().__init__("Semantic error: redefined variable '%s'" % name)
+
```

assignment中，对左右两个语句依次访问即可：

```
def visitAssignment(self, expr: Assignment, ctx: Scope) -> None:
    """
    1. Refer to the implementation of visitBinary.
    """
    raise NotImplementedError
    expr.lhs.accept(self, ctx)
    expr.rhs.accept(self, ctx)
```

最后在identifier中，检查在当前ctx环境中是否有该变量定义，没有的话就raise undefine的 error。如果能找到对应的VarSymbol，就将这个identifier的symbol特征与该VarSymbol绑定。

```
-        raise NotImplementedError
+        if ctx.lookup(ident.value) == None:
+            raise DecafUndefinedVarError(ident.value)
+
+        ident.setattr("symbol", ctx.get(ident.value))
```

在Tacgen中，首先也是增加对declaration的处理。首先为该declaration的symbol声明一个tac层的临时寄存器。如果有初始化语句，在访问该初始化语句后，处理对该变量的赋值，赋的值就是初始化语句的值，从其val特征获取。

```
raise NotImplementedError
decl.getattr("symbol").Temp = mv.freshTemp()
if bool(decl.init_expr):
    decl.init_expr.accept(self, mv)
    mv.visitAssignment(
        decl.getattr("symbol").Temp, decl.init_expr.getattr("val")
    )
```

在assignment中，依次对左右语句进行访问，调用TacFuncEmitter的assignment处理函数，真正实现tac指令的增加。一个赋值语句的值就是它赋的值，因此将该语句的val值设成mv.visitAssignment的返回值，也就是赋的值的临时变量的寄存器。

```
-         raise NotImplementedError
+         expr.rhs.accept(self, mv)
+         expr.lhs.accept(self, mv)
+         expr.setattr(
+             "val", mv.visitAssignment(expr.lhs.getattr("val"), expr.rhs.getattr("val"))
+         )
```

在访问identifier，这个identifier语句的值就是identifier绑定的symbol的对应临时寄存器。

```
raise NotImplementedError
ident.setattr("val", ident.getattr("symbol").Temp)
```

最后在后端增加对赋值tac指令的处理，在riscv中就是mv指令：

```
def visitAssign(self, instr: Assign) -> None:
    self.seq.append(Riscv.Move(instr.dst, instr.src))
```

思考题

栈帧

addi sp, sp, -16

同名变量

在ctx中，一个ident.value应该对应一个symbol的栈。

定义变量：在declaration的访问中，如果有初始化语句先访问初始化语句。再声明变量，就在ctx字典中ident.value对应的symbol栈中压入一个symbol，declaration语句与该symbol绑定。

查找变量：在identifier的访问时，这个identifier绑定的应该是ctx字典中ident.value对应的symbol栈的栈顶symbol。

（不应该直接覆盖老的symbol，因为tacgen当中，还要用到老的symbol，在namer处理的阶段都要存下来，在tacgen中对应语句直接调用自己绑定的symbol即可）