编译原理: PA2

娄晨耀, 2016011343

1 类的浅复制的支持

scopy 最复杂的地方是理解 README 里面报错的顺序。否则的话很难写出和样例一样的报错。

2 sealed 的支持

在 BuildSym 里面的 visitTopLevel，当扫描完全部类的结构后再循环一遍全部的类，检查一下一个类的父亲是不是 sealed。

3 支持串行条件卫士

一个技巧是利用 checkTestExpr 这个函数来直接检查条件是否满足。没有其他难点。

4 支持简单的类型推导

我把 var x 来当做 Ident 来处理了。所以在 BuildSym 的时候 var x = 233; 这样的语句实际上是不会被访问的，需要注意访问一下 Assign.left。然后判断如果是 var 就把 VarDef 的逻辑抄过来就好。

在 checkType 的时候，由 visitAssign 来决定 var 定义的变量类型。visitIdent 只是简单的设置成 UNKNOW。这样对于 a = var b + c; 这样的句子就会自动出现类型不匹配的错误了。

5 数组操作

5.1 数组初始化常量表达式

这里我之前没有用 Binary 来实现初始化操作，但是发现其实 Binary 的报错可能就够用了。于是改写了之前的实现，改成 Binary 里面新加一个 REPEAT 操作。

5.2 数组下标动态访问表达式

其实我这里实现的时候是直接改的 Tree 的 Indexed，所以可以利用 visitIndexed 原有的代码。

遇到一个问题是一模一样的问题定义了不同的错误类型，怀疑是祖传代码里面助教实现的时候新建了一个 Default 类，在判断错误的时候没有注意就多加了两个错误类。

5.3 数组迭代语句

这里遇到的主要问题是需要理解 Scope 是如何实现以及如何利用。因为 Block 会新开一个 Scope，所以我按照 README 上的做法直接把变量开到 Block 的 Scope 里面了。

需要注意一个问题是如果在 visitForeach 里面初始化了 Block 的 associatedScope，那么 accept Block 的时候就不要再初始化一遍。