版本配置:

flex 2.6.4 && Bison 3.0.4

运行指令:

```
make
./proj3 < 输入文件 > 输出文件
```

我的输出文件为.myout结尾,标准答案为.out结尾,可以进行比对。 不同的点如下:

1.当classbody为空时,我的结果与标准答案分别如下图的左/右:



注意到语法树的 [STNode,10,"hahah"] 下面一行略有不同,少了两行 [BodyOp]、[DUMMYnode] 。

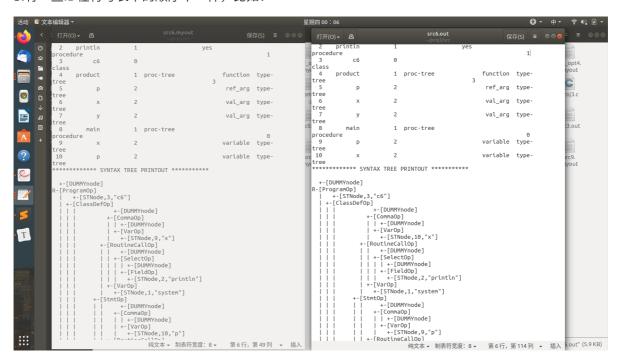
2.当一个类内没有方法时,我的输出(左)与答案(右)相比会多出两行:

```
星期四 11:34
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           ⊕ • 中• 후 📢 🗹
                                                                                                                                  +-[INTEGERTNode]
--[ArrayTypeOp]
| +-[NUMNode,10]
+-[BoundOp]
+-[DUMMYnode]
                                                                                                                                                                                                                                                                                                                                                                                                                                                                           +-IUMMYNOGE|

BodyOp|
+-[ArrayTypeOp]
| +-[ArrayTypeOp]
| +-[RUMNode,10]
| +-[BOUNDOp]
| +-[DUMWYNOde]
+-[CommaOp]
| +-[UMWYNOde]
| +-[IndexOp]
| +-[IndexOp]
| +-[TypeIdOp]
| +-[TypeIdOp]
| +-[CommaOp]
| +-[STNOde,5,"x"]
+-[DUMYNOde]
+-[STNOde,5,"x"]
+-[DUMYNODE]
<u>و</u> م
                                                                                              | +-[DUMYnode]
+-[comma0p]
| +-[DUMYnode]
| +-[Indexop]
| | | +-[Indexop]
| +-[Type1dop]
| +-[Dexlop]
 0
 | | -- [DUMWYnode]
+- [Bady0a]
+- [DUMWYnode]
(Classop]
+- [Classefap]
+- [Classefap]
| -- [Lassefap]
| -- [Lassefap]
| -- [Lassefap]
| -- [DUMWYnode]
| -- [Comma0p]
| -- [LUMWYnode]
| -- [LUMWYnode]
| -- [Comma0p]
| -- [LUMWYnode]
 ?
                                                                                                                                                                                                                                                                                                                                                                                                                                                                     . . [Declop]
| | +-[DUMMYnode]
+-[BodyOp]
+-[DUMMYnode]
>_
                                                                                                                                                                                                                                                                                                                                                                                                                                            ClassOpl
                                                                                                                                                                                                                                                                                                                                                                                                                                               T
                                                                                            | +-[Type1dop]
| +-[INTEGERTNode]
| +-[CommaOp]
| +-[STNode, 3, "length"]
| +-[DeclOp]
| | +-[F.Node, 3]
| | | +-[CommaOp]
| | | +-[DUMWYnode]
| | | +-[Type1dop]
| | | +-[Type1dop]
| | | +-[Type1dop]
| | | +-[STNode, 2, "width"]
| | +-[CommaOp]
| | | +-[STNode, 2, "width"]
| | +-[DUMMYrode]
+-[DUMMYrode]
+-[DUMMYrode]
                                                                                                                                                                                                                                                                                                                                                                                                                                       | | +-[IypeIdOp]
| | +-[CommaOp]
| +-[CommaOp]
| | +-[STNode,2,"length"]
| +-[DeclOp]
| | +-[CommaOp]
| | | +-[CommaOp]
| | | | +-[CommaOp]
| | | | +-[TypeIdOp]
| | | | +-[STNode,3,"width"]
| | | +-[CommaOp]
| | | +-[DeclOp]
| | | +-[DelOp]
| | | +-[DUMWYnode]
| +-[BoUMWYnode]
| +-[BUMWYnode]
| +-[CommaOp]
| +-[DUMWYnode]
                                                                     -[ClassOp]
+-[DUMMYnode]
                                                                                                                                                                                                                    纯文本 - 制表符宽度: 8 - 第82行, 第28列 -
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            纯文本 ▼ 制表符宽度: 8 ▼ 第91行, 第21列 ▼ 插入
```

可以看到在截图中间位置(在 [NumNode,7] 往上两行)我的多出了两行 [DUMMYNode]、[BodyOp]。

3.有一些ID在符号表中的顺序不一样,比如:



注意到最后两个符号x、p顺序相反了,因此在语法树上 STNode 节点对应的索引值也会反过来,但标准 答案与我的输出都会保证语法树上的索引始终与符号表的索引保持一致。因此这并不算是错误。