11.19talk

1. program：遍历declarations，其中有type和id，利用type的不同：int调用var-declaration ,void 调用fun-declaration
2. Var-declaration: type_specifier 一定为int，对每个ID声明分配空间，同时全局变量声明表；分配空间时是数组还是int（通过num是否大于零）
3. Fun-declaration :
    1. id函数名，利用id以及params的个数生成函数入口，再调用compound-stmt；
    2. params遍历每个param，调用param
    3.
4. compound-stmt:
    1. 调用local-declarations：type一定是int型
    2. Statement-list ,利用ppt上的，遍历每个statement，程序自己知道自己的类型
5. Expression-stmt: 如上利用ppt的，遍历每个expression，
6. expression：利用结构体里的type类型，分别调用assign还有simple-stmt
7. Compound-stmt
8. Selection-stmt: 插入branch即icmp与0，trueBB，根局有没有else_stmt决定是否插入false_BB
9. 在assign-expression和simple-expression定义全局变量，便于在selection-stmt的judgeBB使用**
10. Iteration-stmt与selection-stmt相似
11. return-stmt：CreateRet ，expression值利用全局变量
12. expression:
    1. var=expression：先调用var，应该是将expression全局变量值存入声明时variable的存储空间，store进去
    2. simple-expression：调用addictive-expression并用switch case区分relop
13. var：寻找id的声明时的存储空间，利用全局变量存储，主要用于数组进行区分
14. Simple-expression: 调用addictive-expression
15. addictive-expression：根据addictive-exp域决定是否调用自身，switchcase ，再调用term
16. term：根据term域决定是否调用自己，switchcase，调用factor
17. factor：若左括号调用expression，否则按照PPT上的accept this
18. num:
19. call： args先遍历每个expression，获得值，按顺序CreateCall传入参数，

```
assign_value
expression_value
siple_vallue
addictive_value
term_value
```