

# lab5实验报告

## 1.程序功能

利用java的Illum api实现翻译函数和局部变量。

## 2.实现过程

1.沿用lab3的大框架，利用visitor构建符号表

2.本次符号表采用Map<String,LLVMValueRef>建立的是符号名到ref的映射关系。

3.翻译函数：

首先需要生成函数的返回值类型，只需要使用i32Type以及voidType。

接着生成参数类型，然后利用AddFunction函数生成创建函数的指令。

之后为函数内部的局部变量，创建指针分配空间，把参数的值store在该地址中。

4.局部变量：

局部变量在Decl语句中声明，在此处添加进符号表并且为其分配空间。

对于局部变量的使用，利用递归函数getExpRef来获取任意一个exp类型的LLVMValueRef类型的引用，以此为核心来处理其他语句，包括赋值、运算、函数调用等等。该函数通过instanceof来判别exp的派生类型，递归结束条件是const类型的引用或者lval类型的引用，而lval类型应该利用LLVMBuildLoad取出ref对应的val，然后返回值类型的LLVMValueRef。

## 3.遇到过的bug&处理

1.最难理解的部分是数组的部分，这里需要调用LLVMBuildGEP方法，它接受的参数是builder，array的引用，indices（ref的数组），indices的大小，string。而要想确定一维数组中第i个元素的地址，indice就应该是[zero,ConstInt(i)]这样的形式。

2.带参数的函数：这里需要为局部变量分配空间，然后利用LLVMGetParam方法取得函数的参数，并且store到该空间，这样定义出来的函数才可以被正常调用

3.void的类型的函数也需要return void，否则会报错缺少操作符。而且，该类型的函数调用是不能够被命名的，即最后一个String参数应为”，否则会报错。