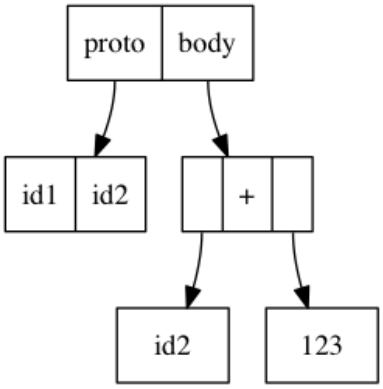
1.解释include/ast.h中ExprAST里的virtual的作用，在继承时的原理（解释vtable）

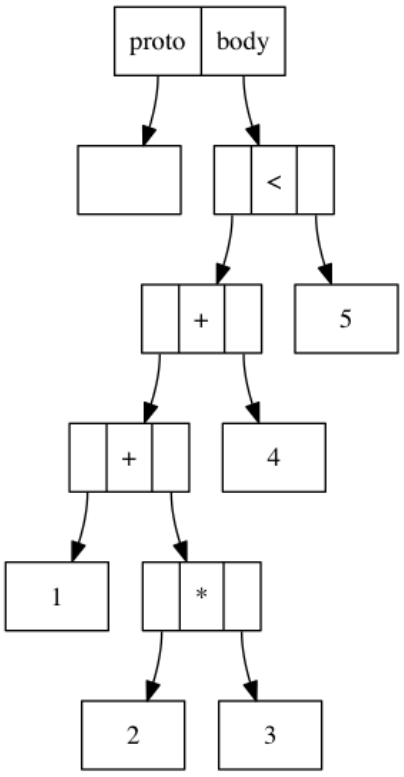
虚函数是指一个类中你希望重载的成员函数，当你用一个基类指针或引用指向一个继承类对象的时候，你调用一个虚函数，实际调用的是继承类的版本。vtable是指一张函数指针表，如同C++中类的实现一样，vtable中的指针指向一个对象支持的接口成员函数。 每个虚函数都在vtable中占了一个表项，保存着一条跳转到它的入口地址的指令（实际上就是保存了它的入口地址）。当一个包含虚函数的对象（注意，不是对象的指针）被创建的时候，它在头部附加一个指针，指向vtable中相应位置。调用虚函数的时候，不管你是用什么指针调用的，它先根据vtable找到入口地址再执行，从而实现了“动态联编”。而不像普通函数那样简单地跳转到一个固定地址。

2.阅读HandleDefinition和HandleTopLevelExpression，忽略Codegen部分，说明两者对应的AST结构



Def的AST:
就是def关键字 + Prototype + Expr，如左图

ToplevelExpression是一个综合性的表达式，既可以是if又可以是while以及算术表达式，它的AST可以是多种树。



3.Kaleidoscope如何在Lexer和Parser间传递信息？（token、语义值、用什么函数和变量）

- 用NumVal把词法分析得到的数字传递给Parser

```
static ExprAST *ParseNumberExpr() {
  ExprAST *Result = new NumberExprAST(NumVal);
  getNextToken(); // consume the number
  return Result;
}
```

- 用IdentifierStr传递标识符

```
std::string IdName = IdentifierStr;
return new VariableExprAST(IdName);
```

- 用tok_xx传递token类型

4.Kaleidoscope如何处理算符优先级？（重点解释ParseBinOpRHS）分析a*b*c、a*b+c、a+b*c分别是如何处理的？

使用算符优先级的方法。在读取新的运算符时，关键在于比较运算符的优先级，然后根据优先级来判断哪个和哪个合成一个新的树。

如a*b*c就先构造a的左子树，发现下一个 * 优先级并不高，就把a*b合并为左子树（其中a、b分别为左右子树），再把c作为右子树。

对于a*b+c,a*b作为左子树，c作为右子树

对于a+b*c，a作为左子树，b*c作为右子树。

5.解释Error, ErrorP, ErrorF的作用，举例说明它们在语法分析中应用

这三个函数的作用都是以字符串输出报告错误。

Error：

static ExprAST *ParseIdentifierExpr()函数被调用时期望当前标记为“(”，但完成子表达式解析之后，后续的标记可能并不是“)”。例如，如果用户输入的是“(4 x”而不是“(4)”，解析器就应该报错。由于可能发生错误，解析器需要一种用于标识发生了错误的方法：我们的解析器在碰到错误时会返回NULL。

ErrorP：

在识别PrototypeAST时检查错误

```
static PrototypeAST *ParsePrototype() {
    if (CurTok != tok_identifier)
```

```
            return ErrorP("Expected function name in
            prototype");
            ...
        }
```

ErrorF：

在Function *PrototypeAST::Codegen( )用到，用来提示关于函数重定义的错误

6.Kaleidoscope不支持声明变量，给变量赋值，那么变量的作用是什么？

变量可以起到指代某个数字以简化代码的作用。比如fib（n），如果不使用变量，就必须定义fib(1),fib(2),fib(3)=fib(1)+fib(2).....使用变量以后，只需一句fib(n)=fib(n-1)+fib(n-2)即可递归表示所有的fib(n)。再例如计算一个很成熟的计算表达式，用函数计算的话，输入数字就能输出结果，略去了每次都要输入长长的计算式的麻烦，而这个函数就必须依赖变量的存在。

7.为什么不需要{...}或begin...end？

Kaleidoscope的语法较为简单，每一种语法都给出了精确定义，它几乎不存在语言的二义性，例如if then如果不写else那就会立即报错，while和for后面也只有单行expression，不存在哪两行expression需要被花括号包围。

8.Kaleidoscope是如何避免if/then/else语句的二义性的？

用简单粗暴的方法，强制要求每个if都要写一个then、else，一一对应来避免二义性

9.Kaleidoscope只有一种数据类型double，那么if/then/else的分支条件是如何实现的？（你可能要看BinaryExpr和IfExpr的Codegen部分）

使用`Builder.CreateUIToFP`函数来将`bool`值转化为了`0.0`或`1.0`