drawSynataxTree函数分析：

函数的依赖：

首先在函数中引进python的graphviz库中的Digraph类，从base.py中引进已经按照列表形式给出的grammar\_tree。

函数的实现：

创建一个Diagraph对象dot，图名设置为Grammar Tree。

创建节点计数器cnt，并初始化为0。

遍历列表形式的grammar\_tree中的每一个元素：

按照当前节点计数器的值，命名一个新节点。

将新节点加入dot中。若节点是终结符，则将节点内容设置为该终结符；否则将节点内容设置为这个节点的类别。

遍历这个节点的所有儿子节点：

命名儿子节点。

在当前节点与儿子节点之间建边，加入dot。

节点计数器自增。

生成png图片，展示语法树。

函数说明：

1. 在命名儿子节点的时候，是如何确定儿子节点的节点名的？

答：事实上，在列表形式的grammar\_tree中，我们可以通过下标定位每一个元素，也就是说所有的节点实际上天然已经有了一个序号，可以用字符串”node”加上这个下标来组成节点的名字，我们在命名新节点的时候，正是使用了这种方法。而在串联儿子节点的时候，由于在grammar\_tree中，储存儿子节点使用的记号是儿子节点的下标，因此只需要取出儿子节点这个元素的内容，就可以得到下标，所以可以直接定位到儿子节点。

1. 第三方库graphviz需要调用graphviz程序，打包时如何处理？

答：你写一点。

调试分析：

针对作业要求给出的类C语言语法，我们测试了5类14种不同的语法错误，均得到了预期输出：

1. 括号类
2. If语句类
3. Return语句类
4. 赋值语句类
5. 程序声明类