

# 第二次实验报告

201250044 王星云

## 一、程序实现功能、实现方式、精巧设计

### (一) 程序实现功能

程序对使用SysY语言书写的源代码进行语法分析与高亮，当输入文件存在语法错误时按要求输入所有的错误，当输入文件不存在语法错误时按要求输入语法树（含高亮）。

### (二) 实现方式

#### 1. 生成语法分析器

1. 通过在SysYParser.g4文件中，根据SysY语言定义中的语法规则，使用正则表达式匹配语法，生成语法分析器
2. 在Main.java接收一个参数，即文件路径，并将文件内容传给词法分析器
3. 从词法分析器获得token，生成语法分析器

#### 2. 报告错误

1. 去除语法分析器原先的所有ErrorListener
2. 实现一个MyErrorListener,自定义syntaxError方法
3. 将MyErrorListener加入语法分析器

#### 3. 打印语法树

1. 实现一个MyVistor
2. 重写visitChildren函数和visitTerminal函数，分别打印语法树节点和终结符
3. 这两个函数获得的参数是一个node节点，通过节点的上下文(ruleContext)获得语法规则的编号，再通过SysYParser的静态成员ruleNames可以获得所有语法规则的名称，下标与编号对应
4. 输出语法规则名称前将首字母大写

#### 4. 实现高亮

1. 从SysYLexer的ruleNames获得所有语法规则名称，创建一个字符串数组，通过表驱动的方式获取每个终结符的颜色并输出

### (三) 精巧设计

1. 使用表驱动的方式获取每种终结符的颜色，避免对每种终结符进行分别处理
2. 调用node节点的上下文的depth函数，来获取当前节点的深度，以实现缩进，避免了父节点循环查找的方式

3. MyErrorListener的实现是单例模式，有err字段，用以判断是否有语法错误，如果err为true，则不输入语法树

## 二、有趣的现象和印象深刻的bug

### （一）有趣的现象

Antlr能够自己处理左递归和优先级关系（通过先后顺序，越靠前的优先级越高），因此可以将SysY语言定义中的语法定义改写为比较简明的左递归形式。

### （二）印象深刻的bug

一开始我对INTEGR\_CONST的内容是直接输出的，后来发现仍然需要参考lab1，将十六进制、八进制转为十进制输出才可以通过OJ。