

Lab1实验报告

201250141 刘屿

实验思路

1. 首先根据SysY词法规则中的词法规则编写 `SysYLexer.g4`，然后为其生成词法分析器 `SysYLexer.java`

```
1 $ make compile
```

2. `main`方法接收文件路径，并将文件内容传给词法分析器

```
1 if (args.length < 1) {
2     System.err.println("input path is required");
3 }
4 String source = args[0];
5 CharStream input = CharStreams.fromFileName(source);
6 SysYLexer sysYLexer = new SysYLexer(input);
```

3. 实现一个继承自`BaseErrorListener`的`MyErrorListener`并添加给`SysYLexer`
 - 之后遇到错误时会向`MyErrorListener`发送错误信息，调用 `syntaxError()` 方法，因此只需要重写 `syntaxError()` 方法，按要求的格式输出错误信息即可
 - 因为需要输出全部错误信息，所以在`MyErrorListener`类中定义一个`boolean`成员变量 `used`，用来判断是否发生了词法错误
 - 通过在 `syntaxError()` 方法中将`used`设为`true`实现

```
1 sysYLexer.removeErrorListeners();
2 MyErrorListener myErrorListener = new MyErrorListener();
3 sysYLexer.addErrorListener(myErrorListener);
```

```
1 public class MyErrorListener extends BaseErrorListener {
2
3     public boolean used = false;
4     @Override
5     public void syntaxError(Recognizer, ? recognizer, Object
6         offendingSymbol, int line, int charPositionInLine, String msg,
7         RecognitionException e) {
8         System.err.println("Error type A at Line " + line + ":"
9             + msg + ".");
10        used = true;
11    }
12 }
```

4. 通过 `sysYLexer.getAllTokens()` 函数触发sysYLexer的错误检查并获得所有 token

```
1 List<? extends Token> allTokens = sysYLexer.getAllTokens();
```

5. 如果 `myErrorListener.used` 为 false, 即没有发生词法错误, 打印正常情况 token 内容, 通过 `sysYLexer.getRuleNames()` 获取所有.g4文件中定义的规则

```
1 if (! myErrorListener.used) {
2     // 没有词法错误
3     String[] ruleNames = sysYLexer.getRuleNames();
4     ...
5 }
```

6. 遍历所有的 token, 通过 `ruleNames[token.getType() - 1]` 获得Token类型
- 此处 -1 是因为SysYLexer中的RuleNames下标是从 0 开始的, 而 `token.getType()` 从 1 开始

```
1 for (Token token : allTokens) {
2     String ruleName = ruleNames[token.getType() - 1];
3     ...
4 }
```

7. 在遇到 `INTEGR_CONST` 时需要将 8 进制与 16 进制转化为 10 进制, 通过编写静态函数实现, 之后按要求输出即可

```
1 if (ruleName.equals("INTEGR_CONST")) {
2     Integer number = parseInt(token.getText());
3     System.err.println(ruleName + " " + number + " at Line " +
4         token.getLine() + ".");
5 } else {
6     System.err.println(ruleName + " " + token.getText() + " at
7     Line " + token.getLine() + ".");
8 }
```

```
1 public static Integer parseInt(String text) {
2     if (text.startsWith("0x") || text.startsWith("0X")) {
3         return Integer.parseInt(text.substring(2), 16);
4     } else if (text.startsWith("0") && text.length() > 1) {
5         return Integer.parseInt(text.substring(1), 8);
6     }
7     return Integer.parseInt(text);
8 }
```

精巧的设计

- 通过在MyErrorListener中添加成员变量来判断输入文件是否有词法错误
- 使用`Integer.parseInt(s, radix)`来快速进行 8 进制与 16 进制到 10 进制的转化
 - 先判断16进制的'0x'开头，然后再判断8进制的'0'开头，来区分这两种进制，并且8进制字符串长度要大于1

遇到的困难及解决办法

问题主要集中在main函数调用完`sysYLexer.getAllTokens()`后不知道有没有出现语法错误，SysYLexer没有一个函数或者成员变量可以获取到这一点。

如果不做处理的话，输入文本全部正确的的情况下没有问题，一旦有词法错误就会打印完错误信息后继续打印全部的token信息

一开始我选择在`MyErrorListener.syntaxError()`方法结尾抛出运行时异常，在main方法中捕捉，但这样只能解决只有一个词法错误的情况

随后我发现通过在MyErrorListener类中添加成员变量used即可完美解决这个问题，如果调用了`syntaxError()`方法，就将used设为true，因为MyErrorListener是在main中创建，因此可以直接获取到used的值