

1. 程序实现功能

通过antlr对SysY语言书写的源代码进行词法分析，并打印分析结果。

2. 如何实现

1. 根据SysY词法规则编写SysYLexer.g4，然后利用antlr为其生成词法分析器，即SysYLexer.java文件。
2. Main.java接收一个文件路径的参数，然后将文件内容传给词法分析器。
3. 实现一个继承自BaseErrorListener的ErrorListener并添加给sysYLexer，并override BASEErrorListener的syntaxError方法，使词法分析器遇到错误时，打印出错的行号。
4. 通过sysYLexer的getAllTokens()函数触发sysYLexer的错误检查并获得所有token。
 - 如果出错，则自动调用ErrorListener的syntaxError方法，打印所有错误信息。
 - 如果词法全部正确，则通过以下方法获取token的值、行号和类型，按要求输出。其中还要注意将八进制和十六进制数字常量转为十进制输出。

```
String[] ruleNames = sysYLexer.getRuleNames();
if(!isWrong) {
    for (Token token : tokens) {
        String text = token.getText();
        String line = String.valueOf(token.getLine());
        String type = ruleNames[token.getType() - 1];
```

3. 精巧设计

我使用一个boolean类型的全局变量isWrong来存储分析结果是否出错，初始值为false。如果词法分析器遇到错误，isWrong置为true，这样就不会输出识别到的正确Tokens的信息了。

4. 印象深刻的bug

4.1

我一开始在本地使用的antlr版本是4.10.1，make compile之后生成的是sysYLexerLexer.java，但是我每次提交oj都是0分；然后我本地antlr换到4.9.1，但是本地运行又报错，于是我又换回4.10.1之后再make compile生成的是sysYLexer.java，提交上去就正常了。现在还是不知道为什么会这样😭

4.2

```
// 数字常量，包含十进制数，0开头的八进制数，0x或0X开头的十六进制数
INTEGR_CONST : ('0' | ([1-9] DIGIT*)) | ('0' [0-7]*) | (('0x' | '0X') DIGIT*);
```

十六进制数一开始是这么写的，忘记还有a-f和A-F了😭

