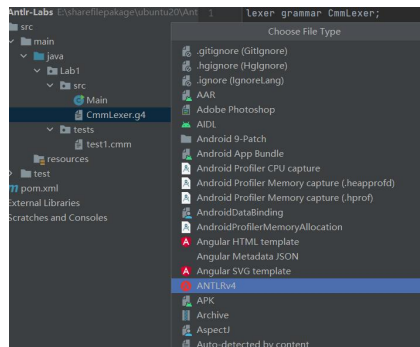


实验报告

1. 关于 idea 的使用

注意：右击.g4 文件，选择 **override file type** 将该文件设置成 **antlrv4** 形式



2. java 基础语法

● 类型转换

String -> Double	<code>double dou_num = Double.parseDouble(value);</code>
String -> Decimal(Long)	<code>long dec_num = Long.parseLong(value, radix);</code>

● 内嵌引用类

因为要在内嵌的引用类的 **override** 方法中修改 **hasMistake**，所以 **hasMistake** 作为 **class Main** 的类变量、修改时用 **Main.hasMistake** 引用。

3. Antlr v4 Lexer 部分

● 资源

官方学习原理和实例

[antlr4/index.md at master · antlr/antlr4 · GitHub](#)

语法文件实例

<https://github.com/antlr/grammars-v4>

● 逻辑

Antlr 是基于某种编程语言（例如 java，python 等）的语言解析器，可以完成词法分析、语法分析、语义分析等任务。Antlr 构建词法解析器的过程如下：

- 1. 构建.g4 文件（语言描述）
- 2. Antlr 为该语言生成 parser（解析器）
- 3. 该解析器可以自动构建解析树（语法如何匹配输入的数据结构）

该解析器也自动生成树行走器，可以使用其访问解析树的节点，以执行特定于应用程序的代码。

● 常用接口

获取待解析内容	CharStream in = CharStreams.fromString(StringName)/.fromFileName(FileName);
创建解析器 lexer	CmmLexer lexer = new CmmLexer(in);
得到符号集合 tokenList， token 形如 [@-1,0:4='hello',<30>,1:0]	List<? extends Token> tokenList = lexer.getAllTokens();
得到 token 中的所需部分，.getText()得到“hello”	int temp = tokenList.get(i).getType()/.getLine()/.getText();
根据 type 编号得到 type 名称	String type = CmmLexer.VOCABULARY.getSymbolicName(temp);
消除 getText()最后的换行	String text = text.trim();