**西北工业大学**

**《编译原理》语法分析实验**

|  |  |
| --- | --- |
| 学 院： | 软件学院 |
| 学　　号： | 2018303081 |
| 姓　　名： | 马泽红 |
| 专 业： | 软件工程 |

西北工业大学

**2021 年 4 月**

目录

[1.代码介绍 2](#_Toc71757626)

[1.1 代码目录介绍 2](#_Toc71757627)

[1.2 代码运行方式 2](#_Toc71757628)

[2.自定义测试方案 3](#_Toc71757629)

[2.1 struct\_type测试 3](#_Toc71757630)

[2.2 数据类型测试 5](#_Toc71757631)

[2.3 其他表达式综合测试 6](#_Toc71757632)

[2.4 错误测试 7](#_Toc71757633)

# 1.代码介绍

此次语法分析实验的代码使用**递归下降算法**实现，在各匹配函数中完成各语法规则的匹配与抽象语法树的构建。

## 1.1 代码目录介绍

MIDL\_compiler

└─src //程序源代码

├─inputFile //输入的测试文件

│ └─语法测试文件

│ ├─base\_type\_spec

│ ├─declarators

│ ├─expr

│ ├─member\_list

│ ├─struct\_type

│ ├─type\_spec

│ └─自定义测试文件 //自己的测试方案存储的位置

│ CompilerTest.java //程序入口，输入源程序，输出语法分析结果

├─outputFile

│ ├─syntaxAnalysisOutput //语法分析结果存储位置

│ └─tokenAnalysisOutput //词法分析结果存储位置

├─syntaxAnalysis //语法分析的源代码包

└─tokenAnalysis //词法分析的源代码包

## 1.2 代码运行方式

方法1：

运行CompilerTest.java，程序会自动完成整套的词法和语法分析过程。

方法2：

①首先运行src/tokenAnalysis下的test.java，自动对所给的测试文件进行词法分析，得到token输出并存放在outputFile/tokenAnalysisOutput文件夹下，并用可唯一标识的文件名进行命名整理。②然后运行src/syntaxAnalysis下的test.java,自动对outputFile/tokenAnalysisOutput下的token文件进行语法分析，并将结果存储在outputFile/syntaxAnalysisOutput文件夹下。如果语法分析出错，分析出错前的语法树会被输出到文件中，并且最后一行输出错误行数及相关信息。如果正确，则输出完整的语法树。

# 2.自定义测试方案

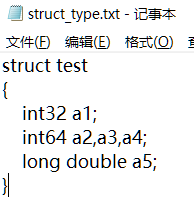
所有测试用例均为自己根据路径覆盖科学设计，完全覆盖了各产生式对应的抽象语法树中的所有情况，验证了代码实现的完整性和正确性。在代码实现过程中严格依据上下文无关文法编写递归下降程序，各产生式对应的函数编写严谨可靠，经过多次测试尚未发现问题，以下自定义测试方案若涵盖不全，请老师将代码文件放在inputFile文件夹下，运行程序进行语法分析测试。

## 2.1 struct\_type测试

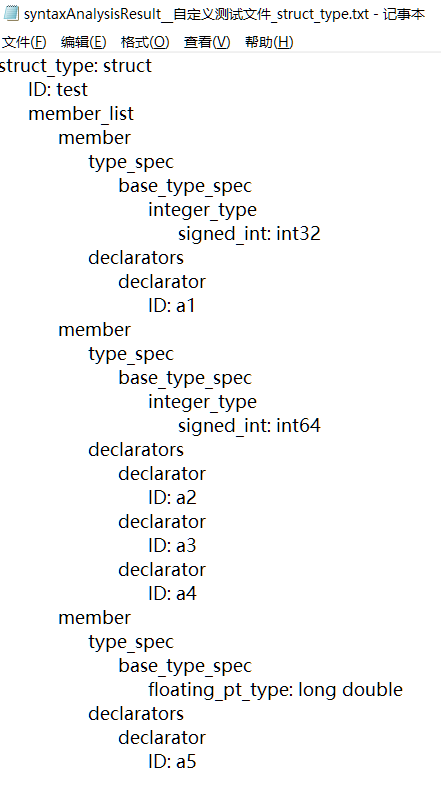
对产生式struct\_type->“struct” ID “{” member\_list “}”进行测试，对应的测试文件为inputFile/语法测试文件/自定义测试文件/struct\_type\*.txt。

**2.1.1正常代码测试**

输入：

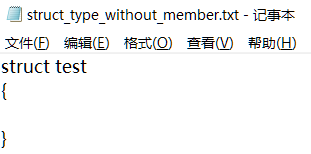


输出：

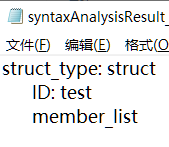


**2.1.2 struct中无member测试**

输入：



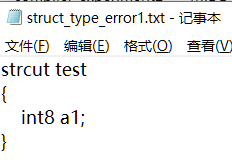
输出：



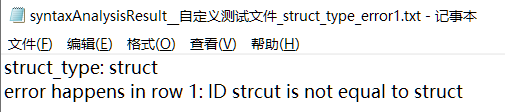
按照题目中给的产生式，但没有member时，struct\_type的表达式也应该是对的，但是老师给的测试文件中认为这个是错误情况，这是此次实验中的一个问题。

**2.1.3 struct匹配失败测试**

输入：



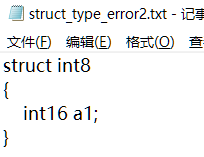
输出：



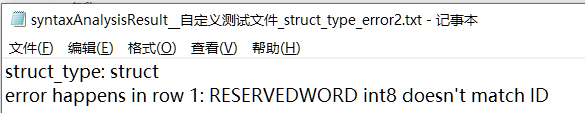
程序报错，指出第一行中的ID类型的单词strcut与关键字struct不匹配

**2.1.3 ID匹配失败测试**

输入：



输出：

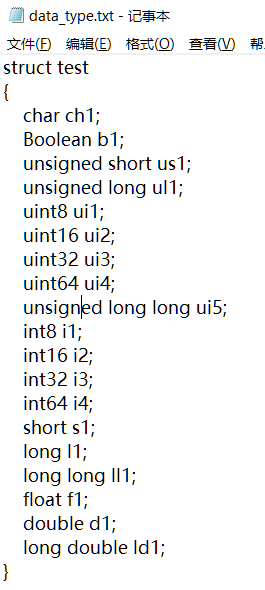


程序报错，指出第一行中的保留字类型的单词int8与ID不匹配。

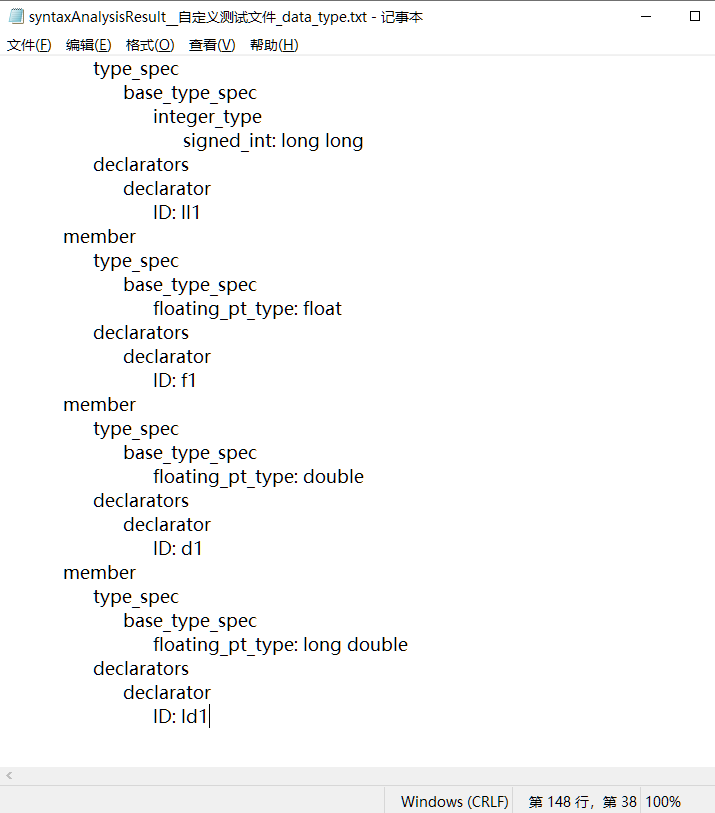
## 2.2 数据类型测试

对所有的数据类型：char，boolean，unsigned short，uint16，unsigned long，uint32，unsigned long long，uint64,uint8，short，int16，long，int32，long long，int64，int8，float，double，long double进行测试，对应的测试文件为inputFile/语法测试文件/自定义测试文件/data\_type.txt。

输入：



输出：

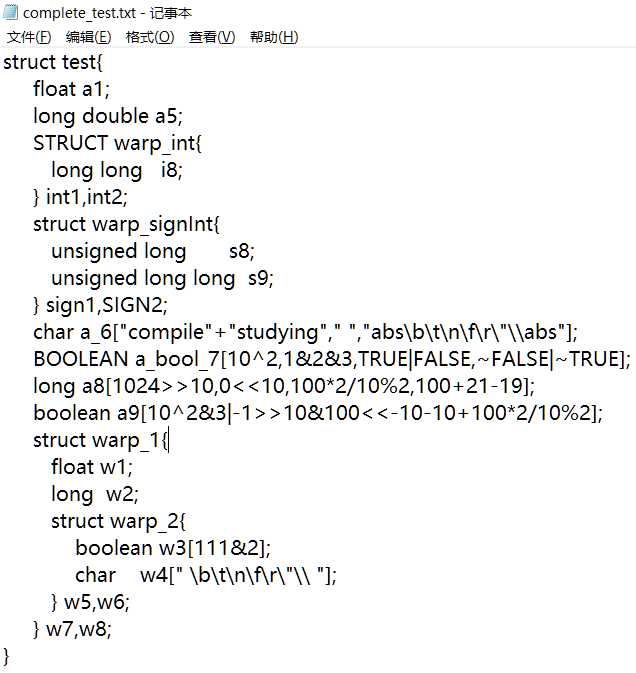


## 2.3 其他表达式综合测试

其他表达式指与struct\_type和基本的数据类型无关的表达式测试，如exp\_list,or\_expr，xor\_expr等。

编写代码，覆盖所给文法中所有可能出现的情况（有repeat的表达式出现至少两次便认为完全覆盖）。

输入：

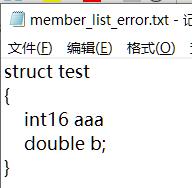


输出过长，请见“syntaxAnalysisResult\_\_自定义测试文件\_complete\_test.txt”

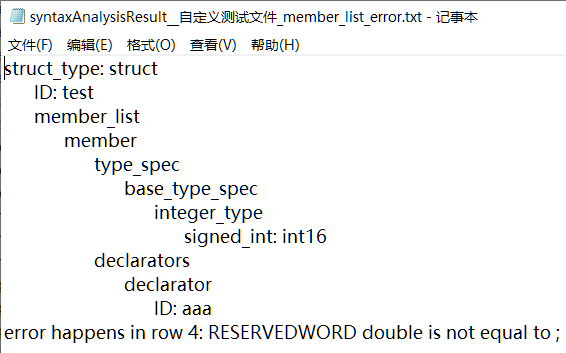
## 2.4 错误测试

**2.4.1 member\_list中“;”不匹配测试**

输入：



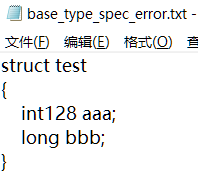
输出：



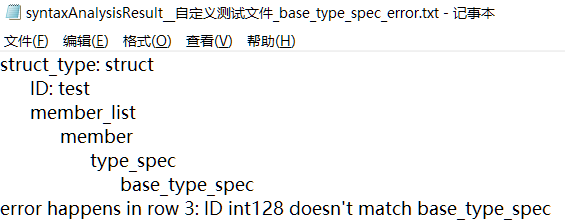
程序报错，指出第4行中的保留字类型的单词double与希望匹配的分号“；”不匹配。

**2.4.2 base\_type\_spec中基本数据类型不匹配测试**

输入：



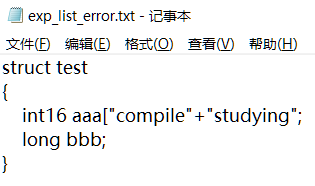
输出：



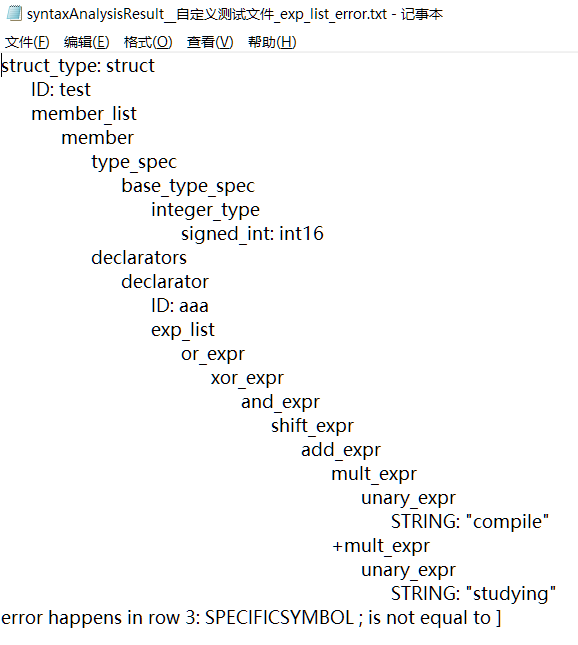
程序报错，指出第3行中的ID类型的单词int128与基本数据类型base\_type\_spec不匹配。

**2.4.3 exp\_list中”]”不匹配测试**

输入：



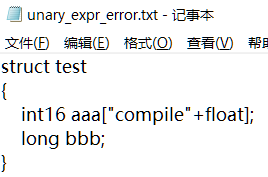
输出：



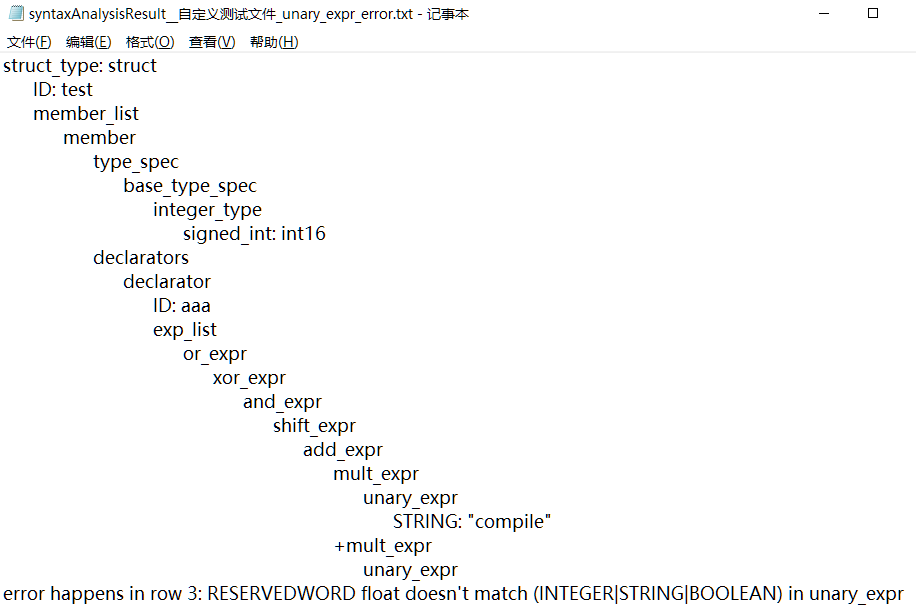
程序报错，指出第3行中的专用符号“；”与“]”不匹配。

**2.4.4 unary\_expr中(INTEGER|STRING|BOOLEAN)不匹配测试**

输入：



输出：



程序报错，指出第3行中的保留字类型的单词float与INTEGER|STRING|BOOLEAN均不匹配。