# Updated Syntax Analysis Grammar

Xavier

2025 年 3 月 7 日

## 声明

此文法仅针对样例 C++ 编写。规定大写字母为首的单词为非终结符，小写字母为首的为终结符，'$\cdots$' 表示可出现一次或多次，？表示可出现 0 次或 1 次，被 $$ 包裹的部分为一个整体，etc. 表示省略。

### 1. 主结构

Program → Decl ⋯ *(程序由一个或多个声明组成)*

### 2. 声明部分 (Decl)

Decl → StructDecl | Type id Suffix *(结构体声明或“类型 id + 后续”)*

VarDecl → Type id VarDeclSuffix

#### 2.1 结构体声明 (StructDecl)

StructDecl → struct id? { Decl ⋯ } id? ;

*(仅示例，表示结构体内可以继续声明变量、函数等)*

#### 2.2 声明后缀 (Suffix)

Suffix → ( Param) FunctionBody | VarDeclSuffix

这里是关键!

如果遇到 *"("*，说明它是函数声明;

否则走 *VarDeclSuffix* 路径处理变量声明。

##### 2.2.1 函数体 (FunctionBody)

FunctionBody → CompoundStatement | ; *(可以是函数定义或函数原型)*

##### 2.2.2 变量声明后缀 (VarDeclSuffix)

VarDeclSuffix → = Expression ; | ;

如果下一个符号是“=”，则初始化；否则直接分号结束。

### 3. 函数参数 (Param)

$$\text{Param} \rightarrow \epsilon \mid \text{ParamList} \quad \textit{(参数可为空，也可有若干个)}$$

$$\text{ParamList} \rightarrow \text{Type id} \left(\text{, Type id}\right) \cdots \quad \textit{(简化表示，多个 type id 逗号分隔)}$$

### 4. 语句部分 (Statement)

| | | | |
|---|---|---|---|
| Statement | → | CompoundStatement | 块语句 |
| | \| | SelectionStatement | 选择语句 |
| | \| | IterationStatement | 循环语句 |
| | \| | ReturnStatement | 返回语句 |
| | \| | ExpressionStatement | 表达式 |
| | \| | VarDecl | 变量声明语句 |
| | \| | StructDecl | 结构体声明语句 |
| | \| | break ; \| continue ; | *break* 语句和 *continue* 语句 |

注意！*Expression* 不含分号，*ExpressionStatement* 含有分号

CompoundStatement → { Statement ⋯ }

SelectionStatement → if ( Expression ) $ CompoundStatement | Statement $
$ else Statement | CompoundStatement $ ?

IterationStatement → while ( Expression ) $ Statement | CompoundStatement $

ReturnStatement → return ExpressionStatement

DeclInFunction → VarDecl | StructDecl

ExpressionStatement → Expression ;

循环语句日后还将支持 *for* 循环

### 5. 表达式部分 (Expression)

Expression $\rightarrow$

### 5. 表达式部分 (Expression)

$$
\begin{aligned}
\text{Expression} &\rightarrow \text{Expression} + \text{Term} \mid \text{Expression} - \text{Term} \mid \text{Term} \\
\text{Term} &\rightarrow \text{Term} * \text{Factor} \mid \text{Term} / \text{Factor} \mid \text{Factor} \\
\text{Factor} &\rightarrow \text{id} \mid \text{Literal} \mid (\ \text{Expression}) \\
\text{Literal} &\rightarrow \text{Numeric} \mid \text{string\_literal} \mid \text{char\_literal} \mid \text{boolean\_literal}
\end{aligned}
$$