1(menu)视图>>阅读视图

2双击表格

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *attribute-specifier-seq* | | | | | | | | |
| ( | *attribute-specifier* | | | | | | | )+ |
|  | ( | [ [ | *attribute-list* | ] ] | | | *alignment-specifier* | ) |  |
|  |  |  | (*attribute*( ...)?)?(,(*attribute*( ...)?)?)\* |  |  | (**alignas** ( *type-id*( ...)? )  |**alignas** ( *constant-expression*( ...)? )  ) |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *attribute* | | | | | | | | | | | | | |
| *attribute-token* | | | | | | | ( | *attribute-argument-clause* | | | | | )? |
| ( | *identifier* | | | *attribute-scoped-token* | | | ) |  | ( | *balanced-token-seq* | | | ) |  |
|  |  |  | *attribute-namespace* | :: | *identifier* |  |  |  | ( | *balanced-token* | )\* |  |  |
|  |  |  | *identifier* |  |  |  |  |  |  | (( *balanced-token-seq* )  |[ *balanced-token-seq* ]  |{ *balanced-token-seq* }  |any *token* other than a parenthesis, a bracket, or a brace  ) |  |  |  |

|  |
| --- |
| attribute-specifier-seq:  　　attribute-specifier-seq opt attribute-specifier  attribute-specifier:  　　[ [ attribute-list ] ]  　　alignment-specifier  alignment-specifier:  　　alignas ( type-id ... opt )  　　alignas ( constant-expression ... opt )  attribute-list:  　　attribute opt  　　attribute-list , attribute opt  　　attribute ...  　　attribute-list , attribute ...  attribute:  　　attribute-token attribute-argument-clause opt  attribute-token:  　　identifier  　　attribute-scoped-token  attribute-scoped-token:  　　attribute-namespace :: identifier  attribute-namespace:  　　identifier  attribute-argument-clause:  　　( balanced-token-seq )  balanced-token-seq:  　　balanced-token opt  　　balanced-token-seq balanced-token  balanced-token:  　　( balanced-token-seq )  　　[ balanced-token-seq ]  　　{ balanced-token-seq }  　　any token other than a parenthesis, a bracket, or a brace |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *init-declarator-list* | | | | | | | | | | |
| *init-declarator* | | | | | | | | | | (,*init-declarator*)\* |
| *declarator* | | | | | | | ( | *initializer* | )? |  |
| ( | *ptr-declarator* | | | *noptr-declarator* | *parameters-and-qualifiers* | *trailing-return-type* | ) |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *cv-qualifier-seq* | | | | | | |
| ( | *cv-qualifier* | | | | | )+ |
|  | ( | **const** | | | **volatile** | ) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *ref-qualifier* | | | | |
| ( | **&** | | | **&&** | ) |

|  |  |  |  |
| --- | --- | --- | --- |
| *ptr-declarator* | | | |
| ( | *ptr-operator* | )\* | *noptr-declarator* |
|  | (**\***( *attribute-specifier-seq*)?( *cv-qualifier-seq*)?  |**&**( *attribute-specifier-seq*)?  |**&&**( *attribute-specifier-seq*)?  |*nested-name-specifier* **\***( *attribute-specifier-seq*)?( *cv-qualifier-seq*)?  ) |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *noptr-declarator* | | | | | | |
| ( | *declarator-id* | ( *attribute-specifier-seq*)? | | | ( *ptr-declarator* ) | ) | (*parameters-and-qualifiers*  |[( *constant-expression*)? ]( *attribute-specifier-seq*)?  )\* |
|  | (...)? *id-expression* |  |  |  |  |  |

|  |
| --- |
| *parameters-and-qualifiers* |
| ( *parameter-declaration-clause* )( *cv-qualifier-seq*)?( *ref-qualifier*)?( *exception-specification*)?( *attribute-specifier-seq*)? |

|  |
| --- |
| *trailing-return-type* |
| -> *trailing-type-specifier-seq*( *abstract-declarator*)? |

|  |  |  |  |
| --- | --- | --- | --- |
| *type-id* | | | |
| *type-specifier-seq* | ( | *abstract-declarator* | )? |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| *abstract-declarator* | | | | | | |
| ( | *ptr-abstract-declarator* | | | (*noptr-abstract-declarator*)? *parameters-and-qualifiers* *trailing-return-type* | | | *abstract-pack-declarator* | ) |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *ptr-abstract-declarator* | | | | | |
| ( | *noptr-abstract-declarator* | | | | (*ptr-operator* )+(*noptr-abstract-declarator*)? | ) |
|  | (( *ptr-abstract-declarator* ))? | (*parameters-and-qualifiers*  |[( *constant-expression*)? ]( *attribute-specifier-seq*)?  )\* |  |  |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *abstract-pack-declarator* | | | | | | | | | |
| ( | *ptr-operator* | )\* | *noptr-abstract-pack-declarator* | | | | | | |
|  |  |  | ( | *parameters-and-qualifiers* | | | [( *constant-expression*)? ]( *attribute-specifier-seq*)? | | | ... | )\* |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *parameter-declaration-clause* | | | | | | | | | | | | |
| ( | ( | *parameter-declaration-list* | | | | )? | ( | ... | )? | | | parameter-declaration-list , ... | ) |
|  |  | *parameter-declaration* | ( | , *parameter-declaration* | )\* |  |  |  |  |  |  |  |
|  |  | (*attribute-specifier-seq*)? *decl-specifier-seq*  (*declarator*  |*abstract-declarator*  )  ( = *initializer-clause*)? |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *function-definition* | | | | | | | | | | |
| ( | *attribute-specifier-seq* | )? | ( | *decl-specifier-seq* | )? | *declarator* | ( | *virt-specifier-seq* | )? | *function-body* |
|  |  |  |  |  |  |  |  |  |  | ((*ctor-initializer*)? *compound-statement*  |*function-try-block*  |= **default** ;  |= **delete** ;  ) |

|  |
| --- |
| init-declarator-list:  　　init-declarator  　　init-declarator-list , init-declarator  init-declarator:  　　declarator initializer opt  declarator:  　　ptr-declarator  　　noptr-declarator parameters-and-qualifiers trailing-return-type  ptr-declarator:  　　noptr-declarator  　　ptr-operator ptr-declarator  noptr-declarator:  　　declarator-id attribute-specifier-seq opt  　　noptr-declarator parameters-and-qualifiers  　　noptr-declarator [ constant-expression opt ] attribute-specifier-seq opt  　　( ptr-declarator )  parameters-and-qualifiers:  　　( parameter-declaration-clause ) cv-qualifier-seq opt ref-qualifier opt exception-specification opt attribute-specifier-seq opt  trailing-return-type:  　　-> trailing-type-specifier-seq abstract-declarator opt  ptr-operator:  　　\* attribute-specifier-seq opt cv-qualifier-seq opt  　　& attribute-specifier-seq opt  　　&& attribute-specifier-seq opt  　　nested-name-specifier \* attribute-specifier-seq opt cv-qualifier-seq opt  cv-qualifier-seq:  　　cv-qualifier cv-qualifier-seq opt  cv-qualifier:  　　const  　　volatile  ref-qualifier:  　　&  　　&&  declarator-id:  　　... opt id-expression  type-id:  　　type-specifier-seq abstract-declarator opt  abstract-declarator:  　　ptr-abstract-declarator  　　noptr-abstract-declarator opt parameters-and-qualifiers trailing-return-type  　　abstract-pack-declarator  ptr-abstract-declarator:  　　noptr-abstract-declarator  　　ptr-operator ptr-abstract-declarator opt  noptr-abstract-declarator:  　　noptr-abstract-declarator opt parameters-and-qualifiers  　　noptr-abstract-declarator opt [ constant-expression opt ] attribute-specifier-seq opt  　　( ptr-abstract-declarator )  abstract-pack-declarator:  　　noptr-abstract-pack-declarator  　　ptr-operator abstract-pack-declarator  noptr-abstract-pack-declarator:  　　noptr-abstract-pack-declarator parameters-and-qualifiers  　　noptr-abstract-pack-declarator [ constant-expression opt ] attribute-specifier-seq opt  parameter-declaration-clause:  　　parameter-declaration-list opt ... opt  　　parameter-declaration-list , ...  parameter-declaration-list:  　　parameter-declaration  　　parameter-declaration-list , parameter-declaration  parameter-declaration:  　　attribute-specifier-seq opt decl-specifier-seq declarator  　　attribute-specifier-seq opt decl-specifier-seq declarator = initializer-clause  　　attribute-specifier-seq opt decl-specifier-seq abstract-declarator opt  　　attribute-specifier-seq opt decl-specifier-seq abstract-declarator opt = initializer-clause  function-definition:  　　attribute-specifier-seq opt decl-specifier-seq opt declarator virt-specifier-seq opt function-body  function-body:  　　ctor-initializer opt compound-statement  　　function-try-block  　　= default ;  　　= delete ;  initializer:  　　brace-or-equal-initializer  　　( expression-list )  brace-or-equal-initializer:  　　= initializer-clause  　　braced-init-list  initializer-clause:  　　assignment-expression  　　braced-init-list  initializer-list:  initializer-clause ... opt  　　initializer-list , initializer-clause ... opt  braced-init-list:  　　{ initializer-list , opt }  　　{ } |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *initializer* | | | | | | | | | | | | | | | | | | | | |
| ( | *brace-or-equal-initializer* | | | | | | | | | | | | | | | | | | | ( *expression-list* ) | ) |
|  | ( | = | *initializer-clause* | | | | | | | *braced-init-list* | | | | | | | | ) |  |  |  |
|  |  |  | ( | *assignment-expression* | | | *braced-init-list* | ) |  | { | ( | *initializer-list* | ( | , | )? | )? | } |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  | *initializer-clause*( ...)?  ( , *initializer-clause*( ...)?)\* |  |  |  |  |  |  |  |  |  |