第32章 新增预处理器功能和宏

《现代C++语言核心特性解析》 谢丙堃

特殊运算符__has_include

- 判断某个头文件是否能够被包含进来

```
#if __has_include(<optional>)
#  include <optional>
#  define have_optional 1
#elif __has_include(<experimental/optional>)
#  include <experimental/optional>
#  define have_optional 1
#  define experimental_optional 1
#else
#  define have_optional 0
#endif
```

特性测试宏

- 属性测试运算符

```
std::cout << __has_cpp_attribute(deprecated);
```

特性测试宏

- 语言功能特性测试宏

宏	值
__cpp_aggregate_bases	201603L
__cpp_aggregate_nsdmi	201304L
__cpp_aggregate_paren_init	201902L
__cpp_alias_templates	200704L
__cpp_aligned_new	201606L
__cpp_attributes	200809L
...	...
__cpp_unicode_characters	200704L
__cpp_unicode_literals	200710L
__cpp_user_defined_literals	200809L
__cpp_using_enum	201907L
__cpp_variable_templates	201304L
__cpp_variadic_templates	200704L
__cpp_variadic_using	201611L

特性测试宏

- 标准库功能特性测试宏

宏	值	头文件
__cpp_lib_addressof_constexpr	201603L	<memory>
__cpp_lib_allocator_traits_is_always_equal	201411L	<memory> <scoped_allocator> <string> <deque> <forward_list> <list> <vector> <map> <set> <unordered_map> <unordered_set>
__cpp_lib_any	201606L	<any>
__cpp_lib_apply	201603L	<tuple>
…	…	…
__cpp_lib_uncaught_exceptions	201411L	<exception>
__cpp_lib_unordered_map_try_emplace	201411L	<unordered_map>
__cpp_lib_unwrap_ref	201811L	<functional>
__cpp_lib_variant	201606L	<variant>
__cpp_lib_void_t	201411L	<type_traits>

新增宏__VA_OPT__

- 可变参数宏的问题

```
#define LOG(msg, ...) printf("[" __FILE__ ":%d] " msg, __LINE__, __VA_ARGS__)
LOG("Hello 2020");
// 展开后
printf("[" __FILE__ ":%d] " "Hello 2020", __LINE__, );
```

- 可变参数不为空时才展开

```
#define LOG(msg, ...) printf("[" __FILE__ ":%d] " msg, __LINE__ __VA_OPT__(,) __VA_ARGS__)
```

感谢您的观看

欢迎关注