

Chapter2 cmake头文件引入

1. 在myCmake文件夹下，新建文件夹chapter1.2，并进入它

2. 设计如图所示的文件层次：

```
txt
1  | — CMakeLists.txt
2  |   |
3  |   | — include
4  |   |   |
5  |   |   | — Hello.h
6  |   |   |
   |   |   | — src
   |   |   |   |
   |   |   |   | — Hello.cpp
   |   |   |   | — main.cpp
```

3. 填充程序内容：

1. Hello.h（在include文件夹下创作头文件，以让别人调用）

```
/*声明了Hello类，Hello的方法是print(),*/

#ifndef __HELLO_H__
#define __HELLO_H__

class Hello
{
public:
    void print();
};

#endif
```

2. Hello.cpp（在src文件夹下调用头文件内的 Hello::print() 函数）

```
/*实现了Hello::print()*/
#include <iostream>

#include "Hello.h"

void Hello::print()
{
    std::cout << "Hello Headers!" << std::endl;
}
```

3. main.cpp（在src文件夹下调用头文件内的Hello::print()函数）

```
#include "Hello.h"

int main(int argc, char *argv[])
{
    Hello hi;
    hi.print();
    return 0;
}
```

4. 填充list文本（设定规则的区域）

```
cmake_minimum_required(VERSION 3.5)#最低CMake版本

project (hello_headers)# 工程名

set(SOURCES
    src/Hello.cpp
    src/main.cpp
)#创建一个变量，名字叫SOURCE。它包含了所有的cpp文件。

add_executable(hello_headers ${SOURCES})#用所有的源文件生成一个可执行文件，因为这里定义了SOURCE变量，所以就不需要罗列cpp文件了
#等价于命令：      add_executable(hello_headers src/Hello.cpp src/main.cpp)

target_include_directories(hello_headers
    PRIVATE
        ${PROJECT_SOURCE_DIR}/include
)#设置这个可执行文件hello_headers需要包含的库的路径

#PROJECT_SOURCE_DIR指工程顶层目录
#PROJECT_Binary_DIR指编译目录
#PRIVATE指定了库的范围，下一节讲
```

1. 一般格式：

- [1] 声明cmake版本
- [2] project(项目名字/可执行文件名)
- [3] set(SOURCES) 对本轮所有需要编译的文件总和进行打包
- [4] add_executable(可执行文件名 待编译的列表中文件) 用所有的源文件[在前文打包成set了，直接传入set即可]生成一个可执行文件
- [5] target_include_directories(hello_headers PRIVATE \${PROJECT_SOURCE_DIR}/include) 设置这个可执行文件hello_headers需要包含的库的路径

2. 注意⚠：

- 与1.1不同的是用set进行“文本打包”
- 需要导入库路径

3. 仔细解释——如何导入“包含目录”（即：此处的库函数路径）：

- 当有其他需要包含的文件夹（文件夹里有头文件）时，可以使用以下命令使编译器知道它们： target_include_directories () 。
- 编译此目标时，这将使用-I标志将这些目录添加到编译器中，例如 -I/目录/路径

```
target_include_directories(target
    PRIVATE
        ${PROJECT_SOURCE_DIR}/include
)
```

PRIVATE 标识符指定包含的范围！

4.常用目录变量：

CMake语法指定了许多变量，可用于帮助您在项目或源代码树中找到有用的目录

Variables	Infomation
CMAKE_SOURCE_DIR	根源代码目录，工程顶层目录。暂认为就是PROJECT_SOURCE_DIR
CMAKE_CURRENT_SOURCE_DIR	当前处理的 CMakeLists.txt 所在的路径

Variables	Infomation
PROJECT_SOURCE_DIR	工程顶层目录
CMAKE_BINARY_DIR	运行cmake的目录。外部构建时就是build目录
CMAKE_CURRENT_BINARY_DIR	The build directory you are currently in.当前所在build目录
PROJECT_BINARY_DIR	暂认为就是CMAKE_BINARY_DIR

5. 查看输出 and 总览全局

```
mkdir build
cd build/
cmake ..
make VERBOSE=1
```

输出：

```
/Applications/CMake.app/Contents/bin/cmake -S/Users/huluobo/cmake-examples/myCmake/chapter1.2 -
B/Users/huluobo/cmake-examples/myCmake/chapter1.2/build --check-build-system CMakeFiles/Makefile.cmake 0
这行命令是 CMake 在检查构建系统，确认 Makefile 是否存在和正确。
[key: --check-build-system]

/Applications/CMake.app/Contents/bin/cmake -E cmake_progress_start /Users/huluobo/cmake-
examples/myCmake/chapter1.2/build/CMakeFiles /Users/huluobo/cmake-
examples/myCmake/chapter1.2/build//CMakeFiles/progress.marks
这行命令是 CMake 开始跟踪构建进度。
[key: progress.marks]

/Library/Developer/CommandLineTools/usr/bin/make -f CMakeFiles/Makefile2 all
make[1]: Nothing to be done for `all'.
这行命令是 make 工具开始执行 Makefile2 文件中的 "all" 目标。
这是 make 工具的输出，表示 "all" 目标没有需要执行的任务。这通常意味着所有的源文件都已经被编译过，没有新的更改需要被编译。

/Applications/CMake.app/Contents/bin/cmake -E cmake_progress_start /Users/huluobo/cmake-
examples/myCmake/chapter1.2/build/CMakeFiles 0
```

全局：

```
├─ CMakeLists.txt
├─ build
│   ├── CMakeCache.txt
│   ├── CMakeFiles
│   │   ├── 3.28.0-rc5
│   │   │   ├── CMakeCCompiler.cmake
│   │   │   ├── CMakeCXXCompiler.cmake
│   │   │   ├── CMakeDetermineCompilerABI_C.bin
│   │   │   ├── CMakeDetermineCompilerABI_CXX.bin
│   │   │   ├── CMakeSystem.cmake
│   │   │   ├── CompilerIdC
│   │   │   │   ├── CMakeCCompilerId.c
│   │   │   │   ├── CMakeCCompilerId.o
│   │   │   │   └─ tmp
│   │   │   └─ CompilerIdCXX
│   │   │       ├── CMakeCXXCompilerId.cpp
│   │   │       ├── CMakeCXXCompilerId.o
│   │   │       └─ tmp
│   ├── CMakeConfigureLog.yaml
│   ├── CMakeDirectoryInformation.cmake
│   └─ CMakeScratch
```

```
| | | | Makefile.cmake
| | | | Makefile2
| | | | TargetDirectories.txt
| | | | cmake.check_cache
| | | | pkgRedirects
| | | | progress.marks
| | | |
| | | | Makefile
| | | | cmake_install.cmake
| | | |
| | | | include
| | | | | Hello.h
| | | |
| | | | src
| | | | | Hello.cpp
| | | | | main.cpp
```

12 directories, 23 files