

cpp_novice_template

[English](#)

这是一个为新手学习 C++ 提供的模板. 使用它不需要任何 C++ 或 CMake 经验.

软件需求

你可以通过 [Windows/MacOS/Linux 上 VSCode 配置 C++: Clang + Clang-based Tools + CMake + Conan](#) 安装所有软件.

- Git
- 一个支持 CMake 的 C++ IDE (最新版的 Visual Studio, Qt Creator, CLion 等)

下载和解压

如果已经下载, 请无视这部分.

1. 点击接近网页顶部的绿色 `Code` 按钮.
2. 点击弹出的 `Download ZIP` 按钮. 这会将本仓库最新版代码下载为一个 zip 压缩文件.
3. 解压该压缩文件到你放置代码的地方.

使用

1. 打开你的 IDE (最新版的 Visual Studio, Qt Creator, CLion 等) 或 [配置好的编辑器 \(VSCode 等\)](#).
2. 在 IDE 中, 按 `文件夹` 或 按 `CMake 项目` 打开解压的文件夹.

如何添加新的程序

基本使用

用 cmake 项目学习 C++ 最好的一点是, 一个项目就能管理多个程序: 你不必新建一个项目来进行下一个练习.

在本模板中, 你按以下步骤简单添加一个程序:

1. 打开根目录下的 `CMakeLists.txt`.
2. 添加 `add_program(<program_name> <source_file1> [source_file2...])`. 例如,
 - `add_program(example_single src/example_single/main.cpp)` 添加了一个可执行程序名为 `example_single`, 其相关的代码文件有 `src/example_single/main.cpp`. 该代码文件中有一个 `int main()` 函数, 它作为程序运行的入口. 自此我们就能通过 cmake 从对应的代码生成该程序, 而后执行该程序.
 - `add_program(example_multiple src/example_multiple/main.cpp src/example_multiple/hello.cpp)` 添加了一个可执行程序名为 `example_multiple`, 其相关的代码文件有 `src/example_multiple/main.cpp` 和 `src/example_multiple/hello.cpp`. 这些代码文件中仅有一个 `int main()` 函数, 它作为程序运行的入口. 自此我们就能通过 cmake 从对应的代码生成该程序, 而后执行该程序.

3. 通过某些按键或重新打开 IDE, 来重新配置本项目.

强烈建议将源文件放进 `src` 文件夹内.

头文件

至于头文件 (`.h`, `.hpp` 等), 你可以简单将它和源文件放在一起. 之后源文件就能正确地 `#include "<header_file>"`. 例如, 在 `src/example_multiple` 文件夹中, `hello.cpp` 文件可以直接 `#include "hello.hpp"`.

如果你想要让一个头文件可以被任意位置的源文件包含, 你可以将它放入 `include` 文件夹. 例如, 在 `src/example_single` 文件夹中, `main.cpp` 文件可以 `#include "add.hpp"`, 而 `add.hpp` 是放在 `include` 文件夹里的.

安装第三方库

见于 [请读我_安装第三方库](#).

参考资料

我基本是通过 [Modern CMake for C++](#) 学习 CMake 内容.

另外, 本仓库非常依赖于 [aminya/project_options](#), 它大量改善了 CMake 的使用体验.

对于 conan 2.0, [官方文档](#) 已经足够有用.

本仓库的细节见于 [对配置文件的解释](#).