

05 How C++ Works

1. 前言

调好了.vssettings,VsVim,但是Ob的 `imap jj <Esc>` 死活搞不定, 事已至此, 学习吧。

(第二天发现重启后映射就能用了.....)

从25课回来的补充, 视频中Cherno用的是Visual Assist, 正版价格对国内不友好。而VS可以装一个JetBrains ReSharper C++插件, 教育邮箱可白嫖, 爽飞了。

2. 基本概念

2.1 头文件

带有 `#` 的为 *preprocessor statement*, 即预处理指令

该类指令发生在真正的编译之前

当编译器收到一个源文件时, 做的第一件事情就是预处理所有预处理指令

```
#include <iostream>
```

C++

即找到 `iostream` 文件, 将该文件中的所有内容复制粘贴到目前的文件里

这些被include的文件一般被称为 *header file*, 即头文件

每个cpp程序都有一个类似 `main` 函数的东西, 被称作 *entry point*

2. 基本概念

2.1 头文件

带有 `#` 的为 *preprocessor statement*, 即预处理指令

该类指令发生在真正的编译之前

当编译器收到一个源文件时, 做的第一件事情就是预处理所有预处理指令

```
#include <iostream>
```

C++

即找到 `iostream` 文件, 将该文件中的所有内容复制粘贴到目前的文件里

这些被include的文件一般被称为 *header file*, 即头文件

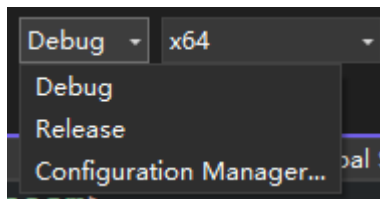
每个cpp程序都有一个类似 `main` 函数的东西, 被称作 *entry point*

后面了解到这个必须的原因在于.exe可执行文件必须需要一个入口, 可在设置中自定义入口点 (不一定必须是main function)

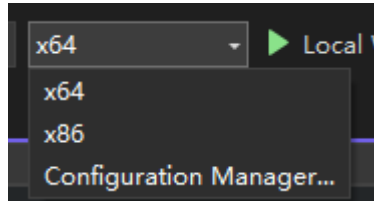
只有主函数可以不返回任何值, 默认返回0

preprocessor statement 被评估后, 我们的文件会被编译, 这个阶段, 编译器把我们的cpp代码转化为实际的机器码

2.2 Visual Studio相关

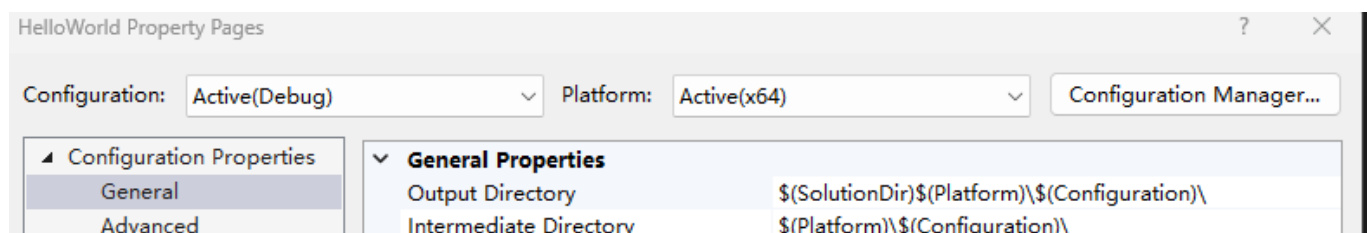


配置`Configuration`是如何build一个项目的一系列规则



`Solution platform`使我们目前编译的目标平台，x86就是生成一个用于windows的32位程序，你要求弄安卓程序就改成安卓相关的

若要修改我们的配置，右键项目点击`Properties`,得到下面界面



注意:确保`Active`的是你想修改的，有可能会不是

2.3 Linker链接器

所有.cpp文件都会被编译，.h头文件不会被编译，它会通过include到cpp中，然后被编译
每个.cpp文件都被编译成一个`Object`文件 (.obj)，将所有.obj文件连在一起，成为一个.exe文件

`Compile`编译单个.cpp文件，快捷键 `Ctrl+F7`

注意:error list 不好用，工作原理是`parse`(解析)output窗口，找error关键字，然后从那里找信息再放入list,仅当做概述看。
要具体的、所有的错误信息，只能通过output窗口

双击output窗口的error信息即可跳转到错误行

2.4 Declaration声明

编译器识别不出另一个.cpp文件中的函数，所以报告编译错误吗，可以通过提供一个声明`declaration`来解决

Declaration	Defination
声明函数存在	这个函数是什么，函数的主体
void Log(x)	void Log(x){ code...}

文件被编译后，linker会去找函数的定义，然后跟main中调用的联系起来。如果找不到会得到linker error
只有主函数可以不返回任何值，默认返回0

`preprocessor statement`被评估后，我们的文件会被编译，这个阶段，编译器把我们的cpp代码转化为实际的机器码