

【注意:】

- 1、除明确要求外，已学过的知识中，**不允许**使用 goto、**不允许**使用全局变量，**不允许**使用 C++ 的 STL 容器等后续知识，**允许使用 string 类**
- 2、**允许**使用 C++ 的 string 类
- 3、本作业仅要求 VS2022 编译通过即可（“0 errors, 0 warnings”）
- 4、不允许使用 scanf/printf 进行输入/输出
- 5、**题目本身是难度不高，但后续大作业中都会用到，必须完成!!!**

综合题 3: main 函数参数分析类的封装及应用

【问题背景描述:】

在命令行方式下执行程序时，main 函数带参数一种常见的形式，参数主要分为两种：固定参数及可变参数，其中固定参数是每次必须的，可变参数按需给出，可变参数还可能会跟附加值

例 1: Windows 下的 ping 命令

ping 10.80.42.230 : 仅有 IP 地址这个固定参数，其余值用默认

ping -t -l 1024 10 10.80.42.230 : 可变参数为 -t 和 -l，且 -l 还带一个整数值，固定参数为 IP 地址，其余值用默认

例 2: Linux 下的 c++ 命令

c++ test.cpp: 编译源程序文件 test.cpp，生成可执行文件 a.out，其余用默认

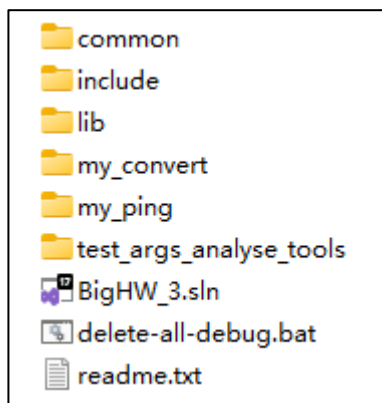
c++ -Wall -o test test.cpp : 可变参数为 -Wall 和 -o test，指定全部 warning 当做 error 以及可执行文件名为 test，其余用默认

【main 函数参数分析类的实现:】

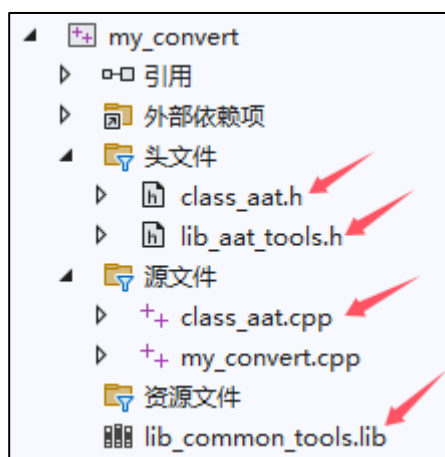
- 1、完成一套内部工具集 class_aat (aat = args_analyse_tools)，用于 main 函数带参数方式下对可变参数进行通用分析，具体要求如下：
 - 所有可变参数均以 --（双减号）开始，相互位置任意，后面再跟 0~n 个固定参数
 - ◆ 固定参数是否有附加值，相互之间的位置是否固定等，由具体应用程序决定，本工具集不考虑固定参数的问题
 - ◆ 必须可变参数在前，固定参数在后
 - ◆ 为了降低难度，约定可变参数的附加值最多一个，同一个可变参数不允许多次出现
 - 参数的类型有 bool/int/int 集合/double/double 集合/string/string 集合/IP 地址等几种
 - ◆ bool 型：后面无附加参数，有参数为 true，无参数为 false
 - ◆ int 型：后面有 1 个 int 型附加参数，初始化时指定数据范围，要求附加参数的值必须在数据范围内，如果不在，有两种处理方式：**置缺省值或报错**
 - ◆ int 型集合：后面有 1 个 int 型附加参数，参数值必须落在指定集合内，如果不在，有两种处理方式：**置集合中的某个缺省值或报错**
 - ◆ double 型：后面有 1 个 double 型附加参数，初始化时指定数据范围，要求附加参数的值必须在数据范围内，如果不在，有两种处理方式：**置缺省值或报错**
 - ◆ double 型集合：后面有 1 个 double 型附加参数，参数值必须落在指定集合内，如果不在，有两种处理方式：**置集合中的某个缺省值或报错**
 - ◆ string 型：后面有 1 个 string 型附加参数（如果内容带空格，则需要加""），string 没有正确性要求，任意字符串均可；初始化时还可以指定缺省值，如果该参数未给出，则取缺省值（不指定缺省值则为""）即可
 - ◆ string 型集合：后面有 1 个 string 型附加参数（如果带空格，则前后加""即可），要求该参数值必须落在指定集合内，如果不在，有两种处理方式：**置集合中的某个缺省值或报错**

- ◆ IP 地址型：后面有 1 个 string 型附加参数，要求该字符串符合“xxx.xxx.xxx.xxx”的点分十进制形式（三个点，四个数，每个数 0-255 之间），如果不符合，有两种处理方式：**置缺省值或报错**
- ◆ 细节请仔细阅读 test_args_analyse_tools、my_ping、my_convert、hw_check 等项目中的测试用例
- args_analyse_print 函数输出参数表时，每行开头一个空格，每列宽度为列名/该列所有内容的最大宽度，列之间一个空格，因此分隔线（===）需要动态确定
- 给出 lib_common_tools.lib 供参考（lib 的使用方法同前，宏定义在开关 class_aat.h 中）

2、附件给出的大作业模板 BigHW_3.rar 及对应文件目录的说明



- 共 9 个目录/文件，但 BigHW_3.sln 打开后应有 5 个项目（**注意缺省为 x64，要改!!!**）
- lib 目录在解决方案中无对应项目，文件 lib_common_tools.lib/lib_common_tools.pdb 是老师实现的工具集对应的静态库文件（因为既不需要编译，也不需要查看，所以 sln 中无项目对应）
- common：公共函数集的源码（本项目已设置为跳过，不需要编译，仅为方便查看）
 - ◆ 有新增文件 class_aat.cpp，需要给出完整实现
- include：公共函数集对应的头文件（含义不变，有新增文件 class_aat.h 和 lib_aat_tools.h，本项目已设置为跳过，不需要编译，仅为方便查看）
 - ◆ 有新增文件 class_aat.h，允许按需修改
 - ◆ 有新增文件 lib_aat_tools.h 文件，不准修改
- my_convert：仿照 15-b3 写的 aat 工具集的测试用例
 - ◆ 源文件只有 my_convert.cpp 一个
 - ◆ 不需要改动，要保证编译通过



注意：

- 1、打开的项目中包含的文件如左图所示，其中仅 my_convert.cpp 在本目录中，其它四个（箭头所指）均不在本目录中
- 2、再次强调：公共函数的头文件、源文件及库文件只允许存在于（include/common/lib）中，**不允许**在各项目的目录下再次存在，**否则得分为 0!!!**

- my_ping：仿照 14-b4 写的 aat 工具集的测试用例
 - ◆ 源文件只有 my_ping.cpp 一个，注意事项同 my_convert
 - ◆ 不需要改动，要保证编译通过

- test_args_analyse_tools: 本次需要实现的 aat 工具集的测试用例
 - ◆ 源文件只有 test_args_analyse_tools.cpp 一个, 注意事项同 my_convert
 - ◆ 不需要改动, 要保证编译通过

3、本次给出的附件与你已有的 BigHW 的合并

方法 1: 在已完成作业的 BigHW 基础上加入本模板中的新增项

- 先把对应目录(lib/test-gmw/90-01-b2-gmw/90-02-b1-gmw)整体复制过来, 再在 sln 中添加现有项目的方法(解决方案“BigHW”上右键菜单-“添加”-“已有项目...”)来添加这些已有项目
- 将 common/include 中的 class_aat.cpp/class_aat.h/lib_aat_tools.h 这三个新增文件复制到对应目录中(必做), 再在 common/include 的项目中对应添加这三个文件(非必做项, 因为这两个项目是不编译的, 放进去只是为了查看方便而已)

方法 2: 以本模板为准, 将已完成作业的对应内容复制过来(包括 sln 和目录两方面)

- 先把对应目录中的文件复制过来, 再在 sln 的各项添加, 使能编译通过
- 注意:** 无论哪种方法(建议方法 1), **必须保证**所有已完成项目可再次编译成功

【完成要求:】

1、完成 class_aat 的定义与实现

- 补充完整 class_aat.h
- 完成 class_aat.cpp 的实现
- 通过 test_args_analyse_tools.cpp 的测试(class_aat.h 中的 ENABLE_LIB_COMMON_TOOLS 值为 0/1 时一致)

2、在此基础上移植小作业 14-b4 (ping 命令参数分析)

- 对应项目为 my_ping(class_aat.h 中的 ENABLE_LIB_COMMON_TOOLS 值为 0/1 时一致)
- my_ping.cpp 中有条件编译参数 IPADDR_IS_FIXED_ARGS, 要求 0/1 时均能通过测试
- my_ping.cpp 的大部分代码已给出, IPADDR_IS_FIXED_ARGS 为 1 时要补全 is_ipaddr_valid 函数的实现, IPADDR_IS_FIXED_ARGS 为 0 时什么都不用做
- 其它
 - ◆ 认真阅读, 理解一个具体程序的特有的参数处理逻辑
 - ◆ 原要求的 -t/-n/-l 改为 -t/--n/--l (单-改为双-)
 - ◆ 原来要求重复参数时覆盖(-n 10 -n 20), 现改为通过本工具集能检出错误即可
 - ◆ 其它要求不变
- 参数分析部分必须使用本工具集, 如果简单的按小作业修改后加入本次 BigHW 中提交, 则 **期末总分-20**
- 本项目和 test_args_analyse_tools 项目的区别: IPADDR_IS_FIXED_ARGS 为 1 时有一个固定参数

3、在此基础上移植小作业 15-b3 (Windows/Linux 文件格式互转)

- 对应项目为 my_convert(class_aat.h 中的 ENABLE_LIB_COMMON_TOOLS 值为 0/1 时一致)
- my_convert.cpp 中有条件编译参数 SRC_DST_IS_FIXED_ARGS, 要求 0/1 时均能通过测试
- 其它
 - ◆ 认真阅读, 理解一个具体程序的特有的参数处理逻辑
 - ◆ 原要求的 --convert 为必选, 现改为不选时默认 wtol
 - ◆ 文件名的正确性不做检查, 具体转换不需要实现
- 参数分析部分必须使用本工具集, 如果简单的按小作业修改后加入本次 BigHW 中提交, 则 **期末总分-20**
- 本项目和 test_args_analyse_tools 项目的区别: SRC_DST_IS_FIXED_ARGS 为 1 时有两个固定参数

【提交要求（仔细阅读，当心0分!!!）：】

- 1、提交作业前，先做好完整备份
- 2、本次的lib及之前大作业的lib记得删掉（项目文件中的lib同步删除、将class_aat.h中的ENABLE_LIB_COMMON_TOOLS改为0）
- 3、按之前的BigHW提交要求，整个BigHW目录压缩成BigHW.rar，再按网页要求改名后提交

【作业要求：】

- 1、仅需要在VS2022下编译通过即可，要做到“0 errors, 0 warnings”
- 2、**11月17日前（两周）**网上提交本次作业
- 3、每题所占平时成绩的具体分值见网页
- 4、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明