这里要编译的工程包含一个库和一个可执行文件。可执行文件依赖于库，所以要先编译库，编译后库放在lib目录里面，可执行文件放在bin目录里面。

.

目录结构如下：

├── app

│   ├── app.pro

│   ├── appwindow.cpp

│   ├── appwindow.h

│   └── main.cpp

├── bin

├── complex.pro

├── include

│   └── base.h

├── lib

└── src

├── base.cpp

└── src.pro

全局的工程文件complex.pro在工程根目录下，内容如下：

TEMPLATE = subdirs

SUBDIRS = src app

CONFIG += ordered

subdirs模板表示要编译多个子目录中的工程。CONFIG选项要求各个子工程按顺序编译，子目录的编译顺序在SUBDIRS中指明。

库的工程文件在src目录中，名为src.pro， 内容如下：

TARGET = base

VERSION = 0.1.0

CONFIG += static

DESTDIR = ../lib

INCLUDEPATH += ../include

DEPENDPATH += . ../include

SOURCES += base.cpp

HEADERS += base.h

TARGET表示编译后的库名字为base，CONFIG += static表示编译后的库是静态库。DESTDIR设置编译后的库的位置。

库编译结束后就是可执行文件编译了，工程文件在app目录下的app.pro，内容如下：

TEMPLATE = app

TARGET = app

DESTDIR = ../bin

INCLUDEPATH += . ../include

DEPENDPATH += .

LIBS += -L../lib -lbase

SOURCES += appwindow.cpp main.cpp

HEADERS += appwindow.h

库编译结束后就是可执行文件编译了，工程文件在app目录下的app.pro，内容如下：

TEMPLATE = app

TARGET = app

DESTDIR = ../bin

INCLUDEPATH += . ../include

DEPENDPATH += .

LIBS += -L../lib -lbase

SOURCES += appwindow.cpp main.cpp

HEADERS += appwindow.h

如果要编译多个库，多个可执行文件，或者多级目录工程，可以参照这个例子写多个pro文件，递归而已，不是什么难事。