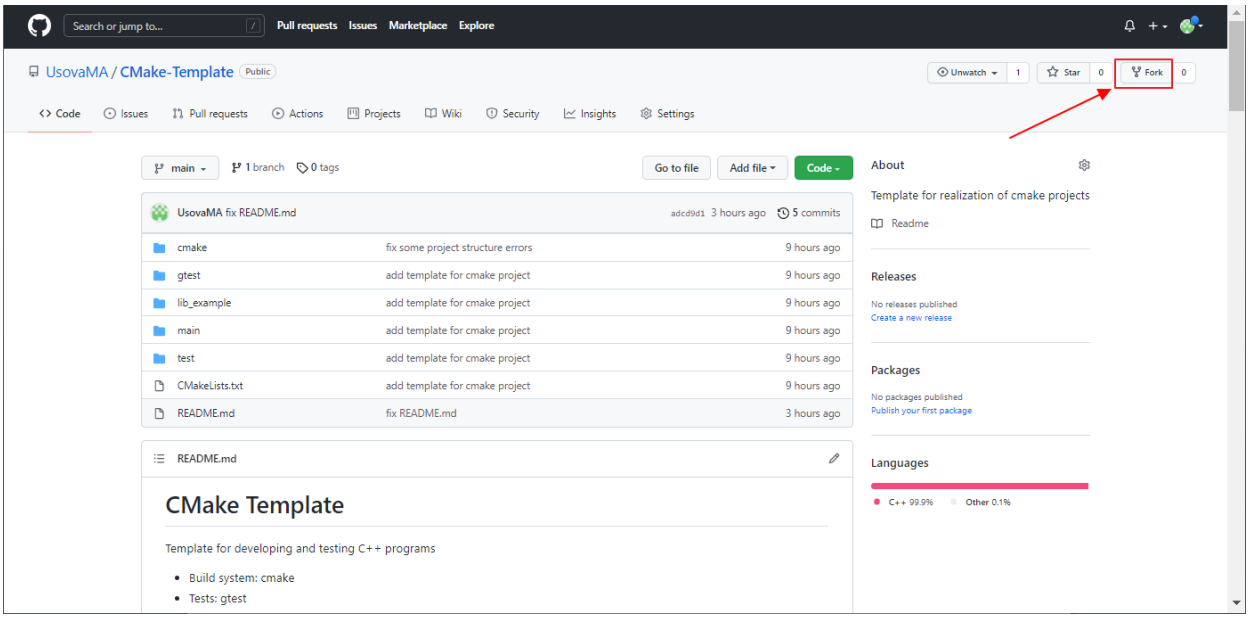
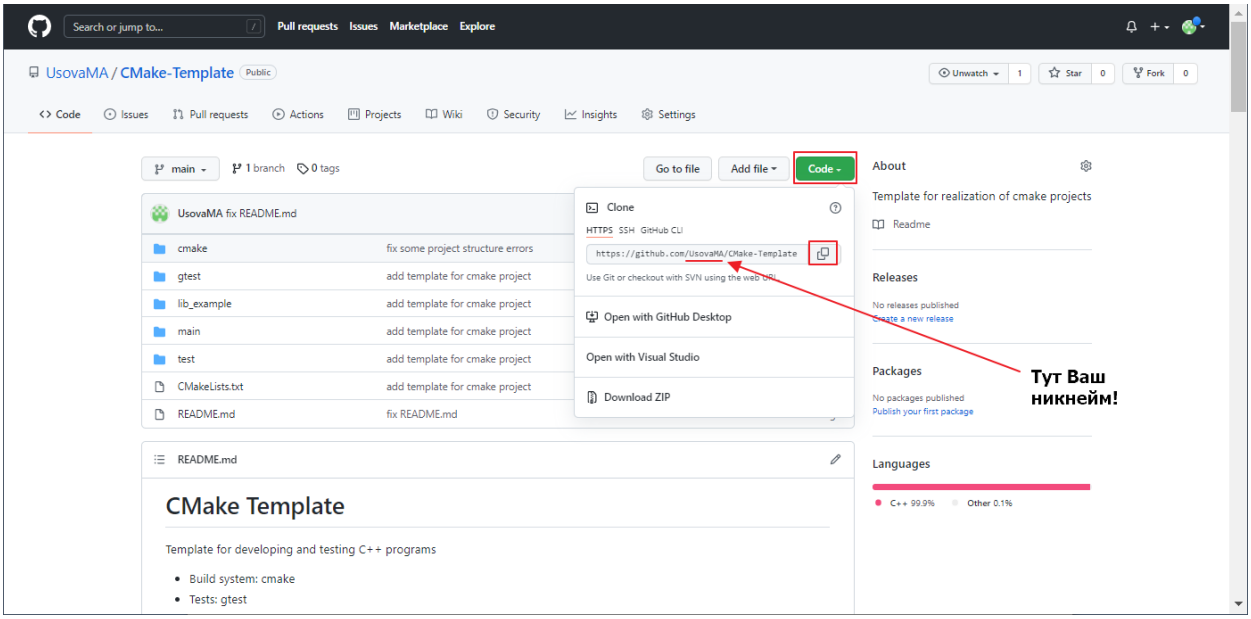
Инструкция по CMake

1) Сделать fork проекта на GitHub https://github.com/UsovaMA/CMake-Template.



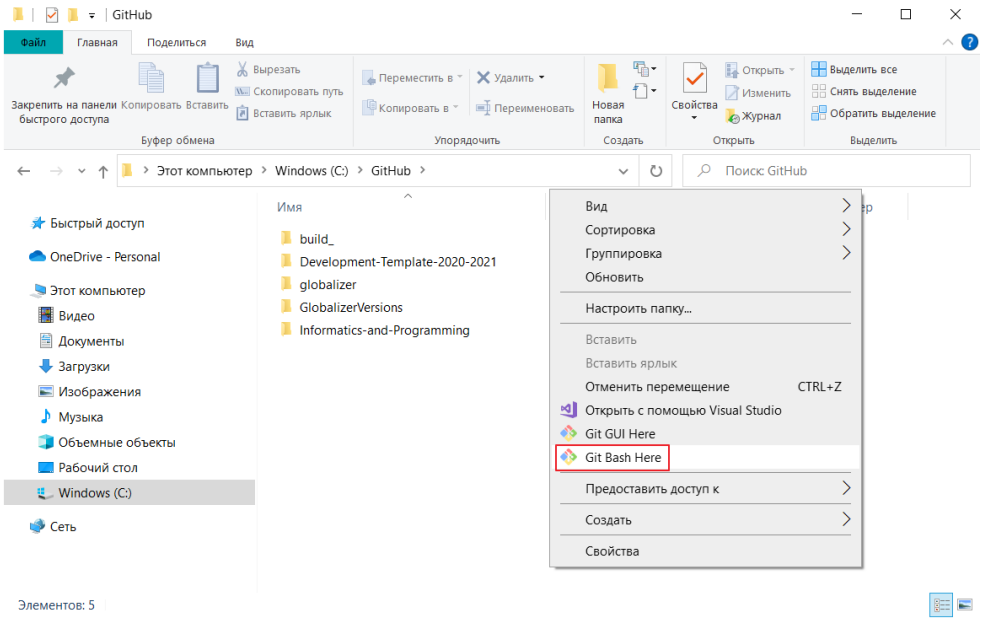
2) Клонировать свой fork проекта на компьютер (назовём его локальный репозиторий, а GitHub – удалённый репозиторий).
   a. Получить ссылку на Ваш удалённый репозиторий

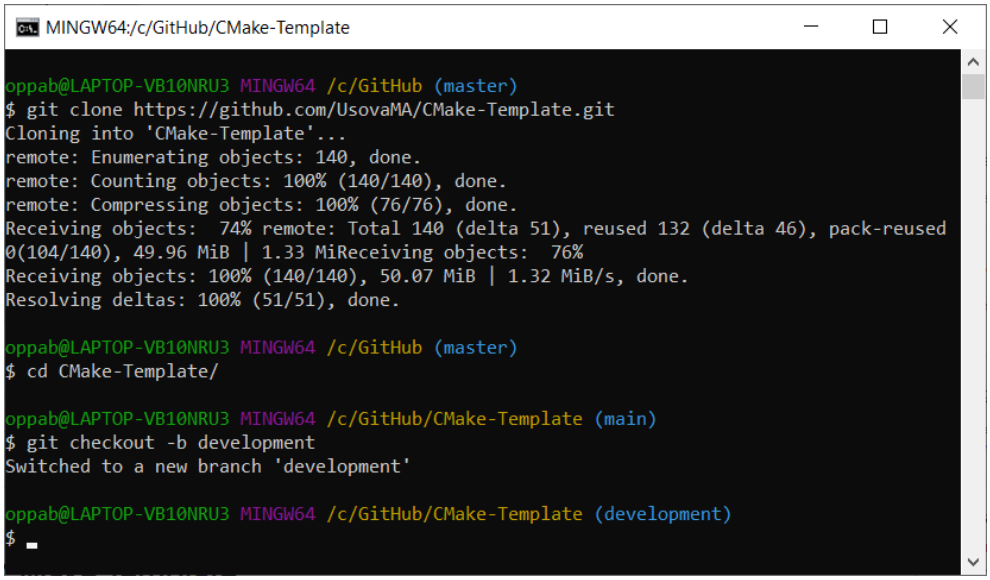


   b. С помощью командной строки клонировать удалённый репозиторий, перейти в создававшуюся папку проекта с помощью команды cd, сразу переключиться в ветку разработки, чтобы не забыть и не начать случайно делать изменения в векте main/master. См. рисунки ниже с примером выполнения данных команд.

Замечание. Выберете для своих GitHub-проектов правильное место на своём компьютере. В пути к папкам не должно быть пробелов, русских букв. Старайтесь привыкнуть давать названия только на английском и с использованием знаков – и _. Примеры:

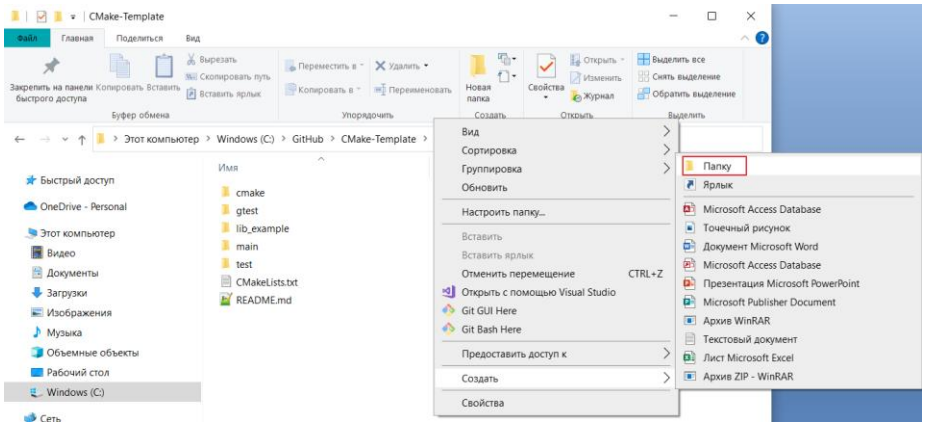
- хорошо – CMake-Template, CMakeTemplate, CMake_Template,
- плохо – CMake Template, CMake-шаблон.





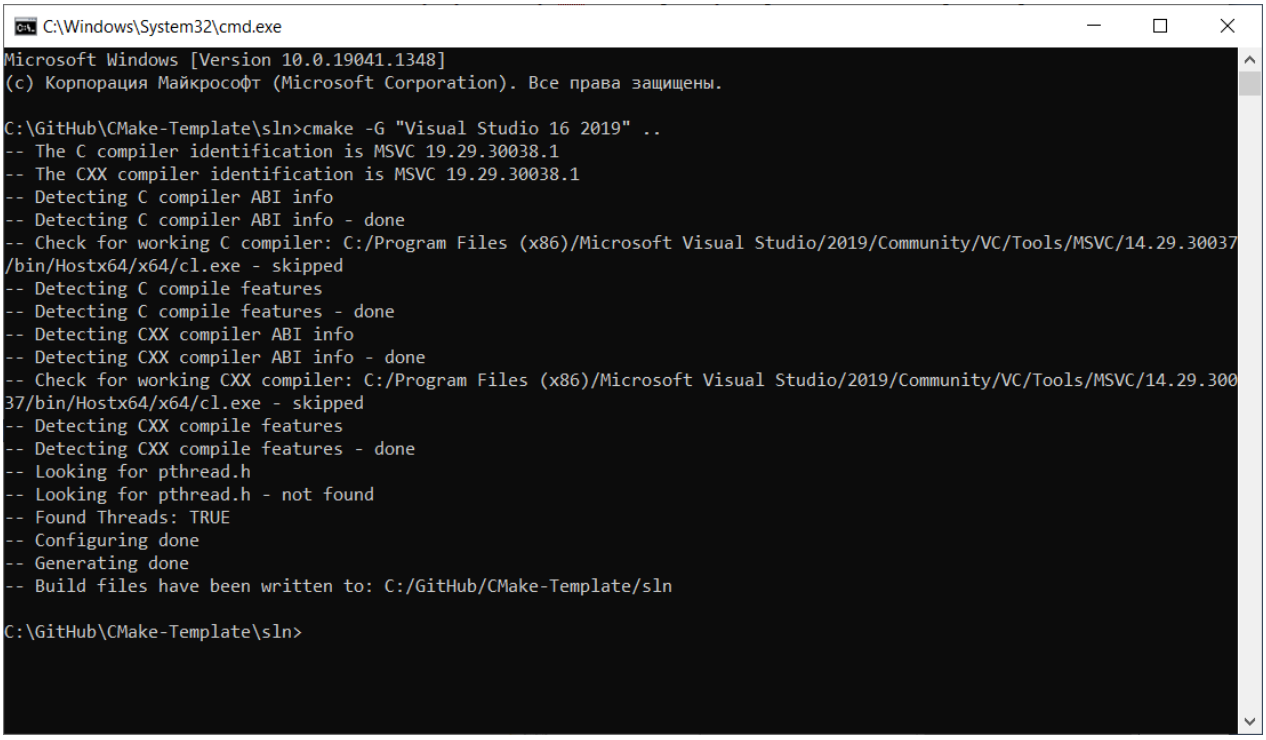
Более подробно о командах, вариантах работы с системой git, часто встречаемых ошибках и проблемах читать в описании проекта (файл README.md) - https://github.com/UsovaMA/CMake-Template/blob/main/README.md.

3) Попробовать собрать шаблон таким, какой он есть
   a. Создайте пустую папку sln, в которой будет располагаться собранное решение.



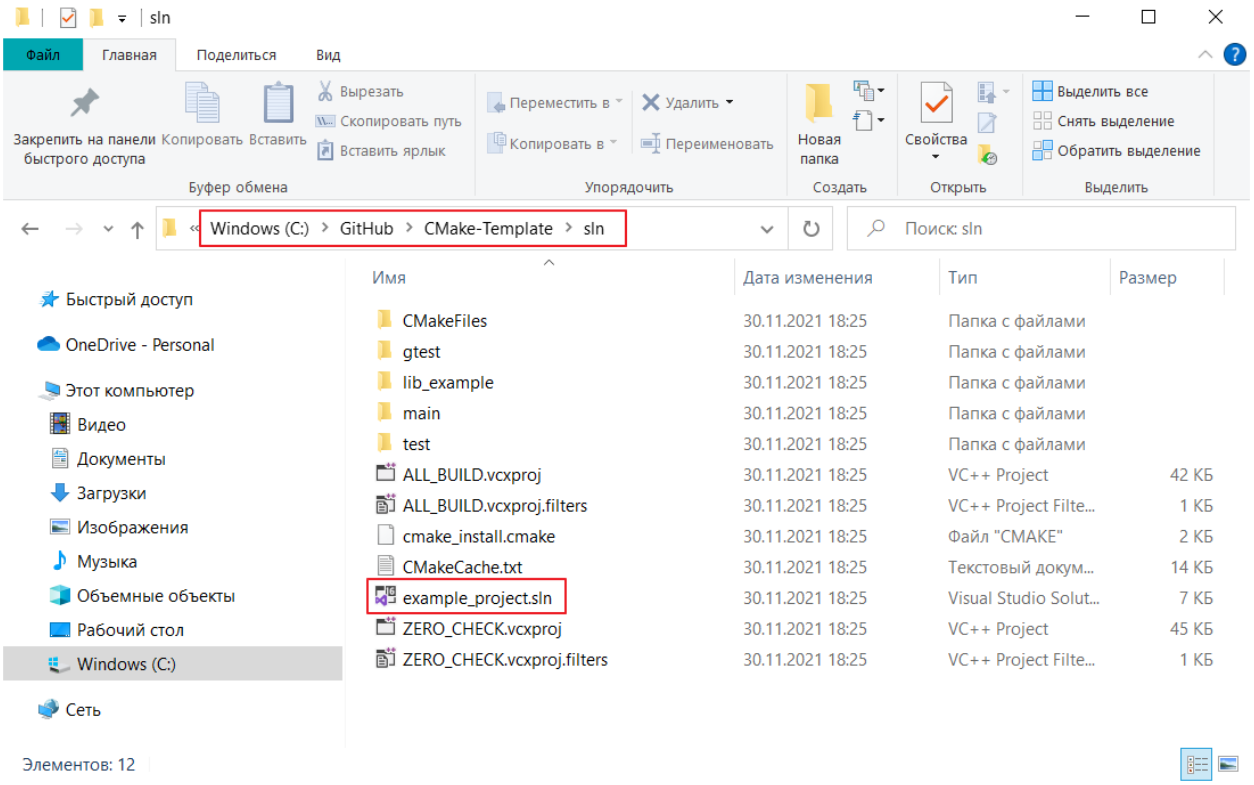
   b. Из данной папки вызовете командную строку (вводим cmd вместо пути).

c. Далее вводим команду для сборки проекта **cmake -G "Visual Studio 16 2019" ..** В команде указываем версию студии, которая у вас установлена.



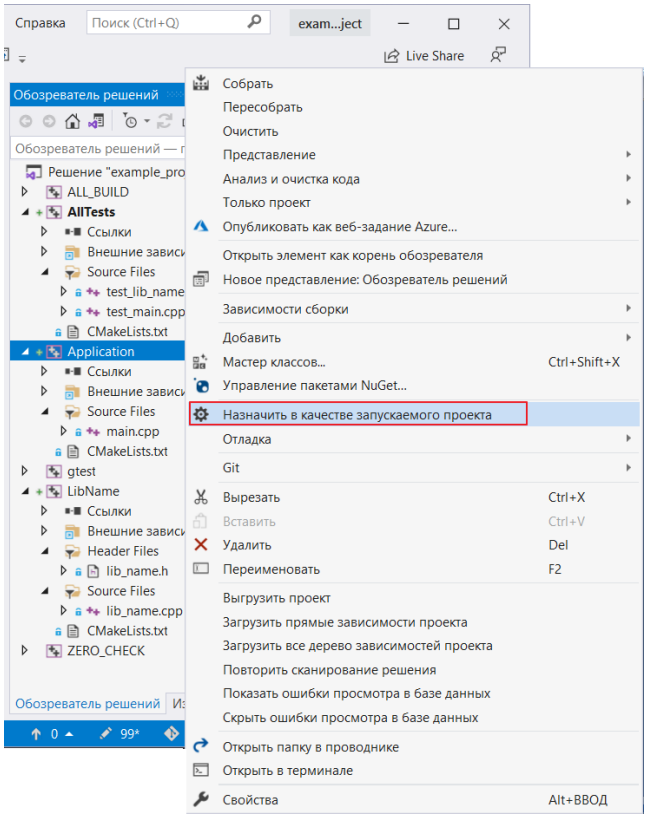
Смотрим на выведенную информацию – ошибок быть не должно.

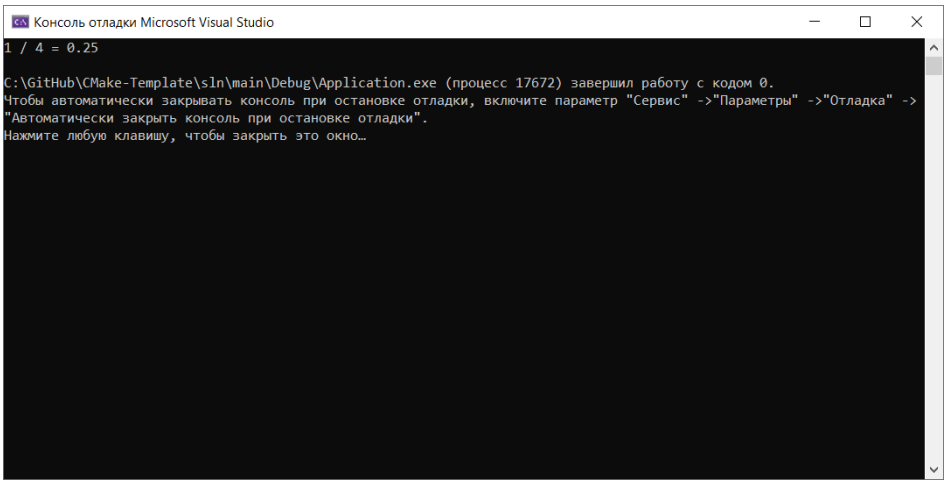
d. Проверяем, содержимое папки sln, там должен появиться собранный проект, запускаем .sln файл.

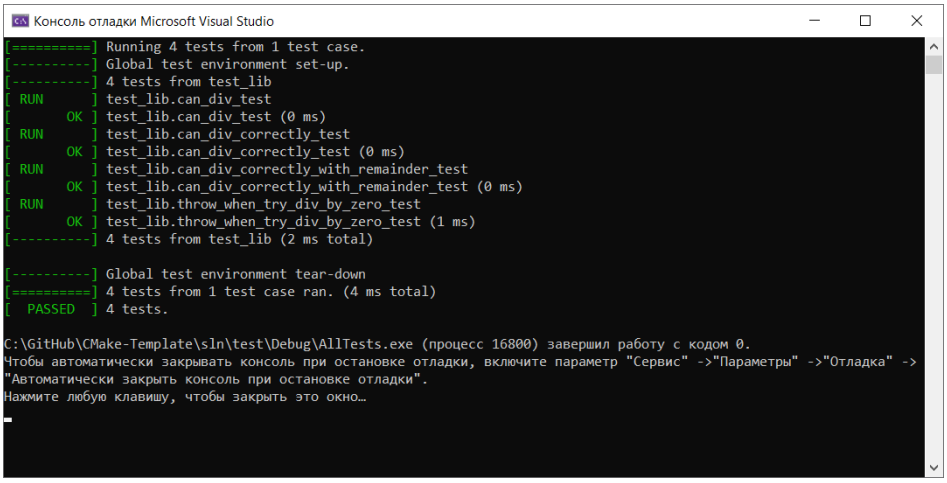


e. Назначаем автозапускаемыми проектами поочередно LibName и AllTests, смотрим что всё корректно работает. Результаты удачной отработки проекта-шаблона представлены на рисунках ниже.

Указания. Правая кнопка мыши по проекту -> Назначить автозапускаемым проектом. Автозапускаемый проект выделяется жирным шрифтом.

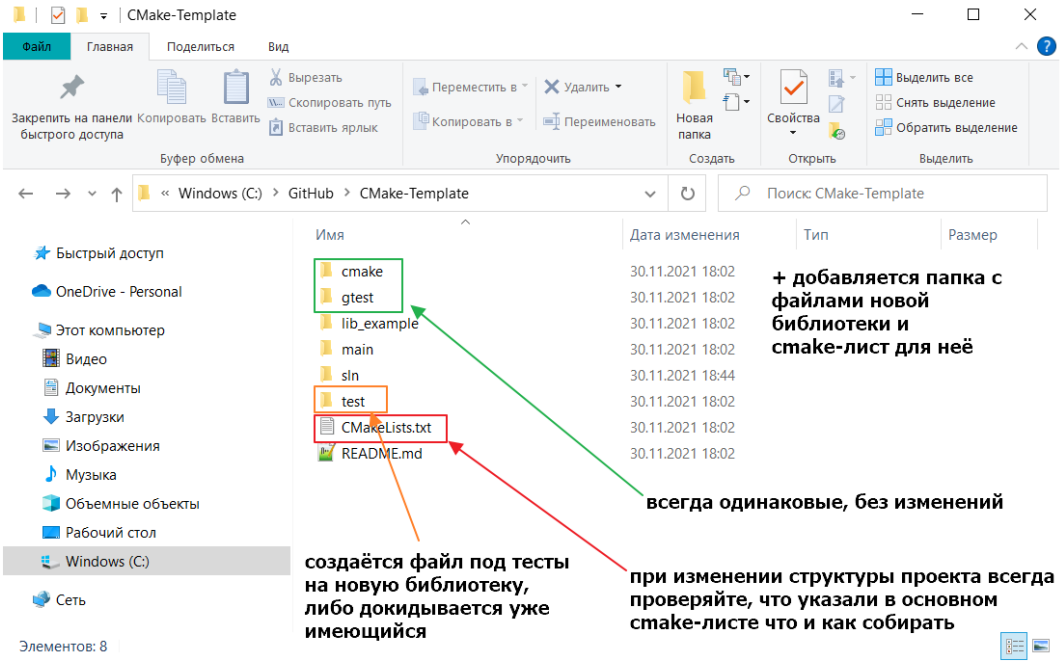






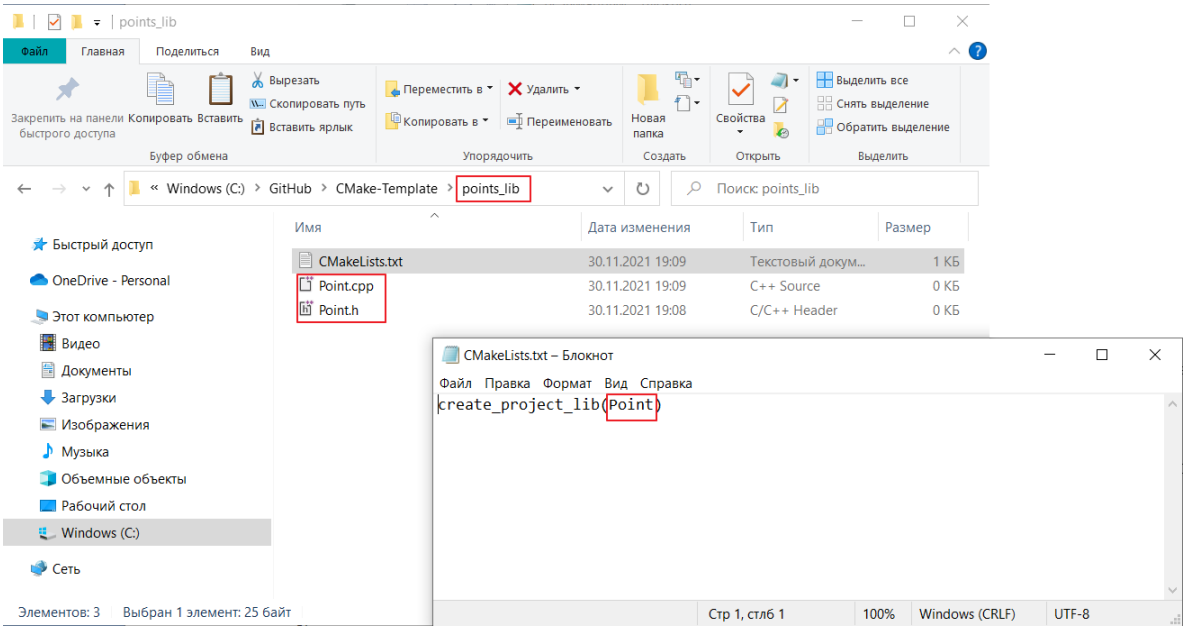
4) После учебной сборки поправить шаблон под свой проект. Я привожу пример с добавлением **ещё одной** библиотеки в проект. Помимо уже имеющейся! Шаблонную статическую библиотеку я оставлю. Убрать её из проекта можно в любое время, но пусть она будет перед глазами как пример оформления.

Замечание. Если вы хотите снести шаблонную библиотеку и вместо неё сделать свою, то достаточно переименовывания соответствующих файлов и названий в cmake-листах, чтобы сделать рабочую заготовку.

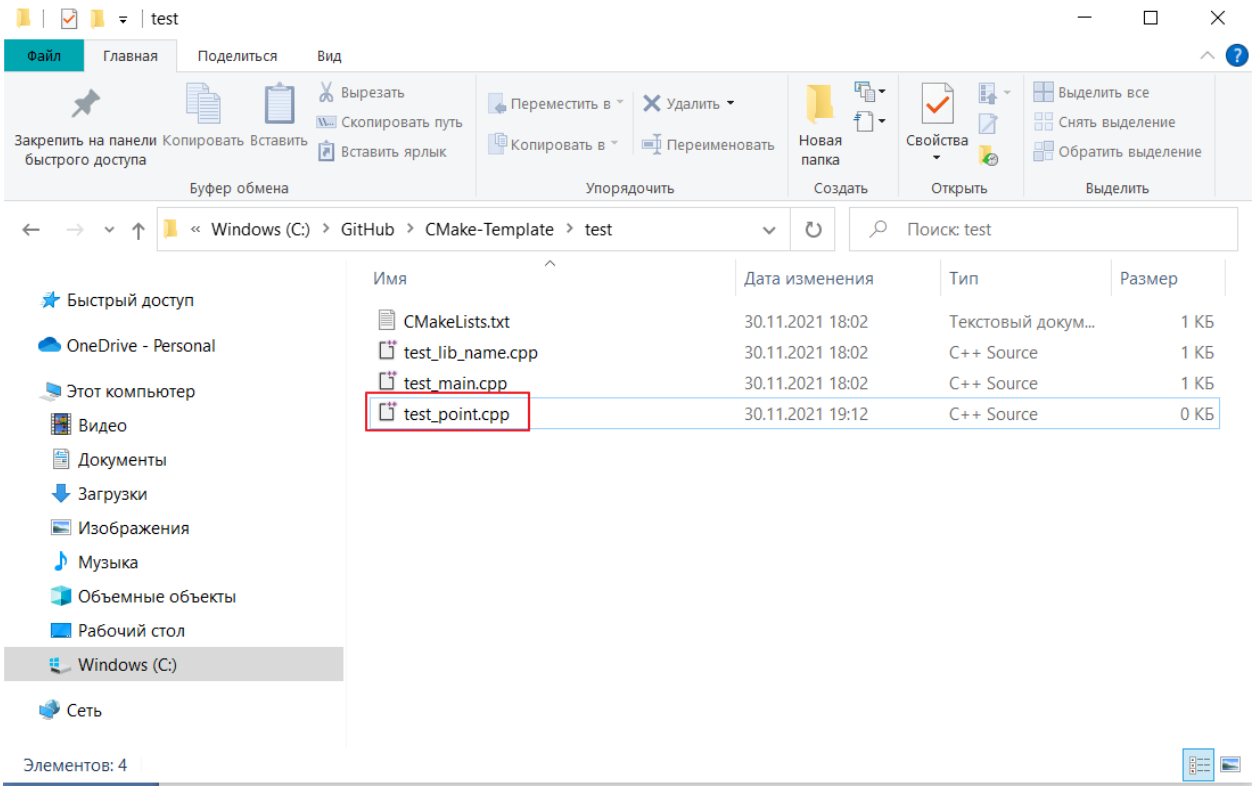


Подробно действия, которые можно/нужно проделать:

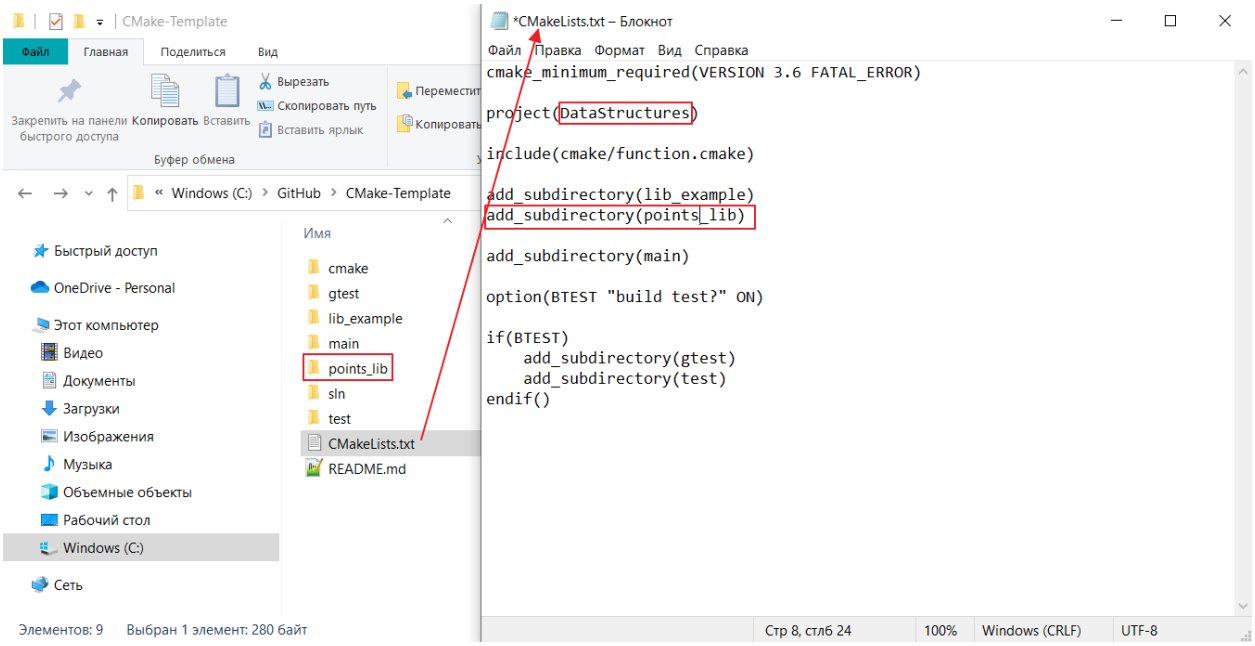
- По необходимости меняем названия в cmake-листах, названия вы определяете сами, сделайте свой наиболее понятный и удобный лично Вам шаблон;
- Добавляем папку под новую библиотеку (points_lib), размещаем в ней файлы (уже существующие или пустые/новые), создаём cmake-лист (вызываем функцию создания статической библиотеки)



- Создать файл для тестов:

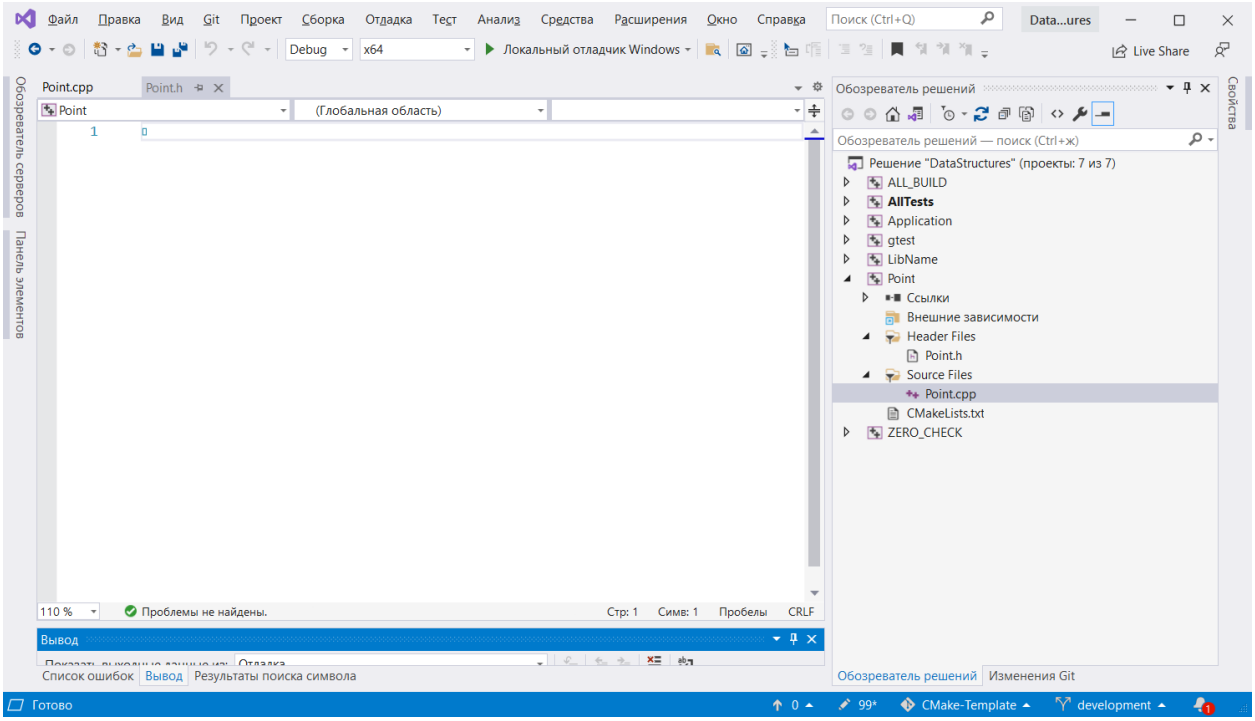


- Изменить основной cmake-лист:



- Под каждую библиотеку можно свой main-проект, а можно использовать всегда один и всё в нём дописывать по мере наращивания числа библиотек в проекте, либо адаптировать его только под последнюю задачу (то есть содержимое main всегда демонтрирует работу только с одной из библиотек, например, последней разрабатываемой). Рекомендовано попытаться сделать самостоятельно вариант с отдельным приложением для каждой отдельной статической библиотеки.

Далее очищаем папку sln, пересобираем проект, запускаем файл .sln, проверяем работоспособность. Должна получиться следующая структура.



Затем необходимо реализовать поставленную задачу и залить изменения на GitHub. Инструкции об этом также есть в README.md к данному проекту-шаблону.

Коротко: добавить файлы для коммита, сформировать коммит, сделать пуш в ваш репозиторий в рабочую ветку. **ВАЖНОЕ ЗАМЕЧАНИЕ: папку sln не заливаем!!! В реальности эта папка может весить очень много, ваш проект с cmake любой другой пользователь вашего кода соберёт за 10 секунд, поэтому данная сборочная информация лишняя к выкладыванию.**