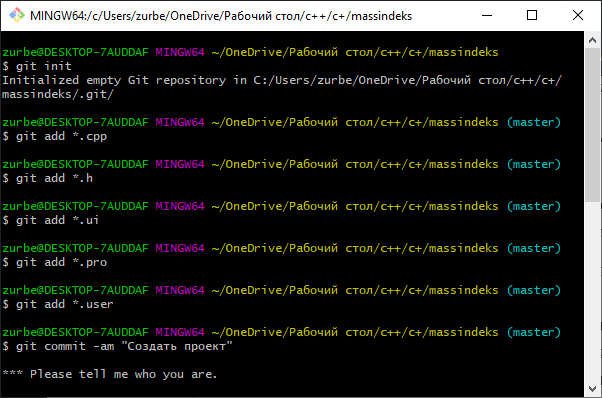
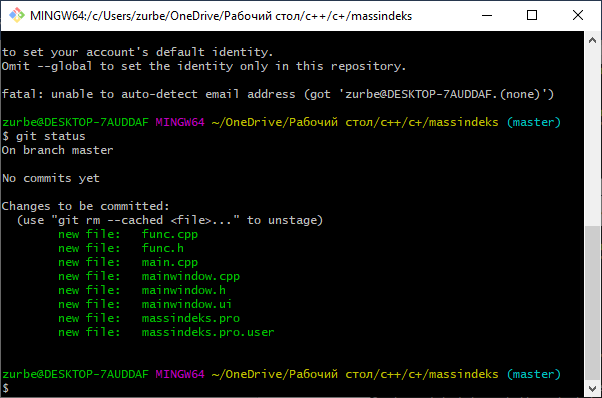
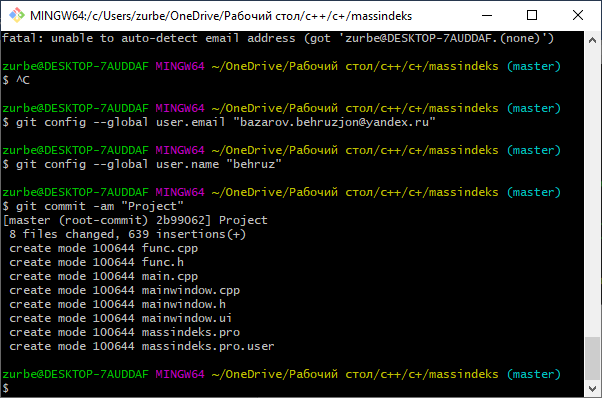
1,2.



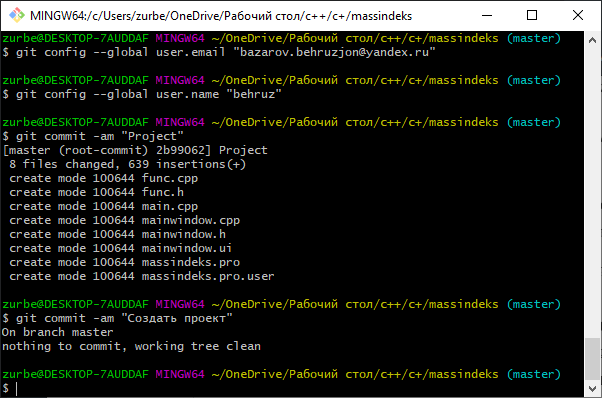
3.



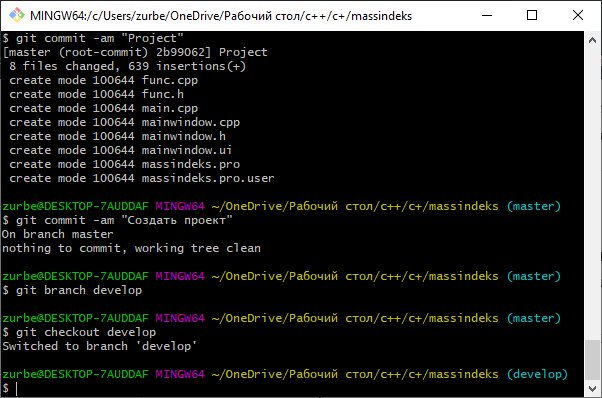
4.



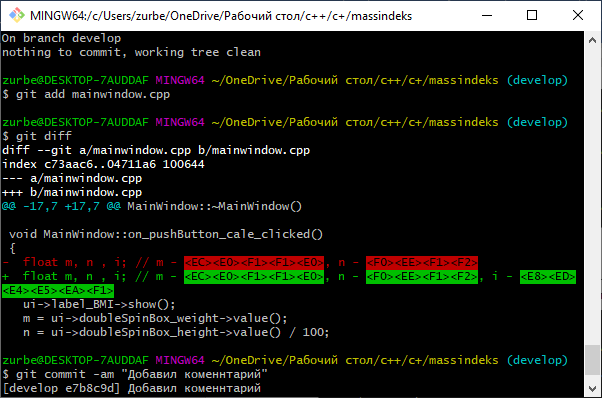
5,6.



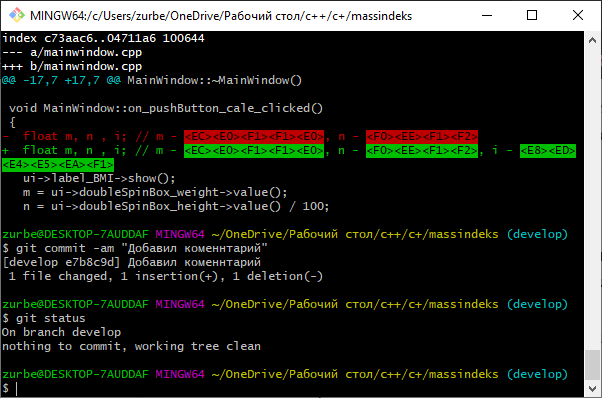
7,8.



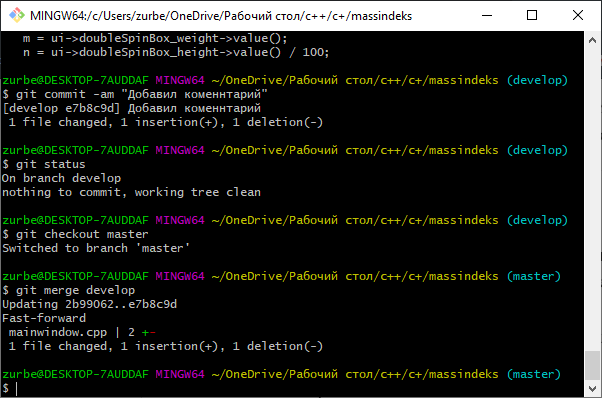
9.



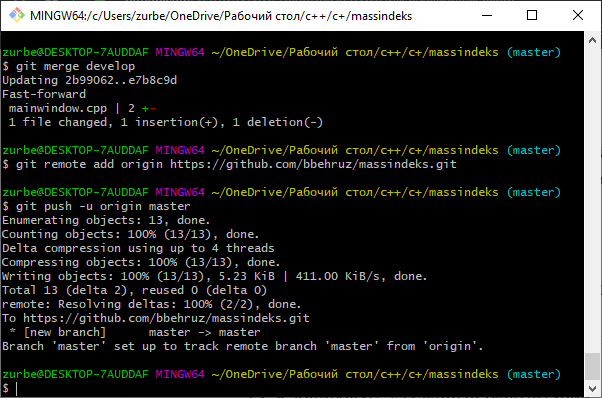
10.

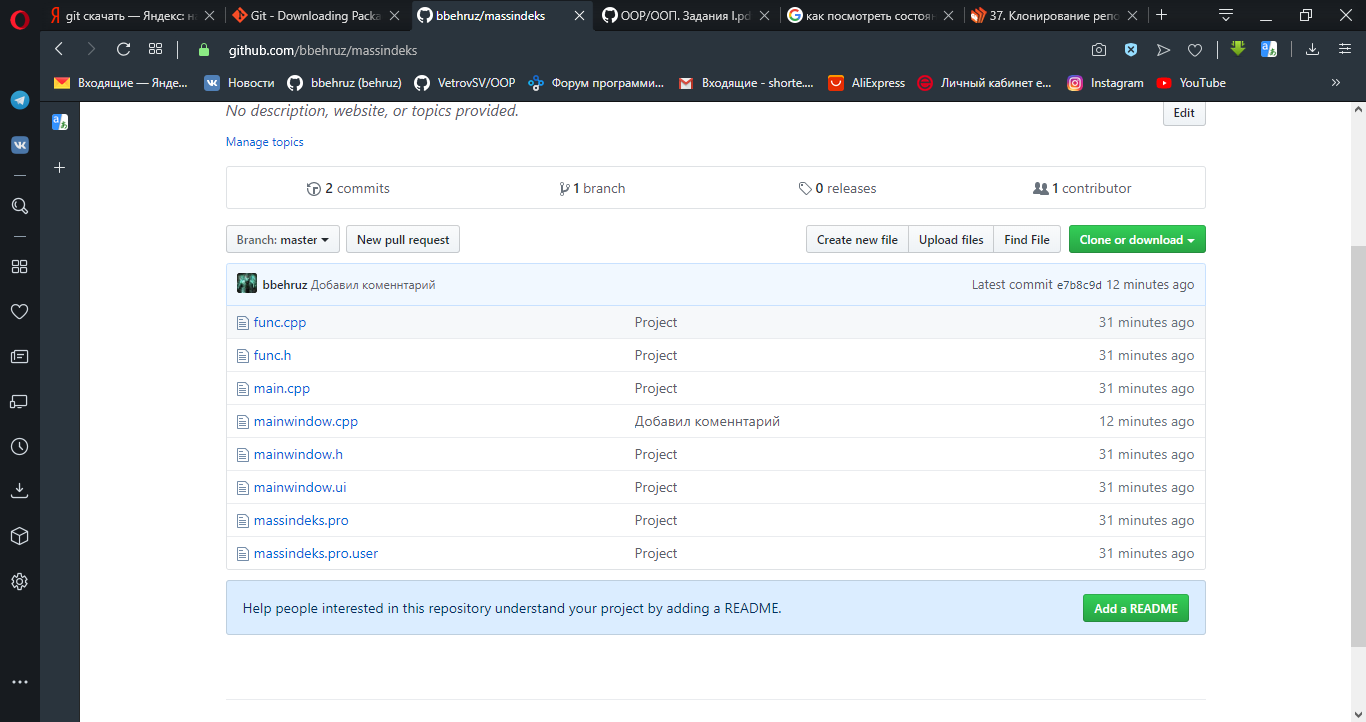


11, 12.

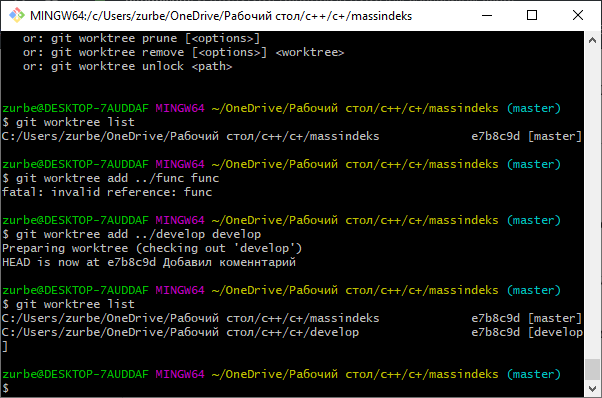


13, 14.

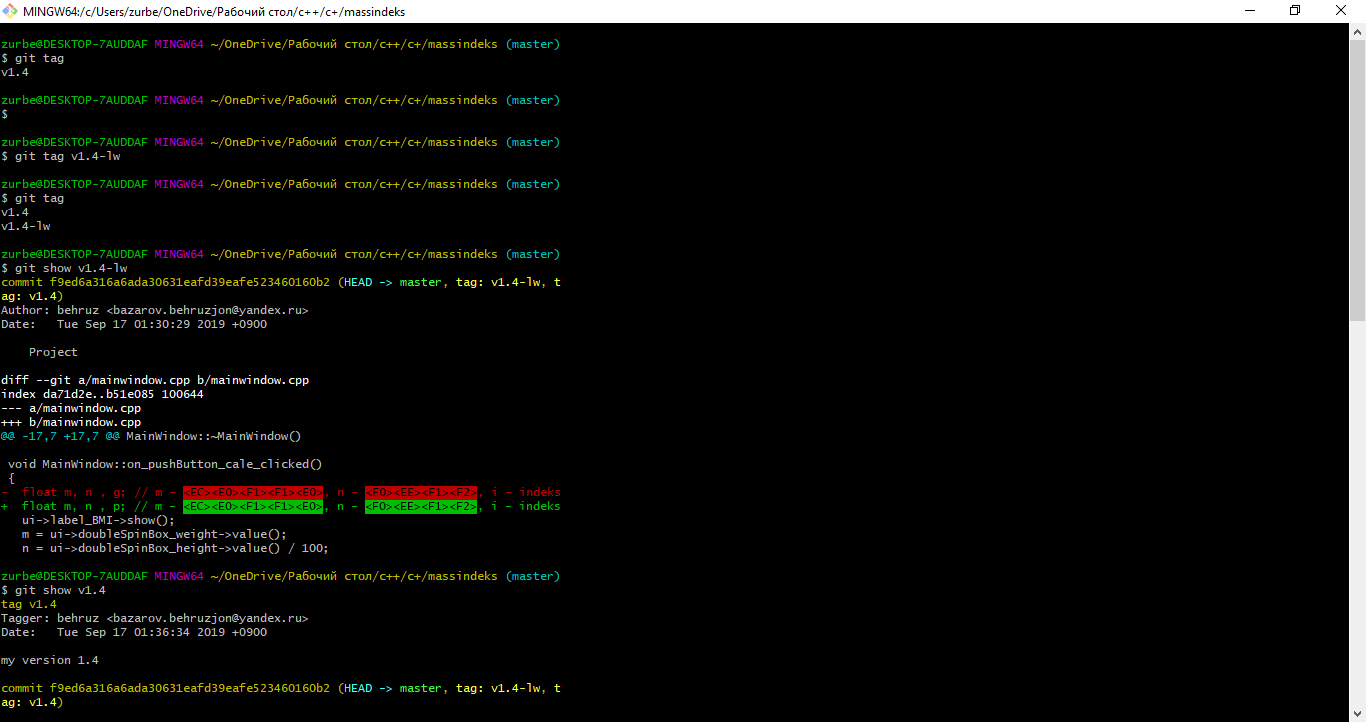




15.1.



15.3.



Вопросы:

1. Что такое система управления версиями? Для чего она используется?

Система управления версиями (Version Control System, VCS) – программное обеспечение для облегчения работы с изменяющейся информацией.

Позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

1. Что такое репозиторий (локальный и удаленный)?

Репозиторий, хранилище – место, где хранятся и поддерживаются какие-либо данные.

Локальный репозиторий позволяет легко отслеживать изменения, хранить несколько версий программы. Всегда есть возможность вернуться к предыдущей версии.

Удаленный репозиторий помимо этого делает удобной групповую разработку: разработчик отправляет свои изменения в общй репозиторий и забирают из него изменения сделанные другими разработчиками.

1. Какие файлы следует добавлять к отслеживанию, а какие нет?

Как правило в отслеживании должны быть файлы исходных кодов и другие файлы, необходимые для компиляции и запуска программы. Исполняемые файлы не отслеживаются. Потому, что их всегда можно получить после компиляции и чтобы не засорять ими репозитарий.

1. В каких случаях создавать ветку?

Если планируется обширные изменения, то стоит подумать над созданием отдельной ветви, чтобы параллельно существовало исходная версия программы и версия, в которую вносятся изменения – рабочая версия.

Допускается, что рабочая версия может не транслироваться, работать с ошибками. Однако в любой момент должна быть возможность вернуться к исправной версии программы.

1. Что такое конфликт? Как исправить?

Если вы изменили одну и ту же часть одного и того же файла по-разному в двух объединяемых ветках, Git не сможет их чисто объединить. Если ваше исправление ошибки #53 потребовало изменить ту же часть файла, что и hotfix, вы получите примерно такое сообщение о конфликте слияния:

$ git merge iss53

Auto-merging index.html

CONFLICT (content): Merge conflict in index.html

Automatic merge failed; fix conflicts and then commit the result.

Git не создал коммит слияния автоматически. Он остановил процесс до тех пор, пока вы не разрешите конфликт.

Чтобы разрешить конфликт, придется выбрать одну из сторон.

Разрешив каждый конфликт во всех файлах, запустите git add для каждого файла, чтобы отметить конфликт как решенный. Подготовка (staging) файла помечает его для Git как разрешенный конфликт.