Лабораторная работа № 6  
«Технология разработки программного обеспечения»

Выполнили студенты группы 19-КБ-ПР1

Воропай Александр и Кравцов Олег

**«Использование систем контроля версий исходного кода программ»**

Цель работы: создать в системе контроля версий репозиторий для нового проекта и выполнить все основные действия с исходным кодом программы, связанные с контролем версий.

Введение: в процессе выполнения лабораторной работы мы желаем научиться создавать репозиторий, работать с ним и контролировать версии наших программных продуктов.

**Программно-аппаратные средства, используемые при выполнении работы:**

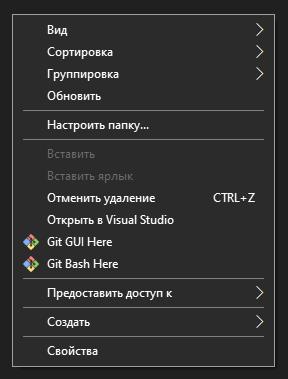
-Microsoft Word.  
-Git version 2.28.0

Лабораторная работа выполнялась на компьютере со следующими характеристиками:

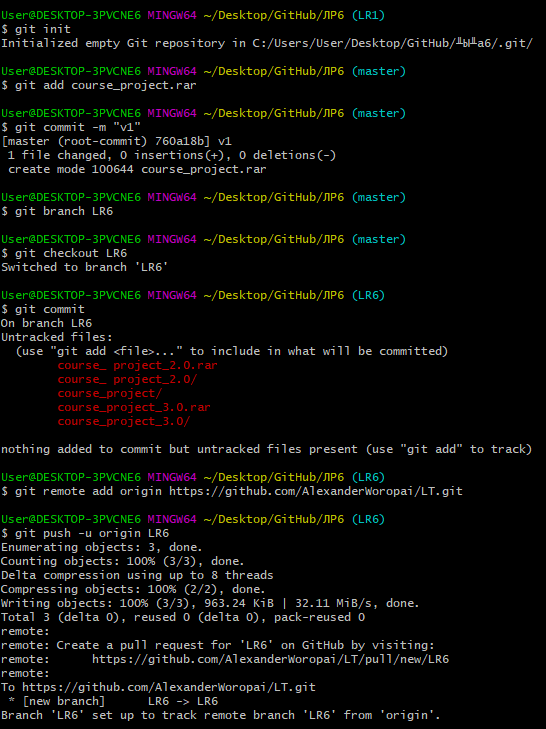
* Intel Core i7-8550U CPU
* Видеокарта - NVidia GEFORCE GTX 1050 Max-Q
* ОЗУ – 16 гб DDR4
* OC – Windows 10

**Этапы выполнения лабораторной работы**

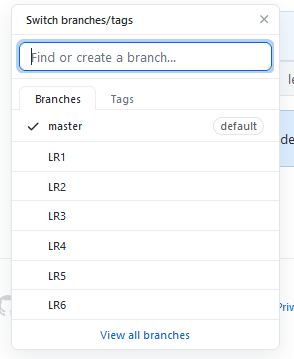
1. Заходим в папку с нашим проектом, кликаем в свободное пространство правой кнопкой мыши и выбираем “Git Bash Here”



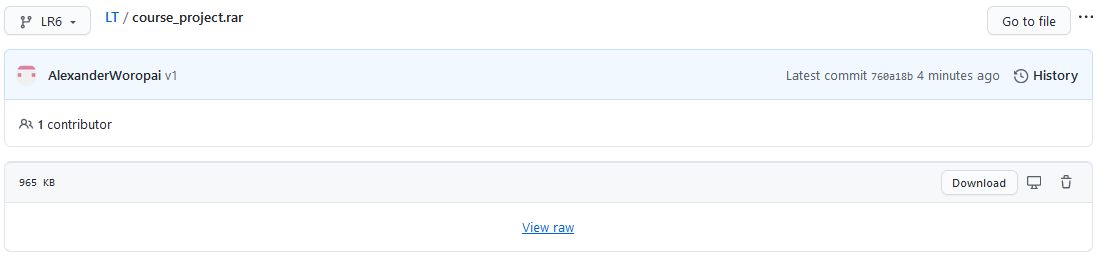
1. Запустится консольное окно, с помощью которого мы можем обмениваться информацией с нашим локальным и онлайн репозиториями. На данном шаге мы инициализируем репозиторий (git init), добавляем наш проект (git add < имя\_файла.расширение >), создаем для него коммит (git commit –m “any\_text”), создаем ветку LR6 (git branch LR6), переходим на нее (git checkout LR6) и впредь производим наши изменения именно в этом репозитории (git commit). Дальше мы подключаем наш онлайн репозиторий (git remote add origin <ссылка\_на\_репозиторий>) и выгружаем наши файлы на него (git push –u origin LR6)



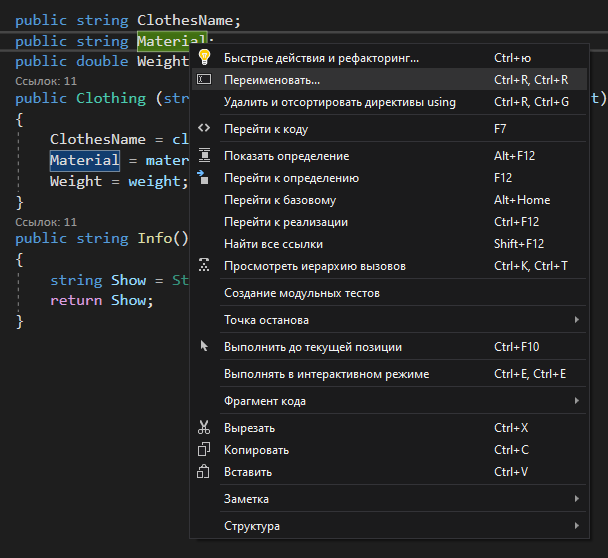
1. Новая ветка LR6 создалась успешна, переходим в нее и видим наш файл. Удаляем наш проект с компьютера.



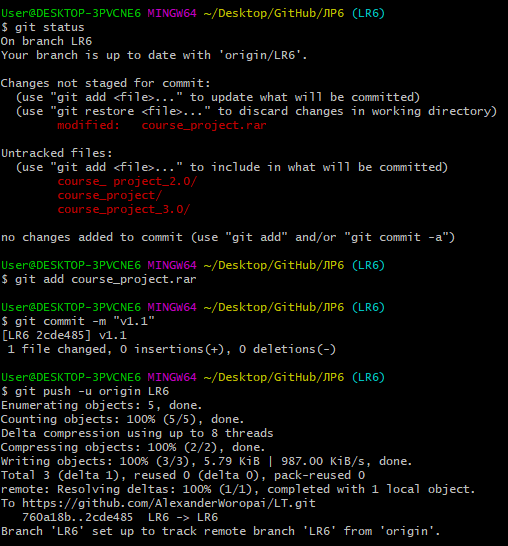
1. Загружаем наш проект нажатием на кнопку “View raw”



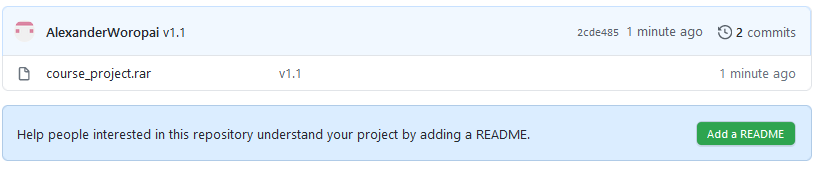
1. Открываем проект и производим любые изменения (я переименовал одну из переменных)



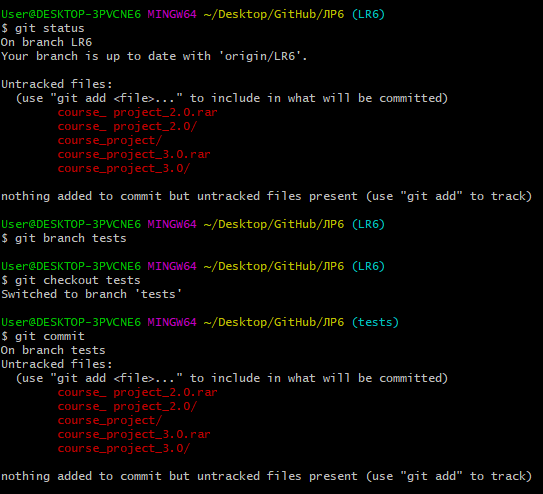
1. При вводе команды “git status” мы увидим что система заметила, что наш файл был изменен и будет требовать дальнейших указаний (вернуться на старую версию или обновить версию на наших репозиториях). Вводим команду “git add < имя\_файла.расширение >”, после “git commit -m “any\_text””. После выгружаем наш проект на онлайн репозиторий: “git push –u origin LR6”.



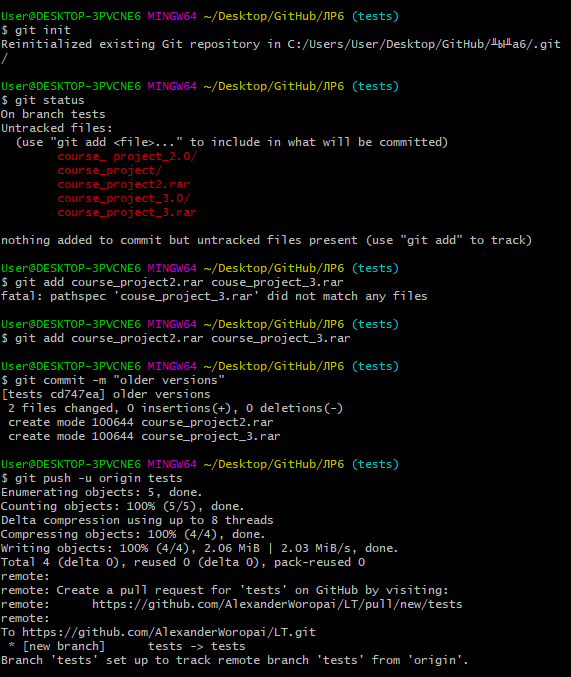
1. Файл на онлайн репозитории был изменен, что можно заметить по версии проекта (v1.1).



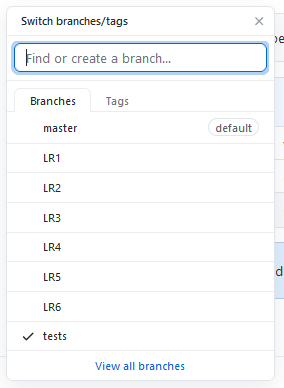
1. Создаем ветку “tests” (git branch tests), переходим в эту ветку (git checkout tests) и производим наши изменения на текущей ветке (git commit).



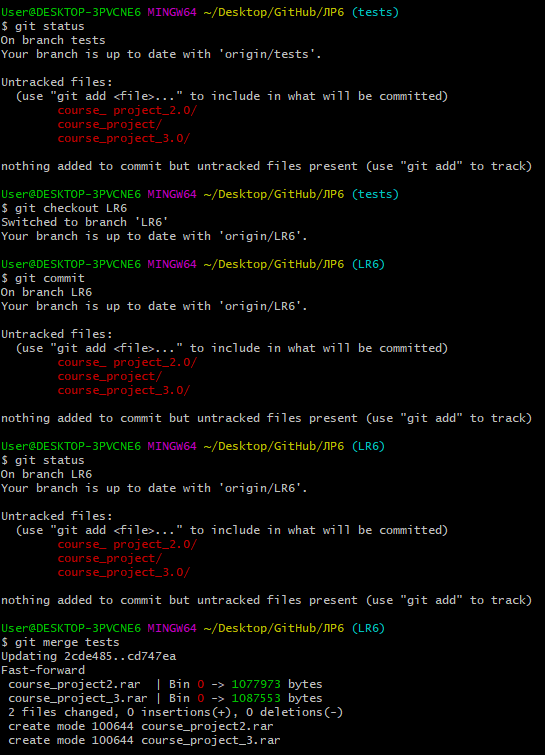
1. Добавляем наши измененные проекты (git add <имя\_файла.расширение>) и также выгружаем их.



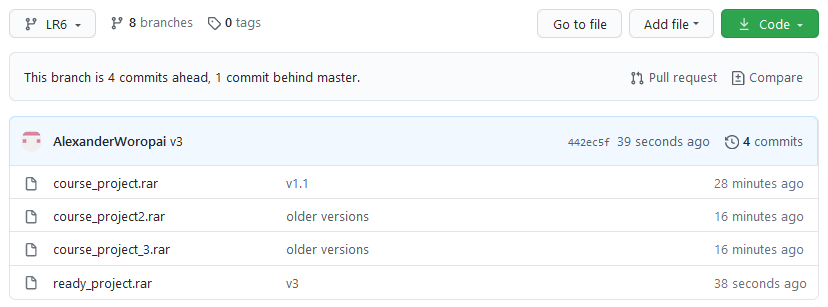
1. В списке появилась новая ветка “tests” на которую были выгружены наши проекты.

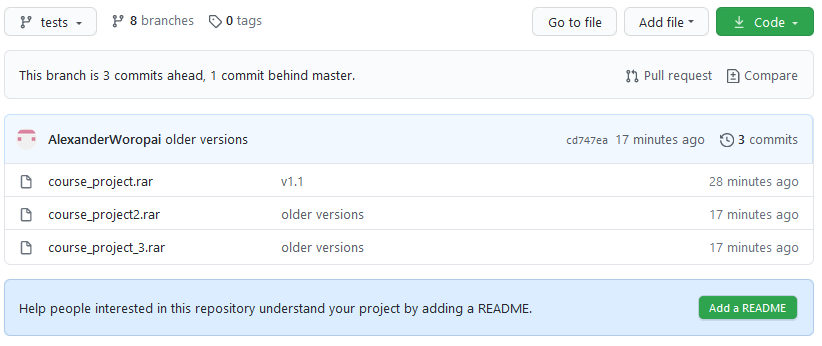


1. Переходим в предыдущий репозиторий, выбираем его для изменения и вводим команду для объединения веток (git merge <имя\_ветки>).



1. Объединение произошло успешно.





Заключение: в процессе выполнения лабораторной мы научились создавать репозиторий, работать с ним и контролировать версии наших программных продуктов.

**Рекомендуемая литература**

1. Орлов С.А. Технологии разработки программного обеспечения: Учебник. – СПб.: Питер, 2002. – 464 с (наличие в библ ТУСУР – 26 экз.).

2. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: учебник. – М.: Финансы и статистика, 2002. (наличие в библ ТУСУР – 36 экз.).

3. Ларман К. Применение UML и шаблонов проектирования: Введение в объектно-ориентированный анализ и проектирование: Учебное пособие: Пер. с англ. - М.: Вильямс, 2001. - 496 с.

4. Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. Управление программными проектами. Достижение оптимального качества при минимуме затрат.: Персона.—М.: Издательский дом «Вильямс», 2004г. – 1136с 69

5. Ф. Брукс. Мифический человеко-месяц, или как создаются программные системы – Символ-Плюс, 2006 – 304с.