ВСР задание 2.1

Задание выполняется после проведения вебинара *«*Использование системы контроля версий Git*».*

На примере основных действий, продемонстрированных на вебинаре по Git, освоить работу (и продемонстрировать выполнение показанных на вебинаре действий) с одним из визуальных клиентов для работы с Git на выбор:

**Вариант 1.**Git – Git Bash

1. git init – команда, которая используется для создания пустого репозитория Git



1. ls –la – команда, которая используется для того, чтобы убедится, что репозиторий создан.



1. git status – команда, которая помогает увидеть статус репозитория, также можно посмотреть на какой ветке находимся.



1. touch readme.txt (touch + название файла) – команда, которая помогает создать текстовый файл в репозитории.



1. nano readme.txt (nano + название файла) – команда, которая позволяет ввести или изменить какой-нибудь текст внутрь файла. После введения данной функции, появится окно, в которое мы записываем необходимый текст, чтобы его сохранить зажимаем клавиши Ctrl+S, чтобы выйти Ctrl+X.



Для фиксации изменений используем команду git status.



1. git add readme.txt (git add + название файла) – команда помогает отслеживать файл. В пункте 5 файл еще не отслеживается, поэтому вводим данную функцию.



После вводим команду git status, что проверить новый статус



1. git commit -m «First file» (-m «комментарий»)– команда, которая фиксирует статус. Так как у нас нет пока больше файлов, ничего фиксироваться не будет.



1. Далее еще раз выполняем команду nano, чтобы изменить текст в файле.



1. После проверки статуса, замечаем, что появились изменения и их выделяют красным цветом. Далее используем команду add.



1. Добавляем файлы, можно с помощью обычного копирования файла. Добавить их можно с помощью команды git add ‘\*.txt’. По известной схеме проверяем статус файла.



1. Фиксируем с помощью команды commit и добавляем комментарий



1. git log (git log --summary) – git log команда, которая позволяет посмотреть историю всех фиксаций. git log --summary команда, которая помогает узнать более подробную информацию.





1. git remote add origin + ссылка на репозиторий – команда, которая позволяет добавлять удаленный репозиторий. Для этого необходимо зайти на сайт github, скопировать ссылку и вставить ее после рассматриваемой команды





1. git push -u origin main (git push –u + имя репозитория + название базовой ветки) – команда позволяет выгружать файлы в удаленный репозиторий.



1. git pull origin main – команда позволяет получить обратно файл, в котором были произведены изменения. В предпоследней строчке появилась информация о том, что прибавилось 2 строчки в репозитории.



1. Сначала изменим текст в файле, используя уже известную команду nano. Далее чтобы посмотреть, что именно изменилось воспользуемся командой git diff HEAD и посмотрим на изменения с версией, которая была зафиксирована. Плюс данной версии, что изменения показаны цветом.



1. Добавляя каталог, используем команду mkdir folder (mkdir + называние каталога). После добавиляем в каталоге файл через команду touch + название файла. Проверяем статус.





1. Команда git diff --staged позволяет сравнить состояние репозитория с тем, который был до этого момента.



1. git reset folder – команда позволяет отменить изменения. После выполнения команды опять набираем команду git diff –staged и видим, что изменений не было.



1. Чтобы вернуть наш файл в первоначальный вид, то используем команду git checkout -- readme.txt (git checkout -- + название файла). Проверяем статус файла.



А команда cat readme.txt (cat + название файла) помогут вывести изначальный текст в консоле.



1. Для того чтобы создать ветку используем команду git branch + название ветки. Далее чтобы посмотреть на какой ветке мы работаем выбираем команду git branch.



1. Чтобы переключится на другую ветку, набираем команду git checkout + название ветки и выводим название ветки на экран через git branch.



1. Команда, позволяющая удалить каталог и файлы этой ветки, выглядит следующим образом git rm -r



Необходимо после этого фиксировать изменения.



1. Чтобы выполнить команду слияния, вводим git merge + название ветки



1. git branch -d clean\_up – команда позволяет удалить ненужную ветку.



1. git push – команда позволяет выгрузить все изменения в удаленный репозиторий.



