Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

специальность 09.02.07 «Информационные системы и программирование»

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО МДК 04.02 «Обеспечение качества функционирования компьютерных систем»

|  |  |
| --- | --- |
| Выполнил студент  группы П50-5-21  Комков Сергей Александрович | Проверил преподаватель  \_\_\_\_\_\_\_\_\_\_\_ К.С. Образцова  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 года |

Москва 2024

Оглавление

[Практическая работа №1 3](#_Toc177260760)

[Практическая работа №2 4](#_Toc177260761)

[Практическая работа №3 5](#_Toc177260762)

[Практическая работа №4 6](#_Toc177260763)

[Практическая работа №5 7](#_Toc177260764)

[Практическая работа №6 8](#_Toc177260765)

Практическая работа №1

# Тема: Работа с файлами в локальном репозитории Git.

Цель работы: научиться работать с файлами в локальном репозитории.

Ход работы:

1. Конспект лекций

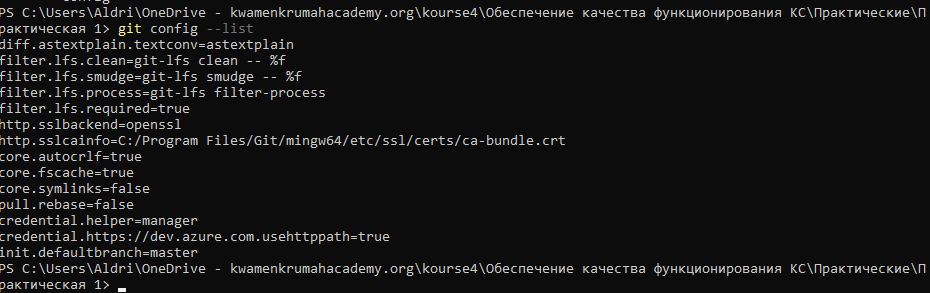
2. Продемонстрировать настройку и вывод конфигурации Git

3. Создание локального репозитория

4. Продемонстрировать работу основных команд для работы с файлами в репозитории

5. Описать работу с указателями HEAD, показать их разницу на реальном примере

Вывести конфигурацию



Изначально нет почты и имени пользователя, поэтому следующий шаг- добавление этих данных. Команды git config --global user.name "" и git config --global user.email "" для настройки имени и почты

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

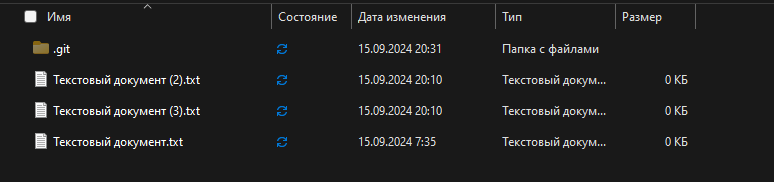
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Настраиваем кодировку



Для создания нового локального репозитория необходимо выполнить команду git init, которая инициализирует пустой репозиторий в текущей директории



Затем. Было установлено полезное расширение. После этого терминал показывает дополнительные параметры, помогающие работать с git.



Показано какие файлы сейчас новые,

~0: это число означает, что нет не отслеживаемых файлов.

Чтобы проверить какие файлы или папки есть репозитории, можно выполнить команду git status. Затем, чтобы добавить файл как отслеживаемый, надо выполнить команду git add.

Изображение выглядит как текст, программное обеспечение, снимок экрана

Автоматически созданное описание

Чтобы добавить в отслеживание все файлы в директории, надо написать команду “git add .”. Чтобы добавить новые изменения достаточно повторить команду. Команда git add не только добавляет новые файлы, но и все изменения файлов, которые были добавлены ранее., чтобы удалить файл из отслеживания существует команда “git rm "file.ex" --cached ”.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Если нужно добавить файлы из текущей директории и из всех поддиректориев, то надо использовать команду “git add –all”

Создание версии проекта

После того, как мы добавили нужные файлы в staging area мы можем создать версию проекта. С помощью команды “git commit -m "текст коммита"”Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Можно просмотреть изменения, которые были внесены последним коммитом “git show”

Изображение выглядит как текст, Шрифт, программное обеспечение, снимок экрана

Автоматически созданное описание

Для просмотра списка всех коммитов надо использовать команду “git log”

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Чтобы посмотреть конфигурацию, надо использовать команду “git config --list”

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

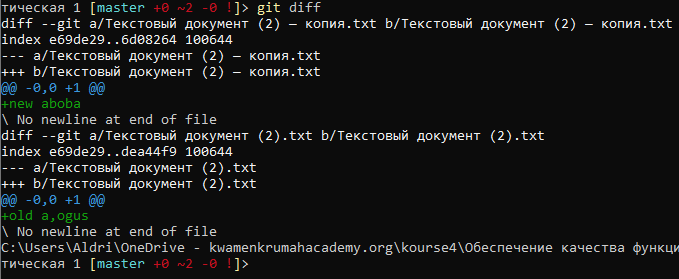
Чтобы создавать коммиты в notepad, надо выполнить команду “git config --global core.editor notepad”. Затем при сохранении wordpad, надо нажать сочетание клавиш “ctrl +s” и тогда коммит добавится



Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Позже были изменены файлы “Текстовый документ (2).txt” и ‘Текстовый документ (2) — копия”, а затем один был удалён для демонстрации работы команды “git diff”, которая показывает вывод разницы между текущим не отслеживаемым состоянием и последним снимком репозитория. У меня показалось только то, что я добавил, так как до этого фалы были пустые.



Также есть команда “git diff --staged”, которая осуществляет вывод разницы между текущим отслеживаемым состоянием и последним снимком репозитория.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Далее продемонстрирую

Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение, программное обеспечение

Автоматически созданное описание



После удаления содержимого можно ввести команду git restore, которая вернёт состояние файла, на последнюю зафиксированную версию

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание



Изображение выглядит как снимок экрана, текст, Мультимедийное программное обеспечение

Автоматически созданное описание

Чтобы посмотреть последние коммиты надо использовать команду: “ git log --oneline”.

Изображение выглядит как текст, Шрифт, снимок экрана, число

Автоматически созданное описание

Затем можно сместить указатель на предыдущий коммит: “ git reset --soft HEAD^”



В этом случае проект откатывается на предыдущий коммит, но изменения остаются в отслеживаемой зоне.

Вернуть проект ещё на один коммит назад и удалить все изменения. Команда: “git reset --hard HEAD^^”

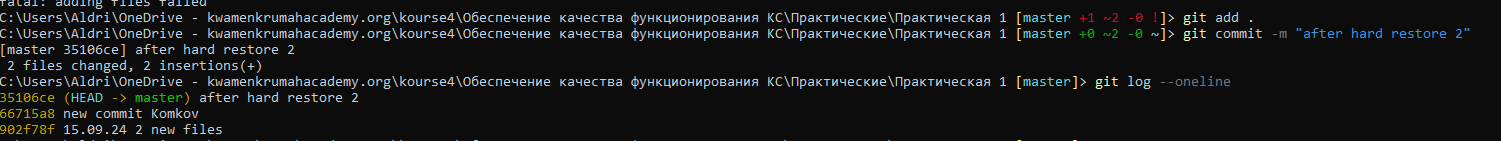


Теперь проект откатился на коммит 66715а8, а все изменения после этого коммита были безвозвратно удалены.



Осталось 2 коммита

Создать новый коммит



Выполнить команду “git reset --mixed HEAD^^”, чтобы откатиться на два коммита назад и перевести изменения в не отслеживаемую зону.



Теперь изменения остаются в файлах, но теперь они отображаются в git status как изменённые, но не добавленные для коммита.

Вывод:

Практическая работа №2

Цель работы:

Ход работы:

Вывод:

Практическая работа №3

Цель работы:

Ход работы:

Вывод:

Практическая работа №4

Цель работы:

Ход работы:

Вывод:

Практическая работа №5

Цель работы:

Ход работы:

Вывод:

Практическая работа №6

Цель работы:

Ход работы:

Вывод: