Homework\_Lesson 10

Индивидуальная часть:

1 Создать репозиторий и клонировать его себе локально. Настроить себе доступы для того, чтобы можно было пушить изменения в ветки.

Init commit может быть и локальный, это на ваше усмотрение.

Далее реализовать следующие сценарии

1. Создать 5-10 коммитов. (Пример: создать файлы, поменять их содержимое). Вывести их лог экран

2. С помощью reflog перейти к предыдущему коммиту (на ваше усмотрение). Output, screen, report. Вывести их лог экран

3. Создайте ветку с названием на ваше усмотрение (можно develop).

4. Создайте коммит и добавьте туда ещё дополнительные измения (добавьте, например в изменяемый файл точку, пробел и тд) с помощью ammend.

5. Сделайте коммит в main. Но не делайте git push (это важно!), сделайте изменения локально.

6. Сделайте так, чтобы этот коммит оказался в новой ветке с помощью git reset --hard.

7. Сделайте изменения в файле локально. Сделайте коммит для этого изменения.

8. Через git checkout отмените изменения в файле через откат по сохранённому хэшу.

9. Начните всё заново (это важно чтобы вы делали локально то, что я указал сделать локально). Можно использовать любой из подходов.

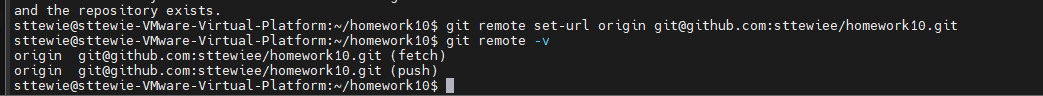
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как снимок экрана, текст, Шрифт

Автоматически созданное описание





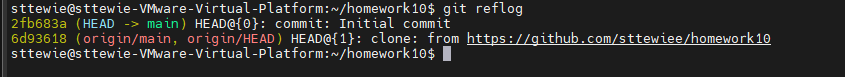


Изображение выглядит как снимок экрана, текст, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание



Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Задание по командам:

1. Создайте себе уютный чатик и придумайте там название своей команде

2. Возьмите скрипт (напр.: создание пользователей) и создайте для него фичи (см. дом. задание по bash). Распределите фичи между собой и, следуя gitflow, добавляйте их в develop-ветку. »»»»» Можно у нас будет build engineer (человек, который будет разгребать все комиты, мерджи и т.д. и мерджить это в мейн)- можно, даже нужно. Идеально - сделать 4 итерации в main и каждый раз build engineer будет разный (но это задание \*\*\*)

3. Создайте намеряно (sic!) конфликты для разных веток с фичами и решите их (только сделайте это так, чтобы в итоге получить рабочий скрипт). ДВУХ конфликта будет достаточно: первый необходимо решить локально, второй решить черз PR (pull request) в облаке. Для конфликта решённого локально необходимо добавить краткое описание конфликта и его решение в тело коммит сообщения (commit message).

4. Сделайте мердж в main через пуллреквест. Скрипт для выполнения домашнего задания \_\_\_\_\_\_\_\_\_\_\_\_ #!/bin/bash file1=$1 file2=$2 if cmp -s "$file1" "$file2"; then echo "Files are identical." else echo "Files are different." fi \_\_\_\_\_\_\_\_\_\_\_\_ \*\*\*For fun - создайте себе список фичей и задачи под них (working items). Ссылайтесь на них в pull request и code review, который будет в итоге для ветки main (пару комментов для галочки будет хорошо). Отчётом будет история коммитов в вашем репозитории. Отчёт в виде PDF или md-файла не требуется.

**Клонируем репозиторий команды и переходим в него**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Убедимся, что все последние изменения из удаленного репозитория загружены**

**Переключимся на ветку develop  
Убедимся, что локальная основная ветка актуальна  
Создам новую ветку для работы feature\_lapikov**  
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Добавил проверку на пустые параметры:**

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

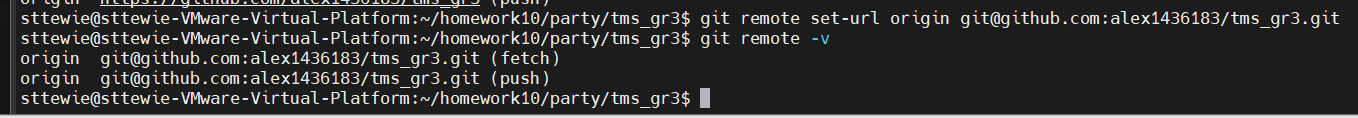
Автоматически созданное описание

**Проверка статуса и завиксируем изменения с описанием**

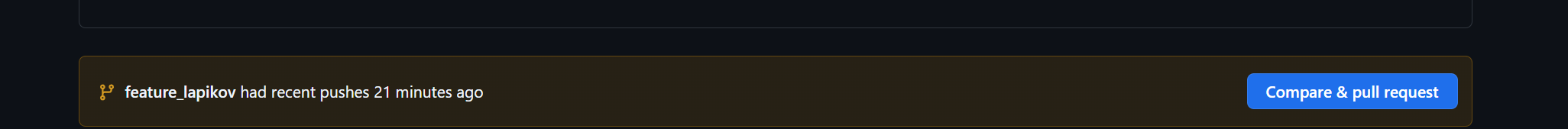
Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Изменим тип подключения, по ключу, чтобы запушить локальные изменения**



**Продолжим в веб и замержим изменения с веткой develop**



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

Автоматически созданное описание

**Командная работа**

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

Автоматически созданное описание