|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

Институт Информационных технологий (ИТ)

Кафедра Математического обеспечения и стандартизации информационных технологий (МОСИТ)

|  |
| --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 1** |
| **по дисциплине** |
| **«Технология разработки программных приложений»**  **Тема: «Системы контроля версий»** |

|  |  |
| --- | --- |
| Выполнил студент группы \_\_\_\_\_ИКБО-03-21\_ | Хречко С.В, |
| Принял | \_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |
| --- | --- | --- |
| Практическая работа выполнена | «\_\_»\_\_\_\_\_\_\_2023\_г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_2023\_ г. | *(подпись руководителя)* |

Москва 2023г.

Оглавление

[Часть первая 5](#_Toc128252251)

[Пункт 1 5](#_Toc128252252)

[Пункт 2 6](#_Toc128252253)

[Пункт 3 6](#_Toc128252254)

[Пункт 4 7](#_Toc128252255)

[Пункт 5 7](#_Toc128252256)

[Пункт 6 8](#_Toc128252257)

[Пункт 7 10](#_Toc128252258)

[Пункт 8 11](#_Toc128252259)

[Пункт 9 11](#_Toc128252260)

[Пункт 10 12](#_Toc128252261)

[Пункт 11 13](#_Toc128252262)

[Часть вторая 14](#_Toc128252263)

[Пункт 1 14](#_Toc128252264)

[Пункт 2 14](#_Toc128252265)

[Пункт 3 15](#_Toc128252266)

[Пункт 4 16](#_Toc128252267)

[Пункт 5 16](#_Toc128252268)

[Пункт 6 16](#_Toc128252269)

[Пункт 7 17](#_Toc128252270)

[Подпункт 1 17](#_Toc128252271)

[Подпункт 2 18](#_Toc128252272)

[Подпункт 3 18](#_Toc128252273)

[Подпункт 4 18](#_Toc128252274)

[Подпункт 5 19](#_Toc128252275)

[Подпункт 6 20](#_Toc128252276)

[Подпункт 7 20](#_Toc128252277)

[Часть третья 21](#_Toc128252278)

[Пункт 1 21](#_Toc128252279)

**Цель работы**: изучить структуру и функционал рассматриваемой информационной системы.

**Задачи:**

**Часть 1. Основные команды Git**

1. Установите и настройте клиент git на своей рабочей станции.

2. Создайте локальный репозиторий и добавьте в него несколько файлов.

3. Внесите изменения в один из файлов.

4. Проиндексируйте изменения и проверьте состояние.

5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.

6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды git status. Сделайте коммит второго изменения.

7. Просмотрите историю коммитов с помощью команды git log. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.

8. Верните рабочий каталог к одному из предыдущих состояний.

9. Изучите, как создавать теги для коммитов для использования в будущем.

10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).

11. Отмените один из коммитов в локальном репозитории.

**Часть 2. Системы управления репозиториями**

1. Создайте аккаунт на GitHub (у кого нет),

2. Создайте репозиторий на GitHub и на локальной машине, согласно выбранной теме проекта,

3. Создайте несколько файлов на локальной машине при помощи консоли,

4. Создайте SSH-ключ для авторизации,

5. Свяжите репозиторий локальной машины с репозиторием на GitHub при помощи консоли,

6. Создайте новую ветку в репозитории с помощью команды, произведите в ней какие-нибудь изменения, а после слейте с веткой master,

7. Выполните цепочку действий в репозитории, согласно вариантам.

Вариант № 5

Задание:

1) Клонируйте непустой удаленный репозиторий на локальную машину

2) Создайте новую ветку и выведите список всех веток

3) Произведите 3 коммита в новой ветке в разные файлы

4) Выгрузите изменения в удаленный репозиторий

5) Произведите revert предпоследнего коммита в новой ветке

6) Выведите в консоли различия между веткой master и новой веткой

7) Слейте новую ветку с master при помощи merge

**Часть 3. Работа с ветвлением и оформление кода**

1. Сделайте форк репозитория в соответствии с вашим вариантом

2. Склонируйте его на локальную машину

3. Создайте две ветки branch1 и branch2 от последнего коммита в master'е

4. Проведите по 3 коммита в каждую из веток, которые меняют один и тот же кусочек файла

5. Выполните слияние ветки branch1 в ветку branch2, разрешив конфликты при этом

6. Выгрузите все изменения во всех ветках в удаленный репозиторий

7. Проведите еще 3 коммита в ветку branch1

8. Склонируйте репозиторий еще раз в другую директорию

9. В новом клоне репозитории сделайте 3 коммита в ветку branch1

10. Выгрузите все изменения из нового репозитория в удаленный репозиторий

11. Вернитесь в старый клон с репозиторием, выгрузите изменения с опцией -force

12. Получите все изменения в новом репозиторииВариант № 7

Репозиторий: <https://github.com/kettanaito/naming-cheatsheet>

# Часть первая

## Пункт 1

В первую очередь устанавливаем и настраиваем Git. На рисунке 1 представлено окно окончания установки Git.

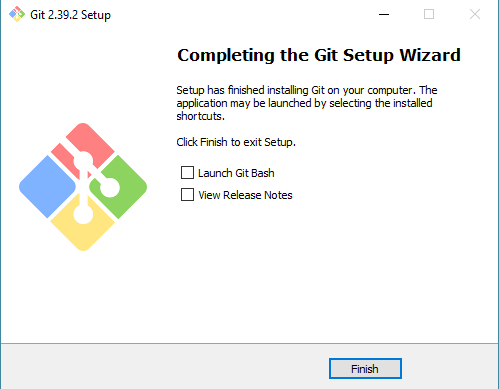


Рисунок 1 – Установка Git

Далее проведем первоначальную настройку Git. На рисунке 2 представлена первоначальная настройка, в частности настройка имени и почты.

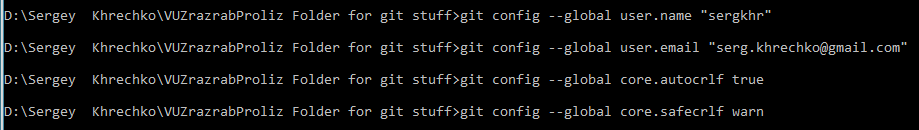


Рисунок 2 – Первоначальная настройка

## Пункт 2

На рисунке 3 представлено создание репозитория в папке hello и добавление в него нескольких файлов.

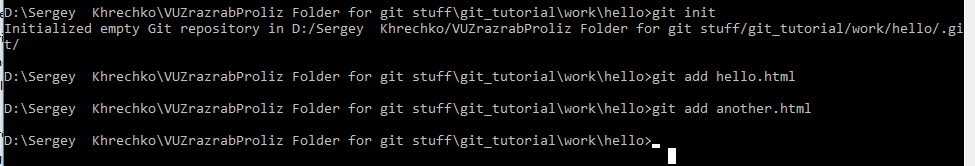


Рисунок 3 – Создание и наполнение репозитория

## Пункт 3

На рисунке 4 представлено внесение изменений в файл hello.html.

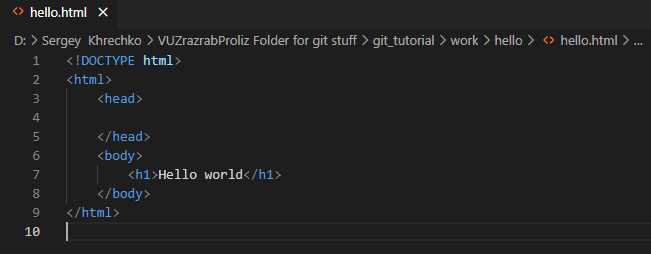


Рисунок 4 – Внесение изменений в файл

На рисунке 5 показано, что изменения еще не проиндексированы.

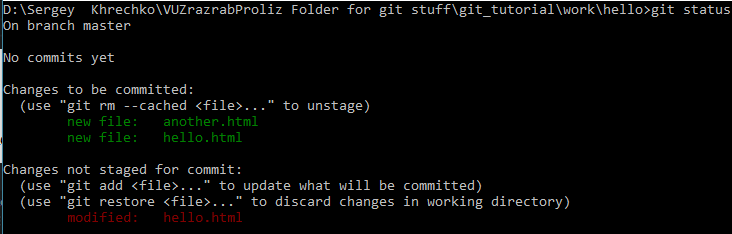


Рисунок 5 – Не проиндексированные изменения

## Пункт 4

На рисунке 6 представлено индексирование изменений в файле hello.html и проверка статуса репозитория.

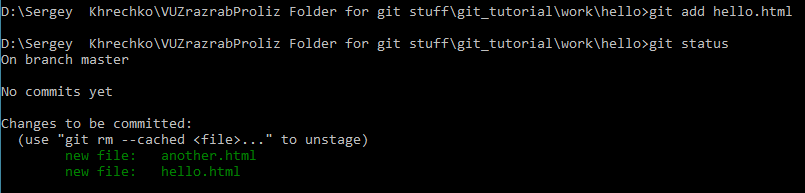


Рисунок 6 – Индексирование и проверка

## Пункт 5

На рисунке 7 представлено совершение коммита, а также добавление комментария.

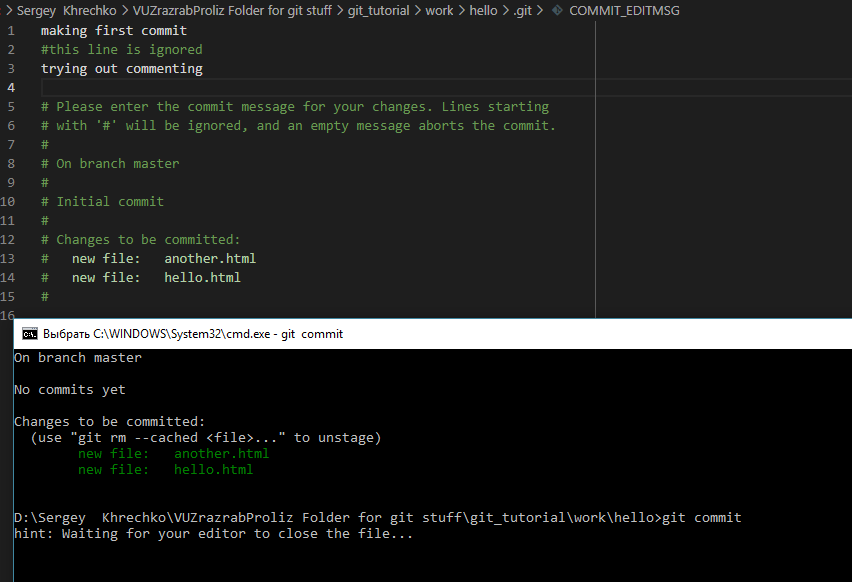


Рисунок 7 – Коммит и комментарий

На рисунке 8 представлены результаты первого коммита.

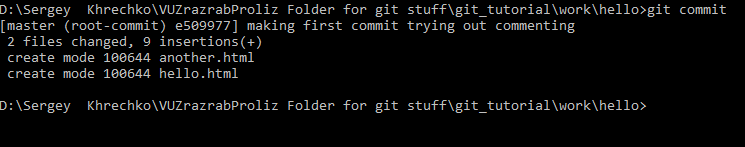


Рисунок 8 – Результаты

## Пункт 6

На рисунке 9 представлено внесение изменений в другой файл и их индексирование.

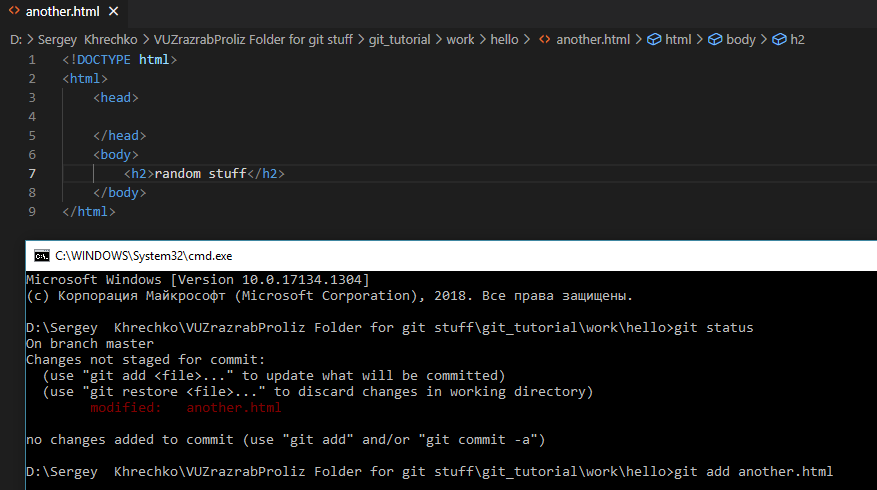


Рисунок 9 – Изменение и индексирование

На рисунке 10 представлено индексирование изменений, а также проверка репозитория после внесения новых изменений.

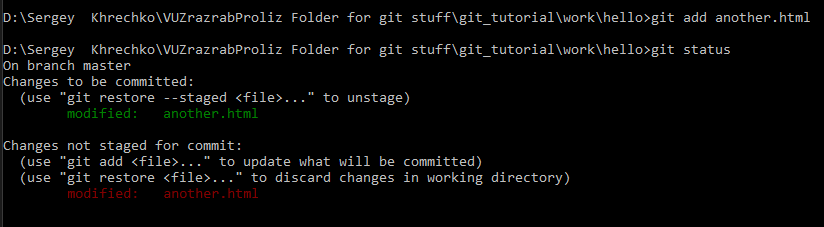


Рисунок 10 – Индексирование и проверка

На рисунке 11 представлено индексирование второго изменения и проверка репозитория.

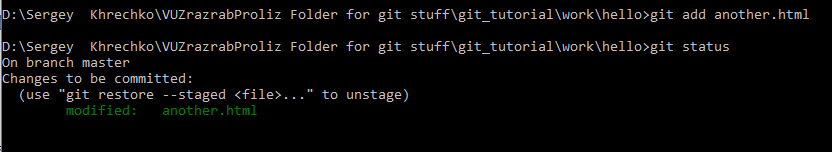


Рисунок 11 – Индексирование и проверка

На рисунке 12 представлено совершение коммита изменений в another.

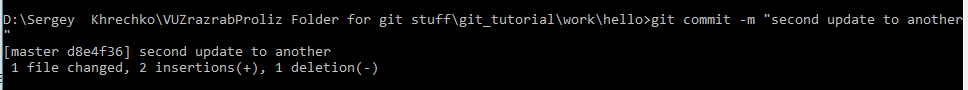


Рисунок 12 – Коммит в another

## Пункт 7

На рисунке 13 представлена история работы с репозиторием.

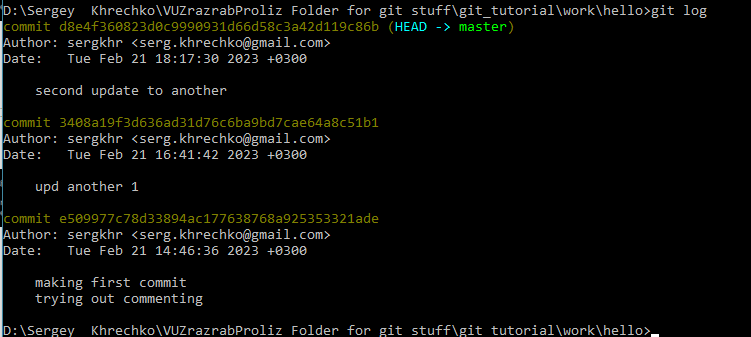


Рисунок 13 – История работы

На рисунке 14 представлены различные отображения истории коммитов. В первом случае это вывод в одну строку, во втором вручную задан формат.

%h – это обозначение для хэша, %ad – для даты, %an – для автора, %s – комментарий, %d - украшения, --date=short – выводит дату в сокращенном виде.

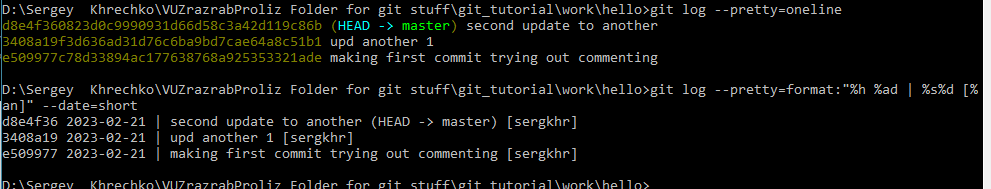


Рисунок 14 – Различные отображения истории

Для удобства был создан псевдоним для красивого вывода истории коммитов. Процесс и результат показаны на рисунке 15.

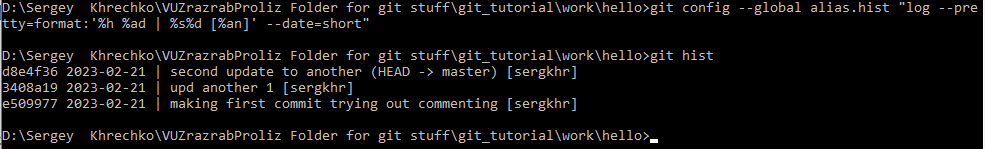


Рисунок 15 – Псевдоним для истории коммитов

## Пункт 8

На рисунке 16 представлено возвращение каталога к состоянию самого первого коммита. Как можно заметить в файле another.html теперь ничего нет, как и должно быть, так как первое наполнение появилось только во втором коммите.

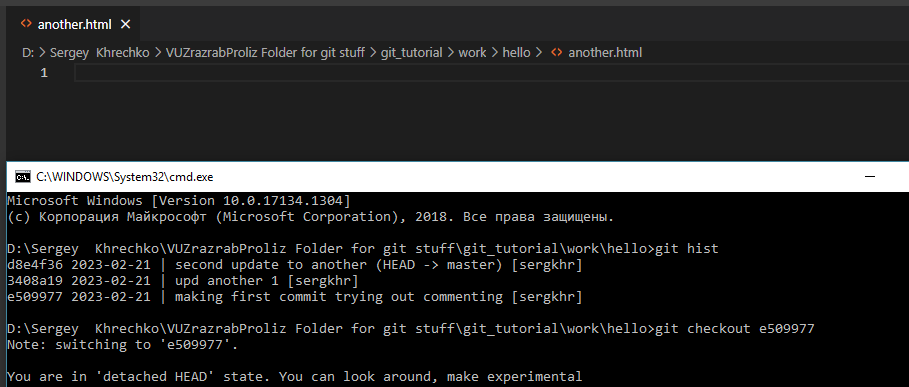


Рисунок 16 – Возвращение к предыдущему состоянию

## Пункт 9

На рисунке 17 представлено создание тегов для коммитов. Версию до изменения мы назвали v1, а последнюю на данный момент версию назвали v2.

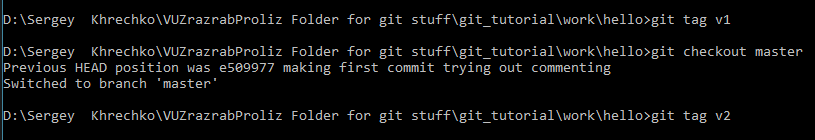


Рисунок 17 – Создание тегов

На рисунке 18 представлена навигация между коммитами с тегами. Нотация v2~1 означает, что мы попадаем в коммит, который на 1 старше коммита v2.

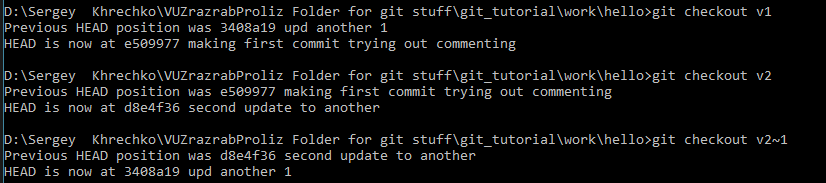


Рисунок 18 – Навигация по тегам

## Пункт 10

На рисунке 19 представлена отмена изменений в рабочем каталоге до их индексирования, при помощи команды checkout.

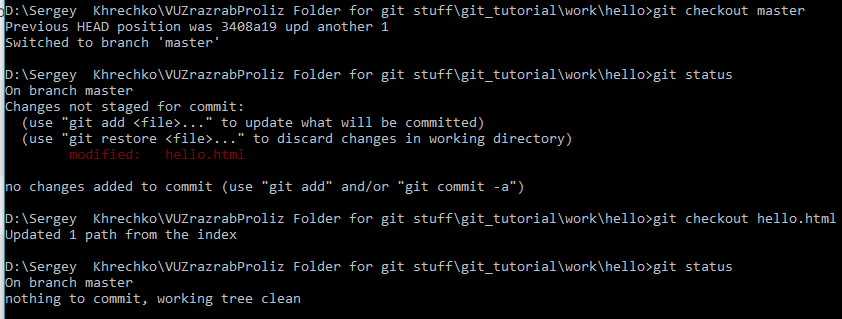


Рисунок 19 – Отмена неиндексированных изменений

На рисунке 20 представлена отмена изменений в рабочем каталоге после индексирования. Команда reset отменяет индексирование, после чего производится отмена уже неиндексированных изменений, как в примере выше.

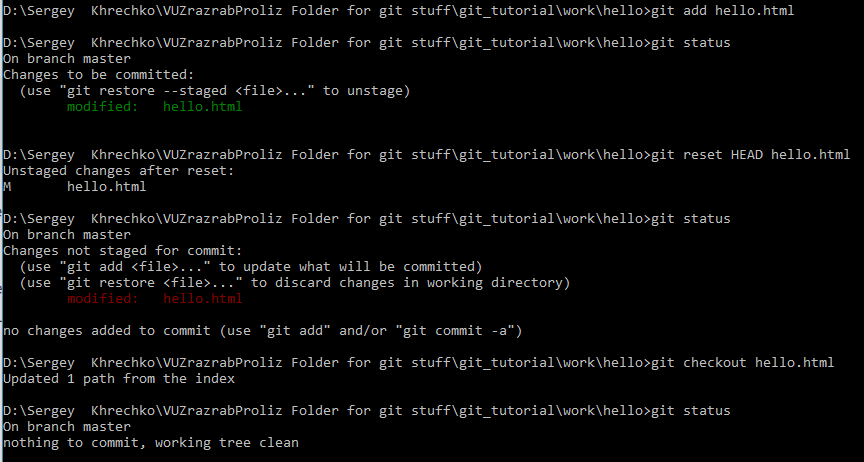


Рисунок 20 – Отмена индексированных изменений

## Пункт 11

На рисунке 21 представлена отмена нежелательного коммита через создание нового, отменяющего изменения, командой revert.

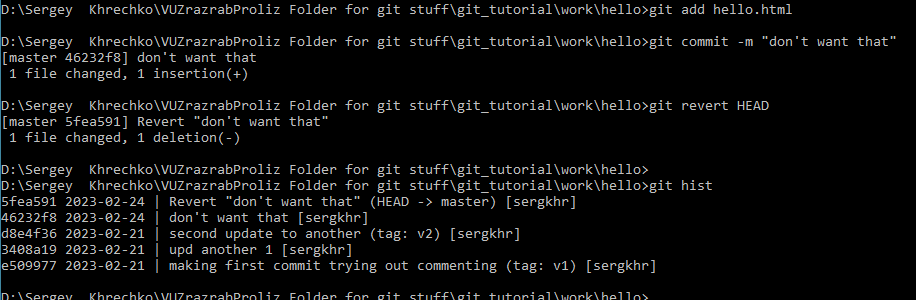


Рисунок 21 – Отмена коммита

На рисунке 22 представлено удаление двух коммитов с помощью команды reset, теперь они не будут отображаться при обычном вызове git log.

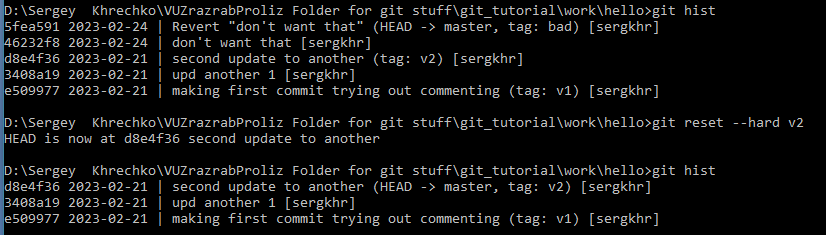


Рисунок 22 – Удаление коммитов

# Часть вторая

## Пункт 1

На рисунке 23 представлен мой аккаунт на github.

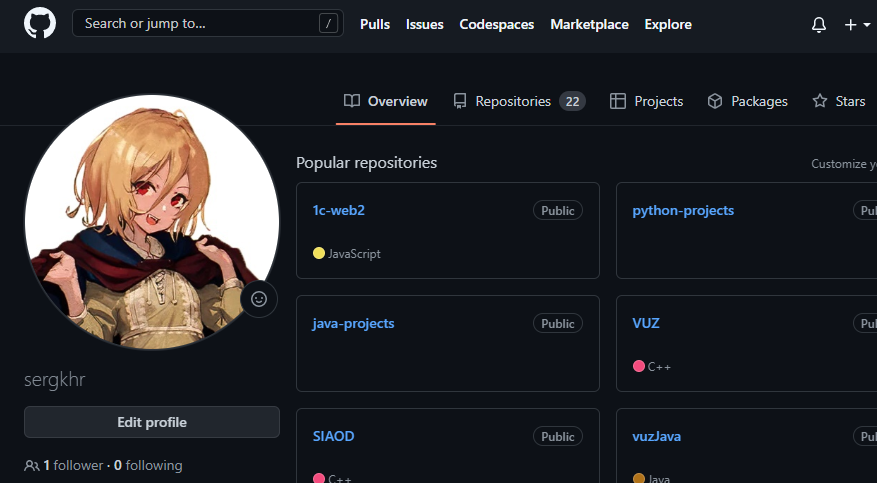


Рисунок 23 – Аккаунт на github

## Пункт 2

На рисунке 24 представлен результат создания репозитория для практики.

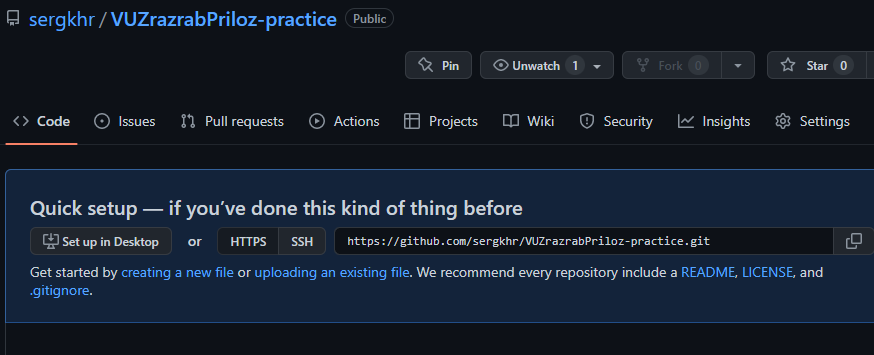


Рисунок 24 – Создан репозиторий на github

На рисунке 25 представлено создание локального репозитория.

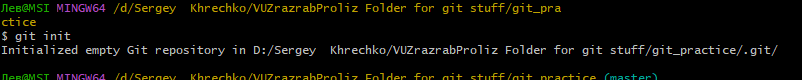


Рисунок 25 – Создание локального репозитория

## Пункт 3

На рисунке 26 представлено создание нескольких файлов в локальном репозитории.



Рисунок 26 – Создание нескольких файлов

На рисунке 27 представлено создание коммита с этими тремя файлами.



Рисунок 27 – Коммит

## Пункт 4

На рисунке 28 представлено получение SSH ключа для настройки соединения с удаленным репозиторием.

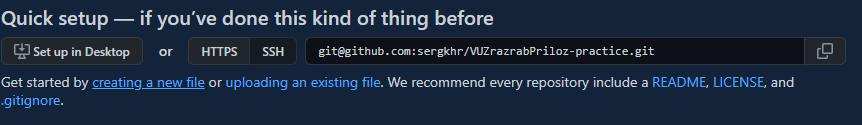


Рисунок 28 – SSH ключ

## Пункт 5

На рисунке 29 представлено связывание локального репозитория с удаленным.



Рисунок 29 – Связывание репозитория

## Пункт 6

На рисунке 30 представлено создание новой ветки и внесение в нее изменений.

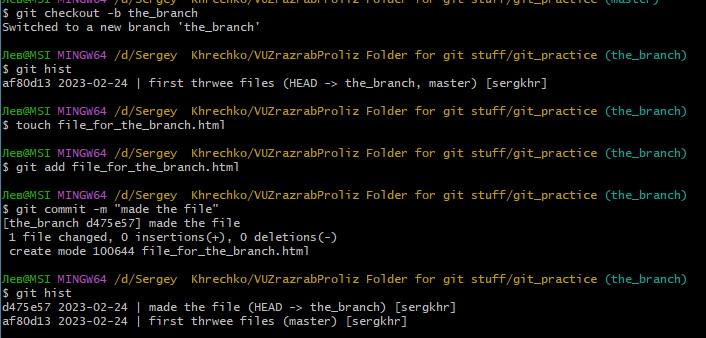


Рисунок 30 – Создание ветки и изменения

На рисунке 31 представлено слияние ветвей the\_branch и master.

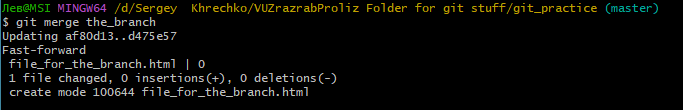


Рисунок 31 – Слияние

## Пункт 7

### Подпункт 1

На рисунке 32 представлено клонирование удаленного репозитория на локальную машину.

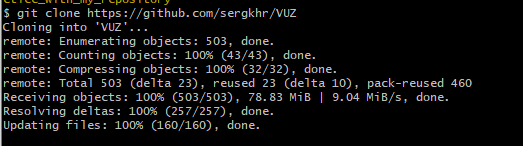


Рисунок 32 – Клонирование репозитория

### Подпункт 2

На рисунке 33 представлено создание новой ветки, а также выведен список всех веток репозитория.

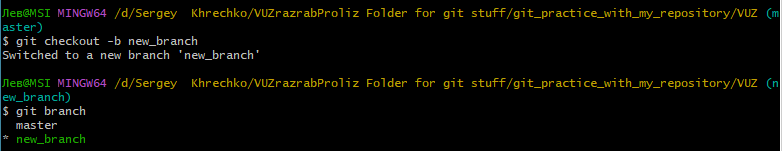


Рисунок 33 – Ветки репозитория

### Подпункт 3

На рисунке 34 представлены записи о трех коммитах на новой ветке. Можно заметить, что до этих коммитов идут коммиты на ветке master.

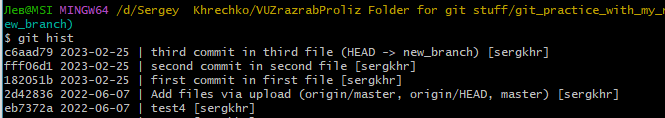


Рисунок 34 – Коммиты на новой ветке

На рисунке 35 представлен третий коммит, предыдущие два сделаны аналогично.

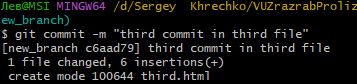


Рисунок 35 – Третий коммит на новой ветке

### Подпункт 4

На рисунке 36 представлена выгрузка изменений в удаленный репозиторий.

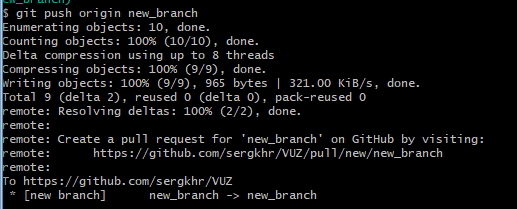


Рисунок 36 – Выгрузка изменений

На рисунке 37 показано, что изменения были успешно выгружены.

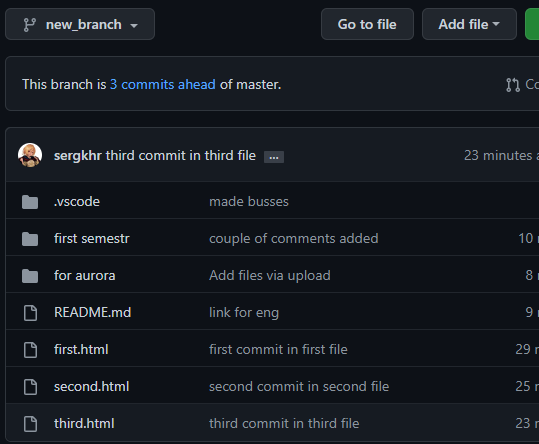


Рисунок 37 – Изменения в репозитории успешны

### Подпункт 5

На рисунке 38 представлено возвращение к состоянию предпоследнего коммита с помощью команды revert.

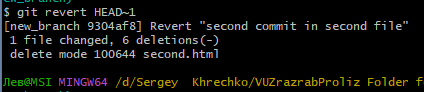


Рисунок 38 – Revert до предпоследнего состояния

### Подпункт 6

На рисунке 39 представлены различия между веткой master и новой веткой.

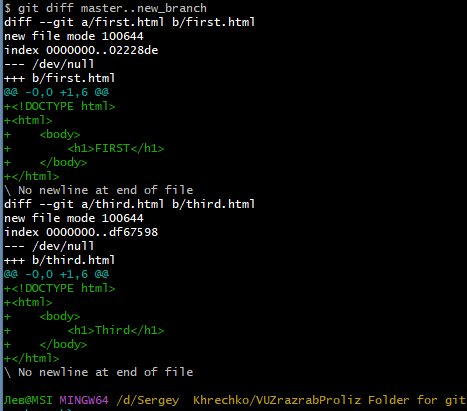


Рисунок 39 – Различия между ветками

### Подпункт 7

На рисунке 40 представлено слияние новой ветки с веткой master.

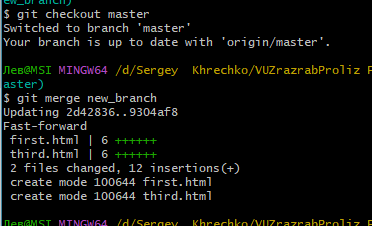


Рисунок 40 – Слияние

# Часть третья

## Пункт 1

На рисунке 23 представлен мой аккаунт на github.

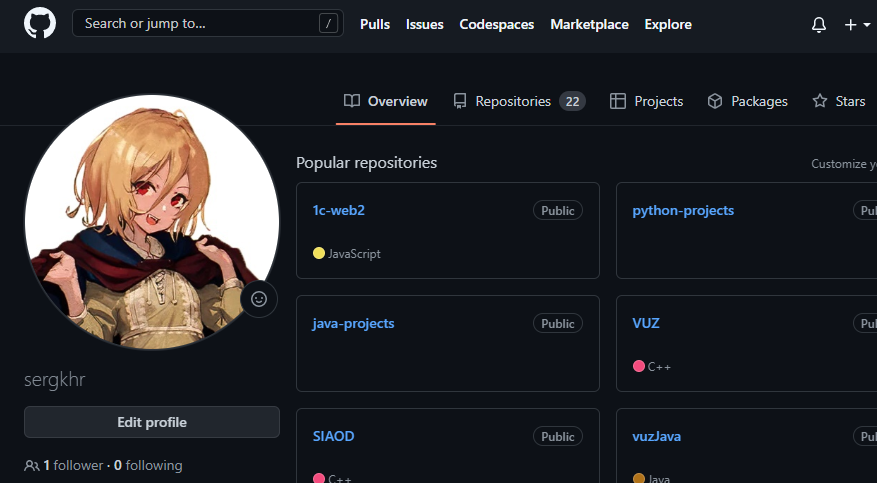


Рисунок 23 – Аккаунт на github