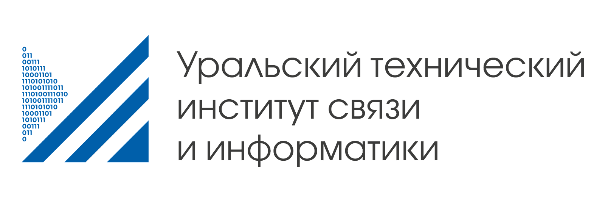
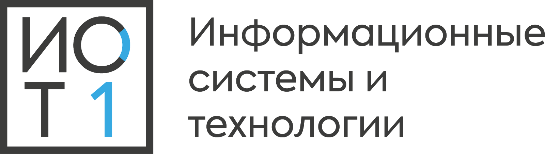
Министерство цифрового развития, связи и массовых коммуникаций РФ

Уральский технический институт связи и информатики (филиал) ФГБОУ ВО "Сибирский государственный университет телекоммуникаций и информатики" в г. Екатеринбурге

(УрТИСИ СибГУТИ)

КАФЕДРА

ИСТ

**ОТЧЕТ**

По дисциплине «Сетевое программирование»

Практическое занятие № 1

«Работа с системой управления версиями»

|  |  |
| --- | --- |
| Выполнил: | студент гр. ПЕ-12б  Камков Д.А. |
| Проверил: | зав. каф. ИСТ,   Бурумбаев Д.И. |
| Ассистент: |  |

 Екатеринбург, 2024

1 Цель работы:

1.1 Научиться работать с системой управления версиями;

1.2 Закрепить знания по теме «Системы управления версиями».

2 Перечень оборудования:

2.1 Персональный компьютер;

2.2 Система управления версиями Git;

2.3 Visual Studio Code.

3.Ход работы:

3.1 Перед началом работы с Git необходимо создать папку с фамилией и номером группы, в которой будут хранится наши элементы. После этого, необходимо создать следующую структуру, как показано на рисунке 1.

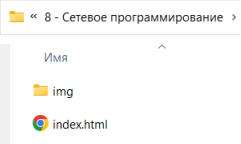


Рисунок 1 – Структура папки

Как видно из структуры, в корневой директории лежат:

1. Основная страница блога index.html

2. Папка img, в которой хранятся изображения (файлы \*.jpg) сайта.

Тематика блога выбирается самостоятельно, картинки – тоже, но в рамках общественной нормы.

В качестве примера предполагается, что блог посвящен диким животным. Тогда структура файла index.html будет выглядеть как показано на рисунке 2.

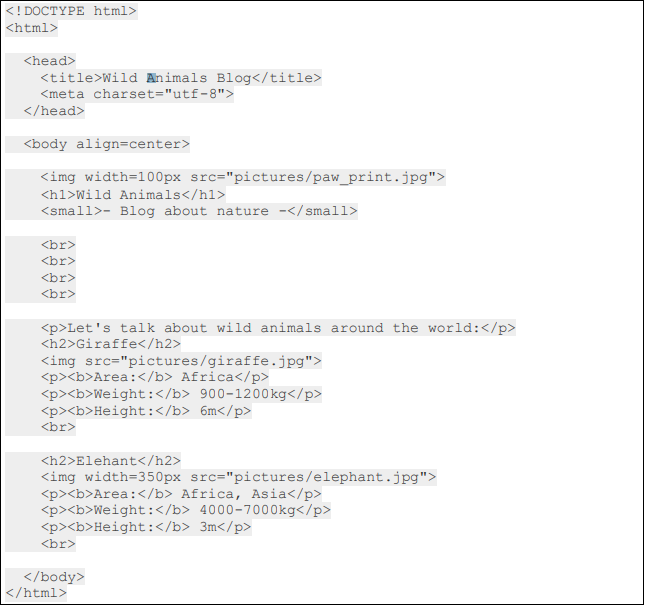


Рисунок 2 – Структура index.html

Если отрыть данный файл в браузере, то выглядит основная (и пока единственная) страница блога показана на рисунке 3:

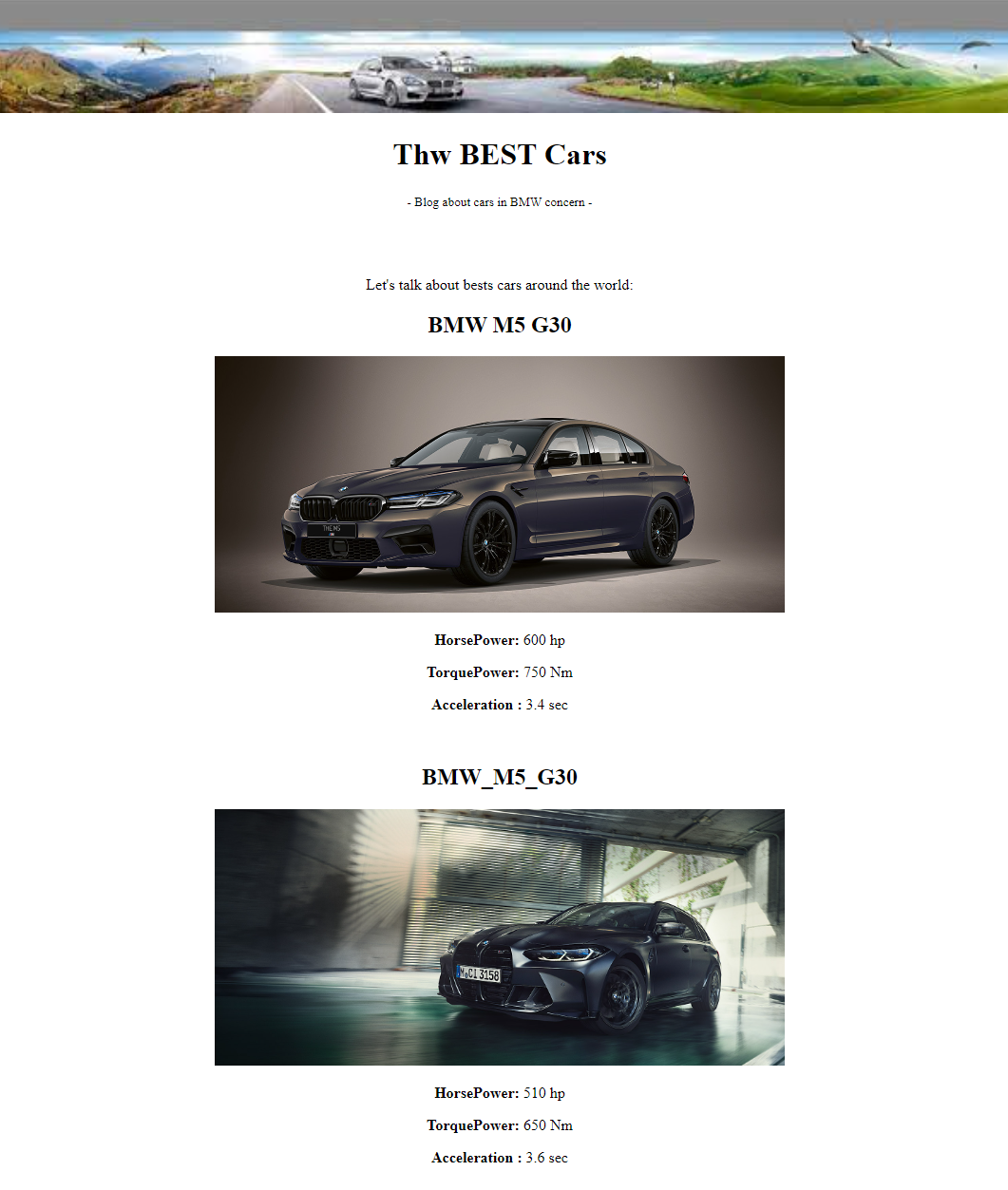


Рисунок 3 – Страница блога в браузере

Далее требуется создать репозиторий для нашего блога. Для этого необходимо перейти в директорию будущего репозитория (или открыть при помощи терминала вашу папку)

Далее необходимо прописать команду git. Если у Вас открывается сообщение, как показано на рисунке 3, то значит git установлен. Если подобное сообщение не открывается необходимо установить его в зависимости от Вашей операционной системы с сайта <https://git-scm.com/book/ru/v2/ВведениеУстановка-Git>

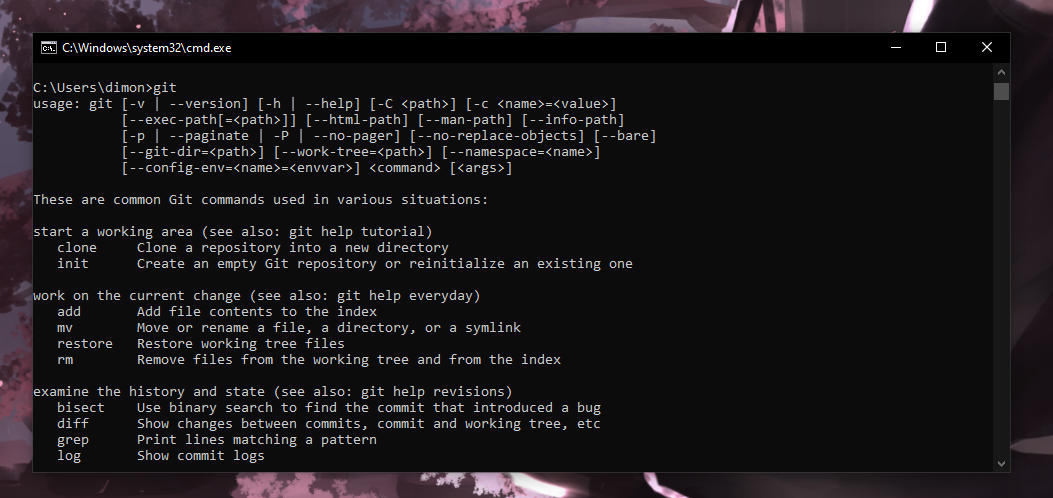


Рисунок 4 – Сообщение, если git установлен

Для создания репозитория необходимо прописать команду git init, после чего появится сообщение:

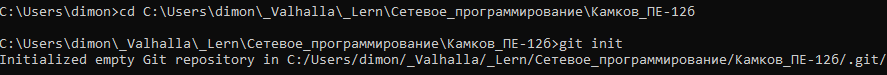


Рисунок 4 – Результат выполнения “git init”

После такого сообщения, можно проверить структуру git файла, как показано на рисунке 5:

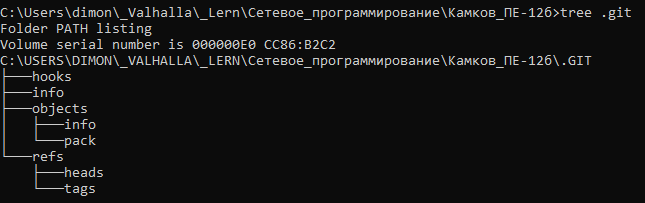


Рисунок 5 – Проверка структуры файла .git

Далее необходимо проверить текущее состояние рабочей директории, как показано на рисунке 6:

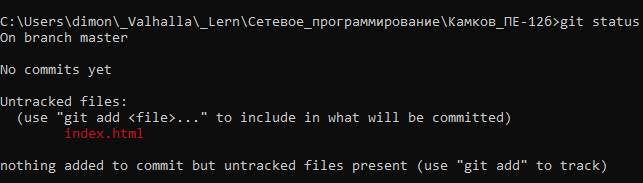


Рисунок 6 – Проверка состояния директории

Как видно из листинга, есть не отслеживаемые файлы. Как сделать отслеживаемыми – будет чуть-чуть попозже.

Далее необходимо выполнить следующее задание – настроить пользователя Git на уровне локального репозитория:

1) Изучите содержимое файла конфигурации Git для текущего репозитория (.git/config)

2) Настройте имя и email пользователя для текущего репозитория

3) Убедитесь, что файл .git/config изменился соответствующим образом

Для вывода содержимого файла конфигурации Git для текущего репозитория необходимо воспользоваться командой, как показано на рисунке 7.

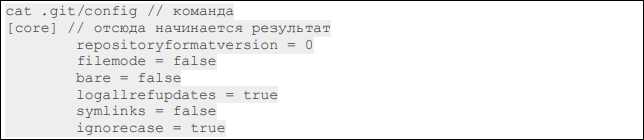


Рисунок 7 – Вывод содержимого конфигурации

Далее необходимо задать имя и email пользователя для текущего репозитория при помощи команды. Значение name и your@email.ru указываются студентами самостоятельно.

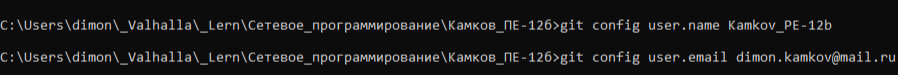


Рисунок 8 – Добавление имени и email пользователя

После этого, необходимо проверить содержимое файла конфигурации при помощи команды, которая была использована на рисунке 7. Результат выполнения для тестового задания представлен на рисунке 9.

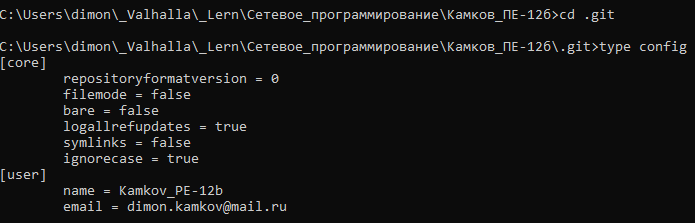


Рисунок 9 – Вывод содержимого файла конфигурации

Следующим заданием будет необходимо создать первый коммит. Для этого необходимо посмотреть состояние рабочей директории, как объяснялось ранее.

После того, как было проверено текущее состояние директории, необходимо добавить все файлы в индекс при помощи команды git add \*.

После этого, снова необходимо проверить статус репозитория. Если все сделано верно, то файлы попали в индекс и теперь готовы к коммиту, как показано на рисунке 10.

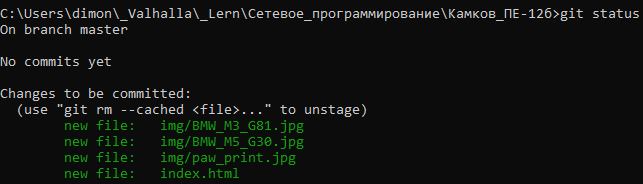


Рисунок 10 – Статус репозитория после добавления в индекс

Далее необходимо выполнить коммит при помощи команды git commit - m "Project initial". После этого должен быть ответ, как показан на рисунке 11.

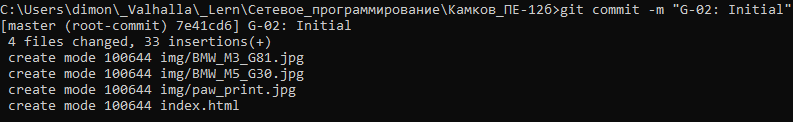


Рисунок 11 – Ответ после выполнения команды

Далее можно проверить последнее изменение при помощи команды git log, как показано на рисунке 12.

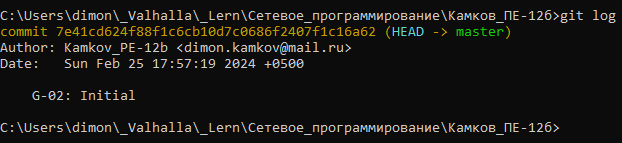


Рисунок 12 – Результат выполнения команды git log

В конце необходимо проверить статус репозитория, результат выполнения представлен на рисунке 13.

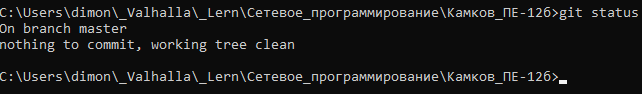


Рисунок 13 – Результат выполнения git status

Индивидуальное задание по пункту 3.1. В существующий файл index.html необходимо добавить еще одну картинку и дополнительный текст и сделать второй коммит самостоятельно. Результат выполнения может быть представлен в форме листинга/скриншота.

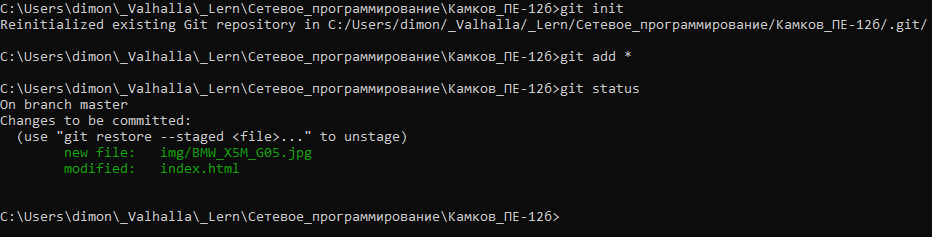


Рисунок 14 – Скриншот листинга индивидуального задания

3.2 Индивидуальное задание студента.

1) Зарегистрируйтесь на GitHub и подтвердите свою почту.

2) Настройте SSH или HTTPS подключение (на ваш выбор). Для того, чтобы настроить SSH, можно прочитать здесь:

<https://timeweb.cloud/tutorials/windows/kak-sgenerirovat-ssh-klyuch-dlya-windows>

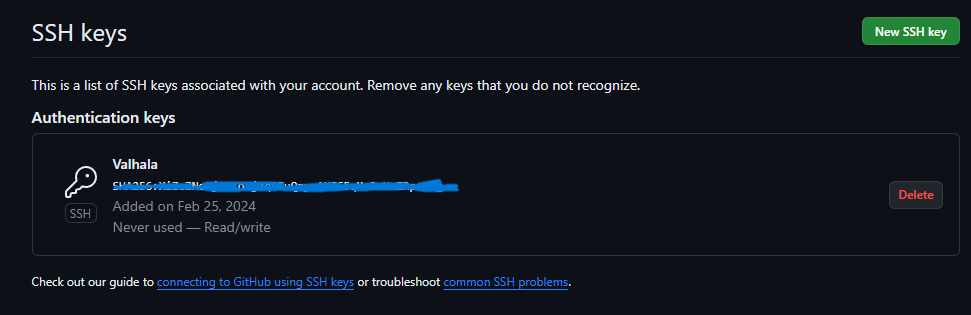


Рисунок 15 – SSH-key задания 3.2.2

3) Отправьте ваш локальный репозиторий (созданный в задании 3.1) на аккаунт GitHub (при помощи командной строки или вспомогательного ПО, не имеет значения).

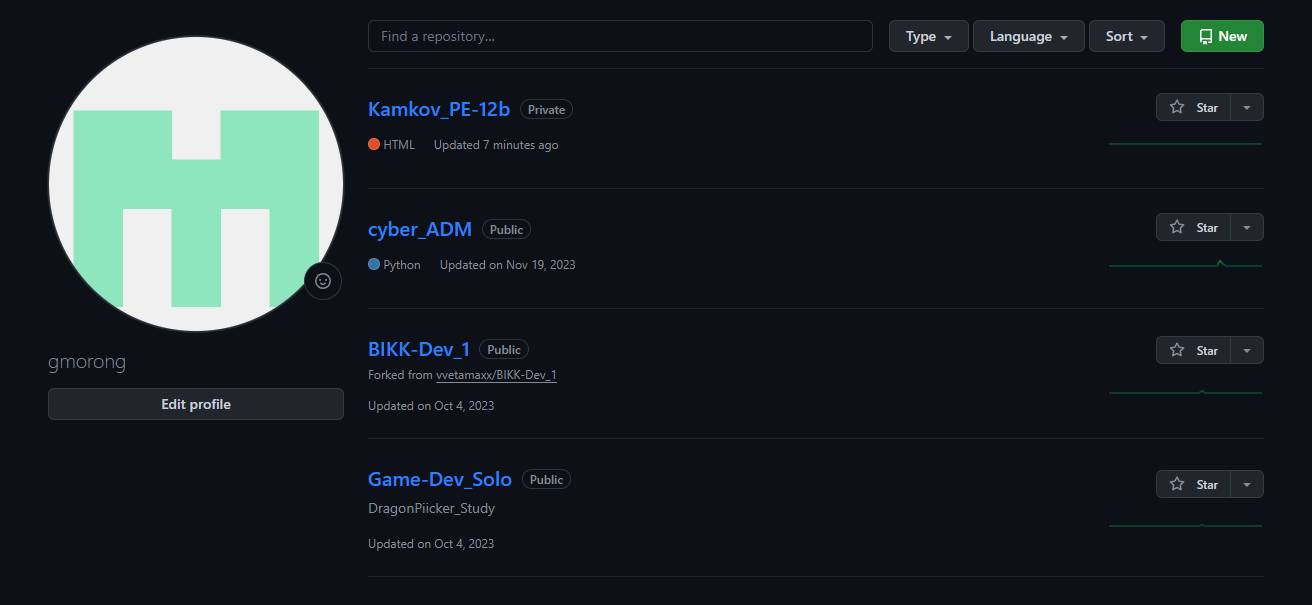


Рисунок 16 – Репозиторий на GitHub’е

4) Внесите изменения в ваш репозиторий на локальном компьютере, сделайте коммит и отправьте его на GitHub.

5) Создайте ветку с указанием Вашей фамилии и инициалов, например, BurumbaevDI

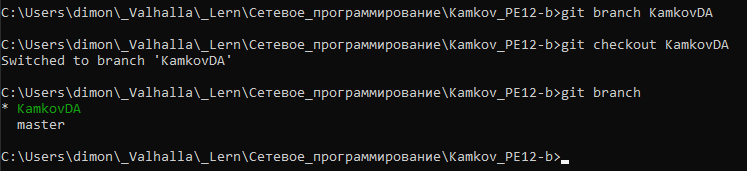


Рисунок 17 – Задание 3.2.5

6) Далее необходимо переключится на ветку с вашей фамилией, изменить ваш файл index.html и отправить его на GitHub (именно с второй ветки).

7) Необходимо выполнить Pull Request, указав комментарий.

8) После этого необходимо произвести слияние версий от главной ветки (master/main) и второстепенной (в случае примера, это BurumbaevDI). Данный раздел необходимо изучить самостоятельно.

Для того, чтобы выполнить индивидуальное задание No2 необходимо ознакомиться с приложением А к первой практической работе.

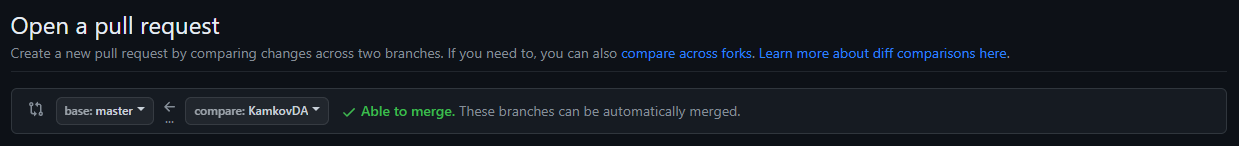


Рисунок 18 – Итог индивидуального задания