**Практическая работа №1: «Работа с Git: создание репозитория, коммиты»**

Цель работы: освоить базовые навыки работы с системой контроля версий Git.

**Продолжительность:** 1 час 20 минут.

**Необходимое ПО:**

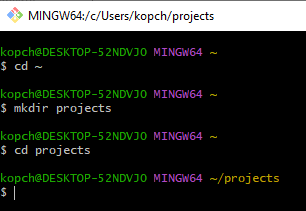
1. Установленный Git
2. Текстовый редактор (любой)
3. Аккаунт на GitHub или GitLab
4. **Терминал:**
   * Windows: Git Bash, Command Prompt (CMD) или PowerShell
   * Linux/macOS: Terminal

**Ход работы**

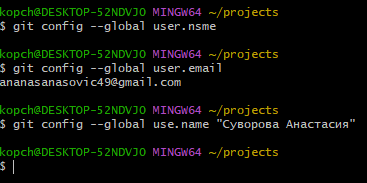
**Часть 1: Подготовка и настройка**

Шаг 1.1: Мы открыли терминал через «Пуск»

Шаг 1.2: Создали рабочую папку

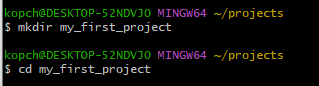


Шаг 1.3: Настроили Git

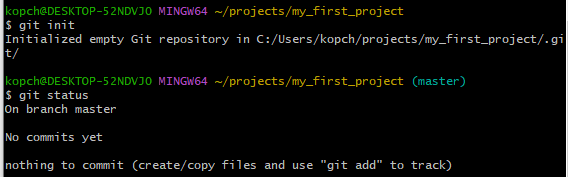


**Часть 2: Работа с локальным репозиторием**

Шаг 2.1: Создали проект



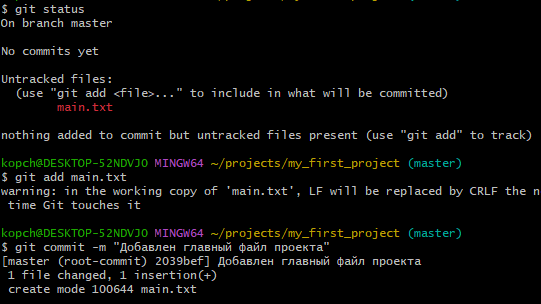
Шаг 2.2: Инициализировали репозиторий



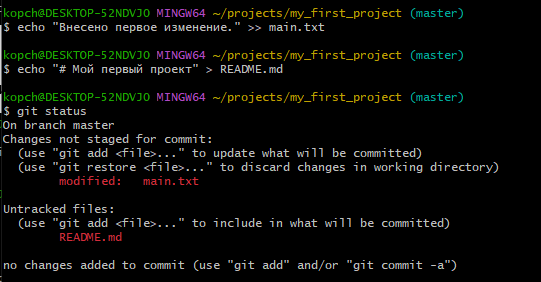
Шаг 2.3: Создали первый файл



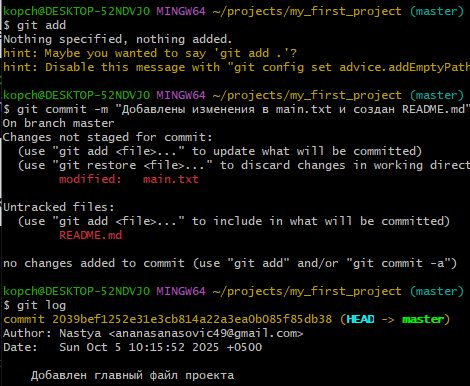
Шаг 2.4: Выполнили первый коммит



Шаг 2.5: Добавили изменения



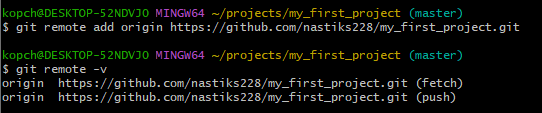
Шаг 2.6: Выполнили второй коммит



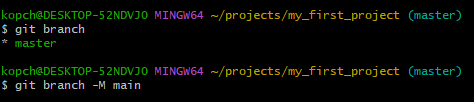
**Часть 3: Работа с удалённым репозиторием**

Шаг 3.1: Создали репозиторий на GitHub

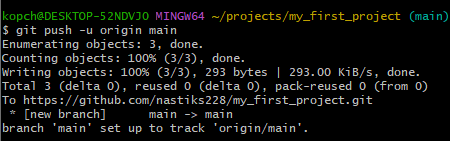
Шаг 3.2: Связали с удалённым репозиторием



Шаг 3.3: Определили имя основной ветки



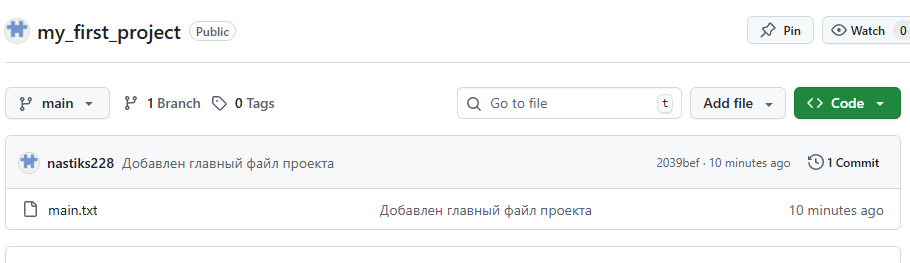
Шаг 3.4: Отправили изменения на GitHub



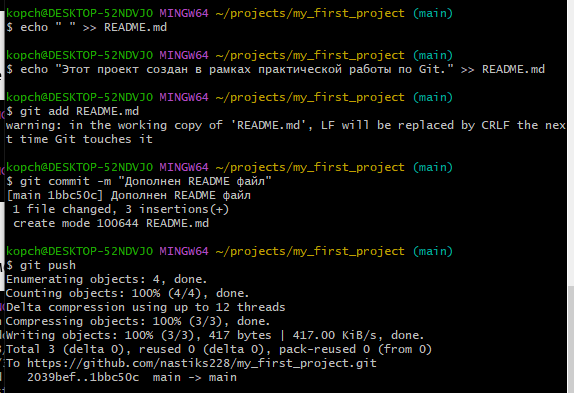
**Часть 4: Проверка и дополнительные задания**

Шаг 4.1: Проверили результат

1. Обновили страницу репозитория на GitHub
2. Убедились, что все файлы и коммиты отображаются



Шаг 4.2: Дополнительное задание



Ссылка на репозиторий: https://github.com/nastiks228/my\_first\_project.git

**Часть 5: Вопросы для самоконтроля**

Шаг 5.1: Создать файл для ответов на вопросы

В самом репозитории создать файл answers.txt или answers.md (создание файла отдельный коммит)

Шаг 5.2: Ответить на вопросы и выполнить коммит с обновлённой версией файла

1. Какие три основные области работы с файлами в Git (рабочая папка, индекс, репозиторий) и чем они отличаются?
2. Что происходит при выполнении команд:
   * git add
   * git commit
   * git push
3. Почему важно писать информативные сообщения к коммитам а не просто "исправление" или "update"?
4. Чем отличается git add имя\_файла от git add .? В каких случаях лучше использовать каждый вариант?
5. Что показывает команда git status и как меняется ее вывод на разных этапах работы?
6. Для чего нужна команда git remote add origin URL и что означает слово "origin"?
7. Зачем нужен флаг -u в команде git push -u origin main при первой отправке изменений?
8. Что такое ветка (branch) по умолчанию и какая у вас создалась (main или master)? Почему это важно?
9. Как просмотреть историю коммитов и какую информацию о каждом коммите вы видите?
10. Что произойдет, если сделать изменения в файле, но не выполнить git add перед git commit?

**Для сдачи работы предоставить ссылку на Github-репозиторий,** в котором:

* полноценная история коммитов;
* все необходимые по заданию файлы;
* информативные сообщения коммитов;
* нет лишних файлов, которых не требовалось по заданию (допускается, что в ходе выполнения работы в репозиторий попадали лишние файлы, но вы их удалили из финальной версии).