



Практическое задание

Тема: Работа с репозиторием и коммитами в Git + GitHub (GUI)

Цель

- Понять структуру Git-репозитория
- Научиться создавать и настраивать репозиторий на GitHub
- Закрепить навык работы с коммитами через интерфейс (VS Code / GitHub Desktop)
- Осознать роль осмысленных сообщений коммитов и истории изменений

Задание

1. Создание репозитория

1. Зайдите на сайт GitHub под своим аккаунтом.
2. Создайте **новый репозиторий**:
 - Название: `git-practice-<ваша_фамилия>`
 - Тип: Public
 - Инициализируйте репозиторий с `README.md`
3. Склонируйте репозиторий **локально**:
 - либо через **GitHub Desktop**
 - либо через кнопку *Clone Repository* в **VS Code**

2. Первые изменения и коммит

1. Откройте проект в **VS Code**.
2. Откройте файл `README.md`.
3. Добавьте в файл:

```
# Git Practice

Студент: <Имя Фамилия>
Группа: <Номер группы>
Дата: <дата выполнения>
```

4. Перейдите во вкладку **Source Control**.
5. Убедитесь, что файл отображается как изменённый.
6. Создайте **первый коммит**:
 - Сообщение коммита: Добавлен README с информацией о студенте
7. Отправьте (Push) изменения на GitHub.

3. Работа с несколькими коммитами

1. Создайте в корне проекта папку `notes`.
2. В папке `notes` создайте файл `git-basics.md`.
3. Заполните файл краткими ответами:

```
Что такое Git?  
Что такое репозиторий?  
Зачем нужны коммиты?
```

4. Сохраните файл.
5. Создайте **отдельный коммит**:
 - Сообщение: Добавлены заметки по основам Git
6. Выполните Push.

4. Исправление и дополнительный коммит

1. Вернитесь в `README.md`.
2. Добавьте раздел:

```
## Что я узнал(а)
- Что такое репозиторий
- Как создавать коммиты через интерфейс
- Как отправлять изменения на GitHub
```
3. Зафиксируйте изменения **отдельным коммитом**:
 - Сообщение: Дополнен README разделом "Что я узнал(а)"
4. Отправьте изменения на GitHub.

Подсказки по ключевым частям

- **Коммит = логически завершённое изменение**, а не “всё сразу”.
- Сообщение коммита должно отвечать на вопрос: *что изменилось?*
- Перед коммитом проверяйте список файлов в Source Control.
- Не объединяйте разные по смыслу изменения в один коммит.

Что проверить перед отправкой (чек-лист)

Перед сдачей убедитесь, что:

- Репозиторий создан на GitHub и является Public
- В репозитории есть `README.md` с данными студента
- Есть папка `notes` и файл `git-basics.md`
- В истории минимум **3 коммита**
- Сообщения коммитов осмысленные и на русском языке
- Все изменения отправлены (Push выполнен)

Советы по улучшению работы

- Делайте коммиты чаще, но меньшего размера.
- Один коммит — одна логическая задача.
- Проверяйте историю коммитов на GitHub перед отправкой работы.
- Привыкайте писать сообщения коммитов так, как будто их будет читать другой разработчик.
- Используйте GitHub как **журнал прогресса**, а не только как хранилище файлов.