**Лабораторная работа №6**

Установка интеграции с Проводником, выбрав чекбокс: Git Bash Here

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

Контент, сгенерированный ИИ, может содержать ошибки.

Проверка установленной версии

Изображение выглядит как текст, Шрифт, снимок экрана, Графика

Контент, сгенерированный ИИ, может содержать ошибки.

Для получения справки по Git используем команду **git help** и получим:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Создание локального репозитория с помощью команды **git init**

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

Контент, сгенерированный ИИ, может содержать ошибки.

Настройка конфигурационного файла:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Проверка настройки конфигурационного файла командой **git config –list**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Инициализация локального репозитория в рабочей папке, где он будет размещён. Исследование изменения, которые произошли в папке:

1) Переходим в рабочую папку (в моём случаем Kononovich -> Север -> 6)

2) ПКМ открываем **Git Bash here**

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

Контент, сгенерированный ИИ, может содержать ошибки.**

3) С помощью команды **git init** создаём локальный репозиторий

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

Контент, сгенерированный ИИ, может содержать ошибки.

4) В папке появилась новая скрытая папка: .git – локальный репозиторий

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

Контент, сгенерированный ИИ, может содержать ошибки.

Где находятся:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Создание текстового файла Hello.txt:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Получение информации о статусе репозитория с помощью **git status**

**Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

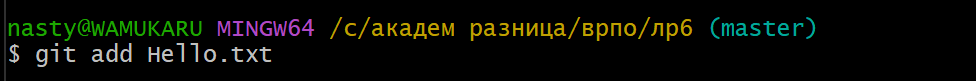
Контент, сгенерированный ИИ, может содержать ошибки.**

Можно заметить, что создана ветка **master**

Красным цветом отмечены новые и модифицированные файлы и папки

Индексация файла Hello.txt и изменения в репозитории:

1) Добавим файл Hello.txt в репозиторий с помощью команды git add Hello.txt



2) Проверим обновлённое состояние репозитория с помощью команды **git status**:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Теперь наш файл Hello.txt подсвечивается зелёным цветом, это значит, что он проиндексирован

Зафиксируем изменения индексированных файлов в репозитории с помощью команды **git commit –m “added Hello.txt to the repo”**:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Ключ –m позволяет добавить комментарий, описывающий, что именно было изменено в коммите ("added Hello.txt to the repo")

Проверим изменения в репозитории:

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Git информирует об успешном создании нового коммита, в ветку **master** добавлен 1 файл:

[master (root-commit) 4993c08] added Hello.txt to the repo

1 file changed, 4 insertions(+)

Просмотр журнала с помощью команды **git log**:

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

Контент, сгенерированный ИИ, может содержать ошибки.

Получение информации о проиндексированных и неиндексированных файлах.

1) Изменим уже существующий файл Hello.txt и сохраним изменения

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

2) Просмотрим статус нашего репозитория

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

Контент, сгенерированный ИИ, может содержать ошибки.

Можем заметить, что наш файл больше не подсвечивается зелёным цветом, и вместо надписи new file: Hello.txt теперь modified: Hello.txt, что указывает на изменения файла

3) При добавлении этого изменённого файла получим

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Теперь оно подсвечивается зелёным цветом

Настройка файла .gitignore

1) Создаём текстовый файл и прописываем внутри данного файла \*[расширения файлов, которые не хотим, чтобы они отображались в репозитории]

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

Контент, сгенерированный ИИ, может содержать ошибки.

В моём случае – это файлы с расширением **.docx**

2) Далее переименовываем наш файл в файл с названием **.gitignore** убирая расширение **.txt** и сохраняем

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

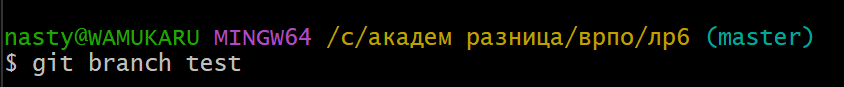
3) Далее просматриваем статус нашего репозитория и видим, что файл с расширением **.docx** больше не отображаются

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Создание новой ветки

Создание (branch) новой ветки, например, test и переключитесь (checkout) в нее

1) С помощью команды **git branch test** создаём новую ветку с именем **test** ****

2) Переключим ветки с помощью команды **git checkout test**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Создаём новый файл **test.html** и фиксируем его

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

Контент, сгенерированный ИИ, может содержать ошибки.

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

Контент, сгенерированный ИИ, может содержать ошибки.

C помощью команды **git branch** просмотрим список всех веток

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Текущей веткой являет **test**

Выполним слияние веток с помощью команды **git merge**

1) Переходим в ветку в которой будем выполнять слияние, а именно **master**

1.1) Для этого используем команду **git checkout master**

**Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.**

2) Просмотрим статус репозитории

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

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, электроника, снимок экрана, дисплей

Контент, сгенерированный ИИ, может содержать ошибки.

1) По очереди отредактируем файл test.html в ветках master и test. Выполним commit.

**Master**

**Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

**Test**

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

Контент, сгенерированный ИИ, может содержать ошибки.**

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Выполним слияние в одной и второй ветках.

**Test**

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

**Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.**

Разрешим ситуацию конфликта слияний.

Изменим наш текстовый документ и оставим тот вариант, который нас устраивает, сохраним

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

