



УРОК 19. КОМАНДНАЯ РАБОТА В GIT

КОМАНДНАЯ РАБОТА В GIT

2

ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

3



КОМАНДНАЯ РАБОТА В GIT

В процессе совместной работы над заданиями, сформулированными в issue, мы воспроизведем процесс командной разработки. Назначая индивидуальные задания по отдельным issue или работая в малых группах мы получим навыки и практический опыт совместной работы в команде над одним проектом.



ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

- Участники команды работают над назначенными задачами, которые представлены в виде отдельных issue.
- Если работа происходит в малых группах, то каждой группе назначается по одной задаче.
- Результатом работы будет создание pull request'ов, которые будут открыты для обсуждения.
- После этого будет начат процесс последовательного слияния каждого из pull request'ов с основной веткой, а также проведен процесс rebase для обеспечения актуальности.

Ссылка на репозиторий: https://github.com/it-career-hub/git_team_work.git

Подготовить скрипт, который будет копировать все файлы с определенным расширением из одной директории в другую.

Скрипт 1 в начальном состоянии:

```
Unset
#!/bin/bash

# Запрос исходной и целевой директории у пользователя

# Запрос расширения файлов, которые нужно скопировать

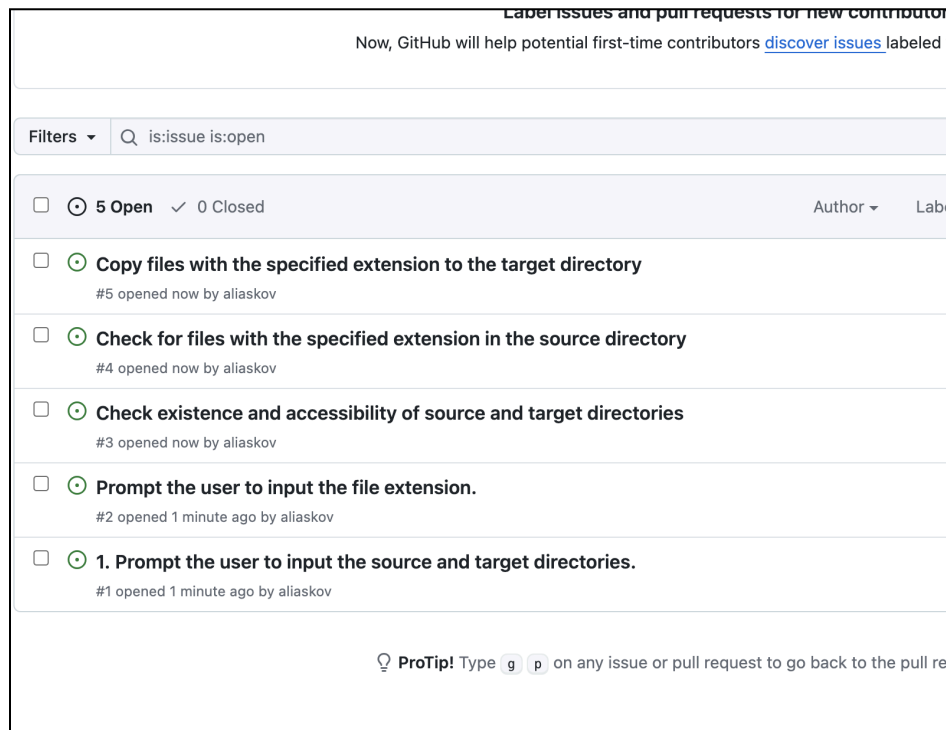
# Проверка существования исходной директории

# Проверка существования целевой директории

# Проверка, есть ли файлы с указанным расширением в исходной директории

# Копирование файлов с указанным расширением в целевую директорию
```

Issues для референса:



Запросить исходную и целевую директории у пользователя

- Описание: Запросить у пользователя указание исходной и целевой директорий.
- Действие: Предложить пользователю ввести пути к исходной и целевой директориям.

Запросить расширение файлов для копирования

- Описание: Запросить у пользователя указание расширения файлов, которые нужно скопировать.
- Действие: Предложить пользователю ввести расширение файлов.

Проверить существование и доступность исходной и целевой директорий

- Описание: Проверить, существуют ли и доступны ли исходная и целевая директории.
- Действие: Проверить существование и доступность директорий. Если они не существуют или недоступны, вывести сообщение об ошибке.

Проверить наличие файлов с указанным расширением в исходной директории

- Описание: Проверить, есть ли файлы с указанным расширением в исходной директории.
- Действие: Поиск файлов с указанным расширением в исходной директории.



Если файлы не найдены, вывести сообщение об ошибке.

Скопировать файлы с указанным расширением в целевую директорию

- Описание: Скопировать все файлы с указанным расширением из исходной директории в целевую.
- Действие: Копирование каждого файла с указанным расширением из исходной директории в целевую. Вывести сообщение о копировании каждого файла.

Решение:

```
Unset
#!/bin/bash

# Запрос исходной и целевой директории у пользователя
read -p "Введите исходную директорию: " source_directory
read -p "Введите целевую директорию: " target_directory

# Запрос расширения файлов, которые нужно скопировать
read -p "Введите расширение файлов для копирования (без точки): "
file_extension

# Проверка существования исходной директории
if [ ! -d "$source_directory" ]; then
    echo "Директория '$source_directory' не существует или недоступна."
    exit 1
fi

# Проверка существования целевой директории
if [ ! -d "$target_directory" ]; then
    echo "Директория '$target_directory' не существует или недоступна."
    exit 1
fi

# Проверка, есть ли файлы с указанным расширением в исходной директории
matching_files=$(find "$source_directory" -maxdepth 1 -type f -name
"*.${file_extension}")

if [ -z "$matching_files" ]; then
```



```
        echo "В директории '$source_directory' нет файлов с расширением
'.$file_extension'."
        exit 1
    fi

    # Копирование файлов с указанным расширением в целевую директорию
    for file in $matching_files; do
        cp "$file" "$target_directory"
        echo "Скопирован файл: $file"
    done

    echo "Готово."
```

Другое решение:

```
Unset
files=$(find "$source_directory" -maxdepth 1 -type f -name "$file_extension")

if [ -z "$files" ];
then
    echo "Файлы не найдены"
    exit 1
fi

# Копирование файлов с указанным расширением в целевую директорию

echo "Введите расширение файлов, которые вы хотите скопировать (например,
txt):"
read file_extension

echo "Введите путь каталога, в который вы хотите скопировать файлы:"
read target_directory
```



```
files=$(find . -type f -name ".*$file_extension")

if [ -z "$files" ];
then
    echo "Файлов с расширением .$file_extension не найдено."
else

    echo "Найденные файлы с расширением .$file_extension:"
    echo "$files"

    echo "Копирование файлов в $target_destination..."
    cp $files "$target_destination"
    echo "Копирование завершено."
Fi
```

Написать скрипт 2 для автоматического commit.

Если текущая директория содержит репозиторий гит, то все изменения фиксируются и автоматически коммитятся с комментарием: "Автоматический коммит \$(date +%Y-%m-%d %H:%M:%S)".

Скрипт 2 в начальном состоянии:

```
Unset
#!/bin/bash

# Этап 1: Проверка текущей директории на наличие репозитория Git

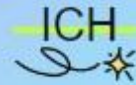
# Этап 2: Добавление изменений в индекс

# Этап 3: Проверка наличия изменений для коммита

# Этап 4: Фиксация изменений и создание коммита

# Этап 5: Вывод сообщений об успешном выполнении или отсутствии изменений
```

Issues для референса:



Проверка текущей директории на наличие репозитория Git

- Действие: Написать скрипт, который будет проверять, находится ли текущая директория в репозитории Git.

Добавление изменений в индекс

- Действие: Разработать команду для добавления всех изменений в индекс Git.

Проверка наличия изменений для коммита

- Действие: Написать проверку наличия изменений для фиксации в репозитории Git.

Фиксация изменений и создание коммита

- Действие: Написать команду для фиксации изменений в репозитории Git с автоматическим созданием коммита, содержащего текущую дату и время.

Вывод сообщений об успешном выполнении или отсутствии изменений

- Действие: Разработать вывод сообщений о результате выполнения скрипта: успешном фиксировании изменений или отсутствии изменений для коммита.

Решение:

```
Unset
#!/bin/bash

# Этап 1: Проверка текущей директории на наличие репозитория Git
if [ ! -d ".git" ]; then
    echo "Текущая директория не является репозиторием Git."
    exit 1
fi

# Этап 2: Добавление изменений в индекс
git add .
```




```
# Этап 3: Проверка наличия изменений для коммита
if git diff-index --quiet HEAD --; then
    echo "Нет изменений для фиксации."
    exit 0
fi

# Этап 4: Фиксация изменений и создание коммита
git commit -m "Автоматический коммит $(date +%Y-%m-%d %H:%M:%S)"

# Этап 5: Вывод сообщений об успешном выполнении или отсутствии изменений
echo "Изменения успешно зафиксированы и сохранены."
exit 0
```

