Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №2**

Дисциплина: «Разработка профессиональных приложений»

Двумерные структуры данных

Вариант № 22

Выполнил:

студент группы ИВТАПбд-21

Старостин К. А.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков Ильнур Ирекович

Ульяновск, 2023

# Задание

В ходе выполнения лабораторной работы необходимо познакомиться с технологией Git, с его базовыми функциями, понять принцип работы данной технологии.

# Ход работы

## Создание локального репозитория

Для создания локального репозитория используется команда

git init

Перед созданием необходимо определиться с местом нахождения папки. C:\Users\rdera\PycharmProjects\lab 5. Переходим в папку lab 5:

cd C:\Users\rdera\PycharmProjects\lab 5

И создаём репозиторий.

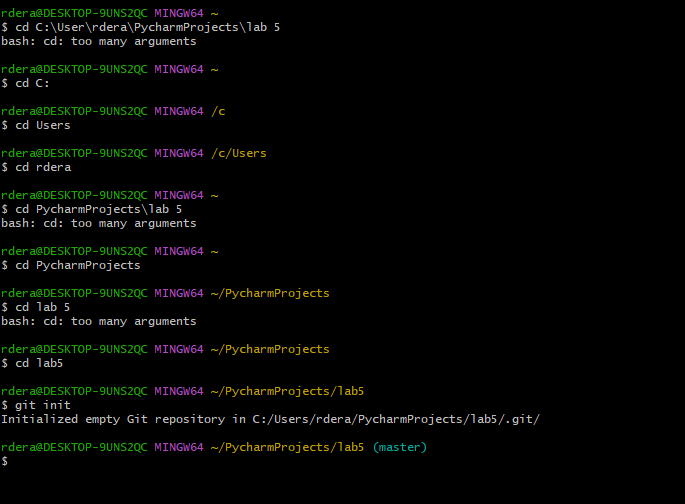


Рисунок Создание локального репозитория

## Добавить несколько файлов

Чтобы добавить файлы нужно запустив команду git add, указав индексируемые файлы, а затем выполнив git commit:

$ git add .

$ git commit –m ‘commit’

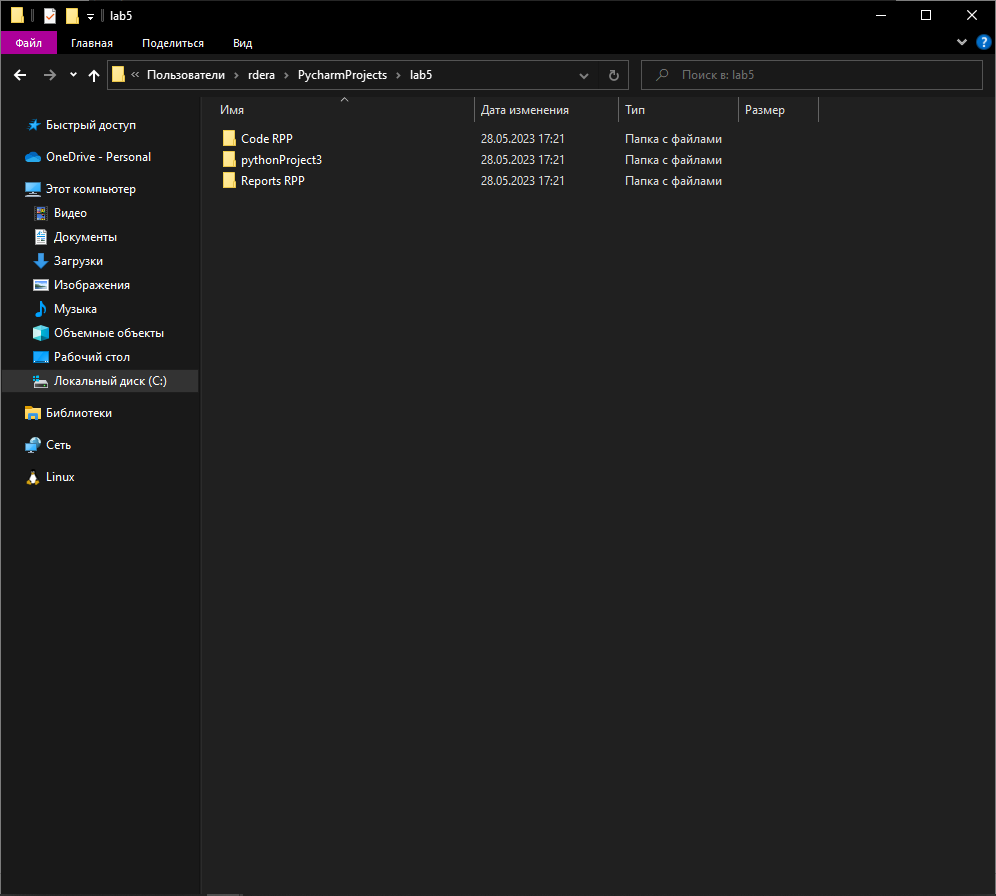


Рисунок репозиторий с добавленными файлами

## Сделать игнорируемый файл

Через Git Bash необходимо создать файл .gitignore:

touch .gitignore

Открыть созданный файл через любой текстовый редактор и вписать туда название игнорируемого файла.

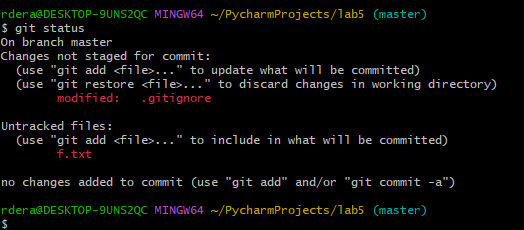


Рисунок До игнорирования

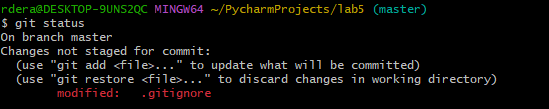
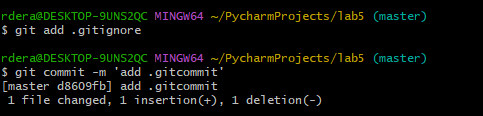


Рисунок После игнорирования

## Фиксация локального изменения

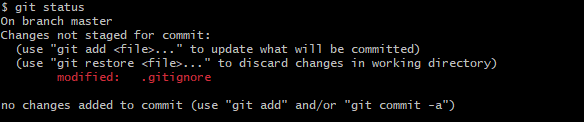
После добавления файла в .gitignore необходимо зафиксировать изменения.



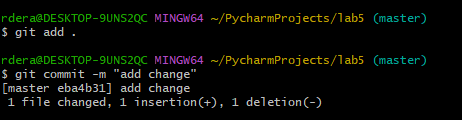
После выполнения этих команд изменения будут зафиксированы.

## Внесение и фиксация изменений

В файл .gitignore были внесены изменения, об этом сообщает на консоль:



Необходимо зафиксировать изменения в файле.



Если файл не один, то вместо названия файла в команде «git add» можно использовать точку. После фиксируем изменения с помощью

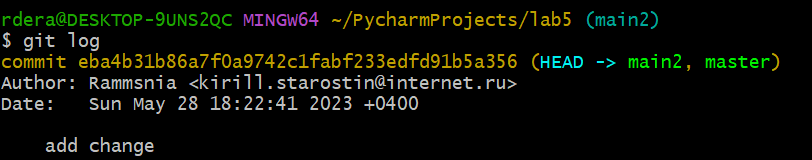
git commit –m “name”

## Создание нескольких бранчей, редактирование файлов в разных бранчах и фиксация изменений. Слияние двух веток в одну

Создание новой ветки и переход к ней:

### 

Чтобы проверить, что указатель находится на ветке “main2”, можно ввести git log:



Чтобы осуществить слияние двух веток, необходимо выполнить команду git merge, указав ветки, которые нужно слить.

### 

## Отправка исходных кодов в удалённый репозиторий

Добавляем удалённый репозиторий в список доступных репозиториев для локального репозитория.

### 

git branch -M master: Команда переименовывает текущую ветку в "master". Опция -M означает "force rename" и переименовывает текущую ветку даже если уже существует ветка с таким именем.

### 

git push -u repos master: Команда отправляет изменения из локального репозитория на удаленный репозиторий “repos” в ветку “master”. Опция -u означает “set upstream” и устанавливает соответствие между локальной веткой “master” и удаленной веткой “master” на удаленном репозитории “repos”.

## Клонирование удаленный репозиторий в папку, отличную от оригинальной.

Для клонирования удаленного репозитория Git в папку, отличную от оригинальной, можно использовать команду git clone и указать путь до папки, в которую нужно склонировать репозиторий.

### 

## Добиться конфликта файлов при применении изменений в оригинальном репозитории.

Конфликт в Git - это ситуация, когда два или более пользователей одновременно изменяют один и тот же файл в разных ветках, и при попытке слияния этих веток Git не может автоматически решить, какое изменение следует принять. В этом случае Git создает конфликтный файл, который нужно разрешить вручную, чтобы восстановить целостность репозитория.

Создать конфликт можно, если в ветке master и dev, между которыми было выполнено слияние, создать два файла. Но в ветке dev меняем содержимое файла.

При повторном слиянии двух веток происходит конфликт:

### 

## Решение конфликта

Проверив статус, можно увидеть, что имеются конфликты:

### 

Открыв файл в редакторе, можно увидеть следующее:

### 

Теперь необходимо отредактировать файл так, как он должен выглядеть и сохранить. После добавляем его на индексацию и делаем коммит.

Комментарий к коммиту писать не нужно, так как был конфликт, будет предложен комментарий, который описывает происходящий конфликт:

### 

Принято использовать именно этот комментарий, поэтому сохраняем файл и выходим.

В логах также видно, что был конфликт.

### 