**Звіт**

**Лабораторної роботи №1**

**З предмету**

**«Технології програмування»**

**Студента МІТ-31**

**Богдан Лапшин**

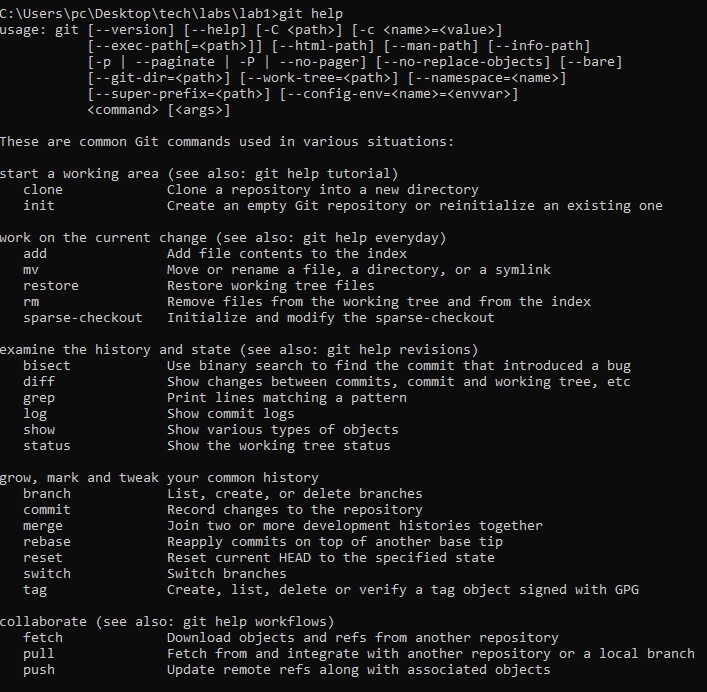
**«ОСНОВИ GIT»**

*Мета: розглянути призначення та можливості систем контролю версій*

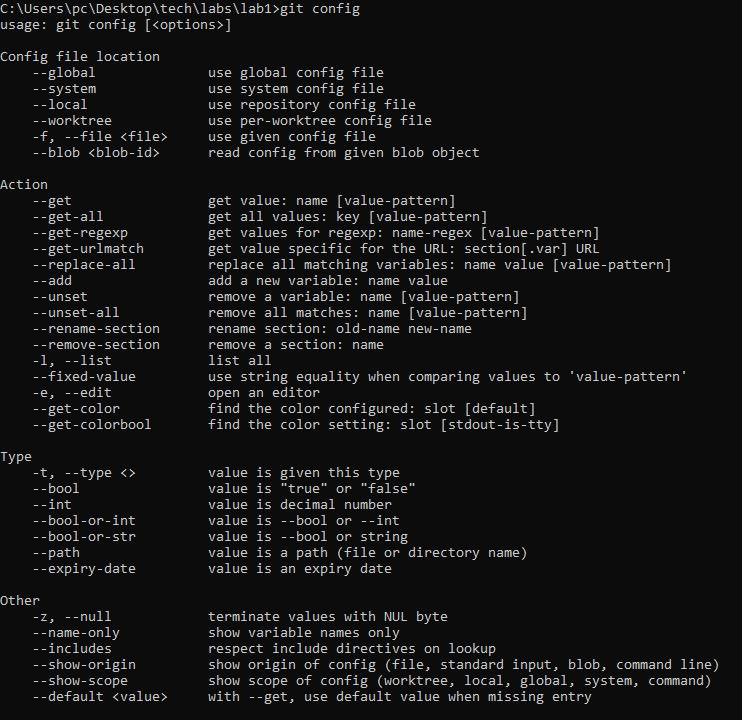
*програмного забезпечення. Ознайомитися з Git та навчитися виконувати базові*

*операції в Git.*

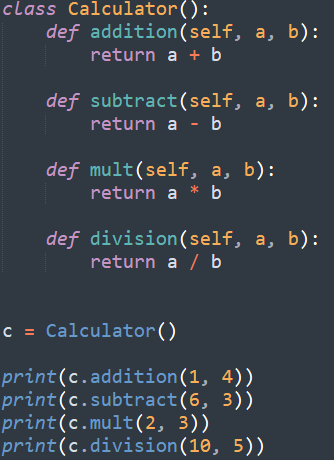
**Хід роботи**



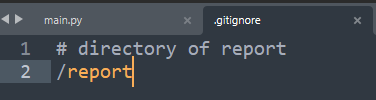
Git help показывает все команды



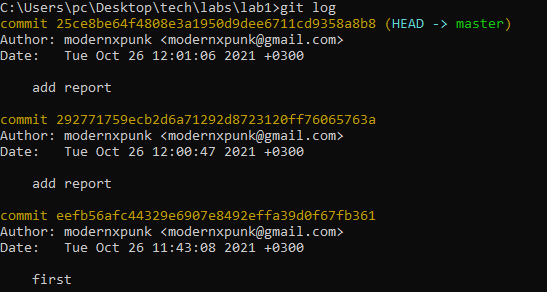
Git config показывает внутренние настройки типа какое у вас имя, если настроили глобальное имя, какая ветка по умолчанию стоит и не только



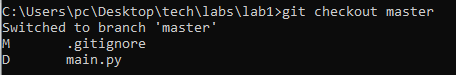
Код калькулятора на python в файле main.py



Файл .gitignore, директория report это отчёт вам ☺

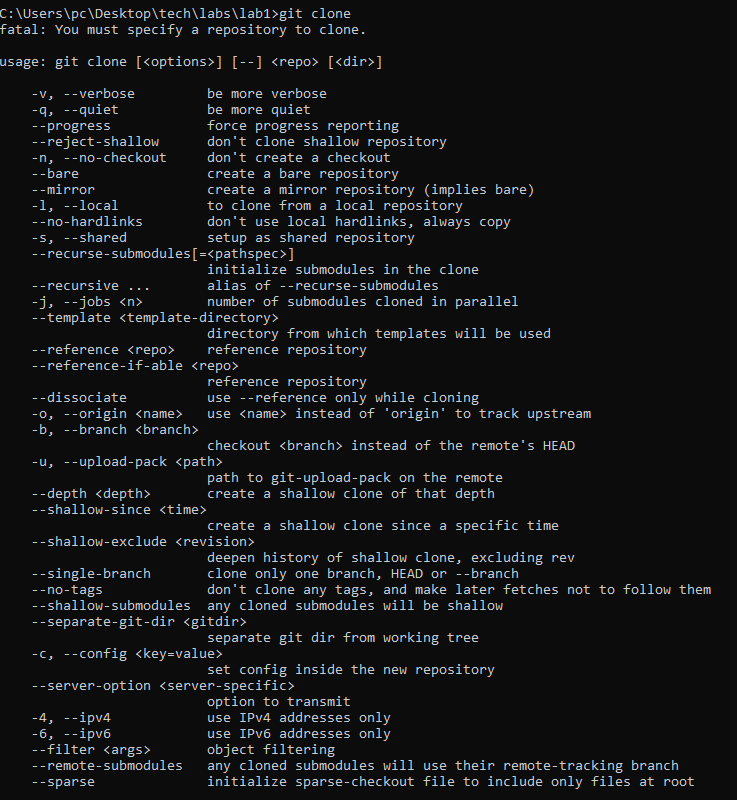


Все мои попытки закоммитить можно получить с помощью git log

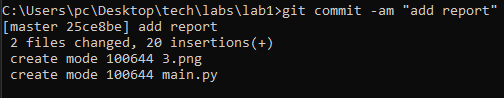


C:\Users\pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\checkout.png

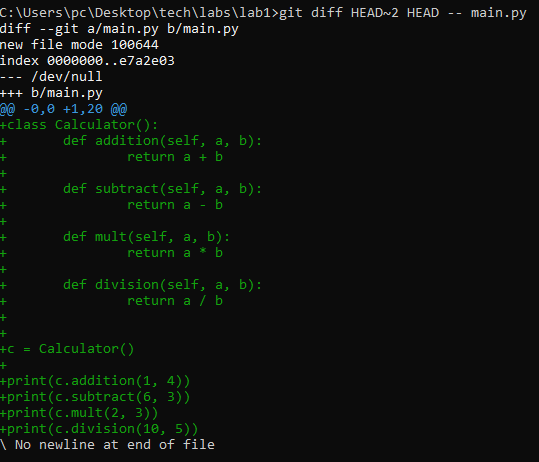
Если хотите создать новую ветку, то можно с помощью git checkout –b название\_ветки, так же можно переключиться на другую ветку с помощью git checkout название\_ветки или git branch название\_ветки



С помощью git clone можно скачать репозиторий, но я не стал, сложно стало искать репозиторий с размером меньше 10мб



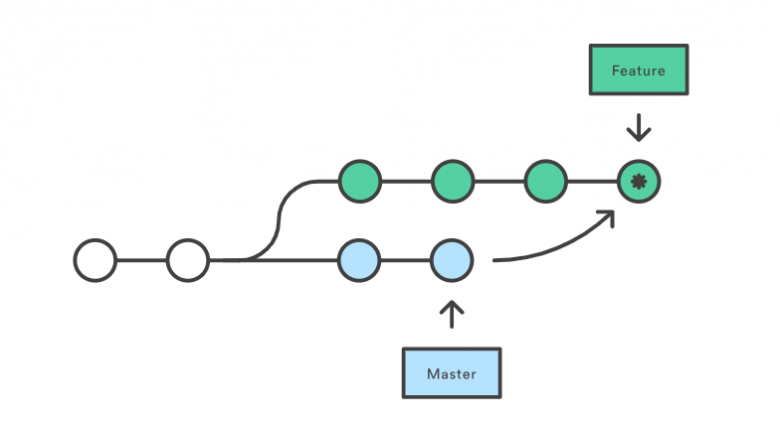
Самая главная команда, так сказать. Чтобы закомитить достаточно и git commit –m ‘message’ но перед этим надо добавить в staging area, она нужна чтобы выбирать что надо отправить, это можно сделать с помощью git add . если все ваши нужные файлы в текущем репозитории или git add –all, чтобы пометить все файлы в месте инициализации, а потом уже сделать коммит



Есть много обозначений для сравнения с текущем или с предыдущими коммитами, например чтобы сравнить текущей файл в текущем коммите с файлом двумя коммитами ранее можно написать HEAD~2 HEAD или HEAD^^ HEAD

C:\Users\pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\merge.png

Merge объединяет два коммита в общую историю, часто используется чтобы «слить» две ветки, примерно так.



C:\Users\pc\AppData\Local\Microsoft\Windows\INetCache\Content.Word\status.png

Git status отображает статус текущего каталога, с помощью неё можно проверить есть ли не проиндексированные файлы, они подсвечиваются красным, в другом случае они зеленые, так же может оказаться, что нечего индексировать. Когда файл вноситься или изменяется, то сразу можно его увидеть в git status красным, но его может и не быть, если он содержится в файле .gitignore, с помощью этого файла можно говорить не следить за этими файлами, часто так делает, если файл очень много весит.

https://github.com/modernxpunk/tech-of-programming