**Прізвище:** Ігнатенко

**Ім'я:** Владислав

**Група:** КН-416

**Варіант:** 6

**Кафедра:** САПР

**Дисципліна:** Управління ІТ-проектами

**Перевірив:** Ханас Ю.Ю.

**ЗВІТ**

до лабораторної роботи №1

на тему "Системи контролю версій"

**Мета роботи:** ознайомитись з системами контролю версій. Зрозуміти принципи синхронізації робочих груп у гнучких командах. Набути навичок використання технічних засобів та протоколів для роботи з системами контролю версій.

**Короткі теоретичні відомості:**

Система контролю версій (від англ. Version Control System, VCS або Revision Control System)

– програмне забезпечення для полегшення роботи з інформацію, що змінюється. Система контролю версій дозволяє зберігати декілька версій одного й того ж документа, при необхідності звертатись до більш ранніх версій, визначати, хто й коли зробив ті чи інші зміни, та багато іншого. Такі системи найбільш широко використовуються при розробці програмного забезпечення для зберігання вихідних кодів розроблюваної програми. Однак вони можуть з успіхом застосовуватися і в інших областях, в яких ведеться робота з великою кількістю електронних документів, що неперервно змінюються. Зокрема, системи контролю версій застосовуються в САПР, зазвичай в складі систем управління даними про виріб (PDM). Контроль версій використовується в інструментах конфігураційного управління (Software Configuration Management Tools).

Git (вимов. «гіт») – розподілена система керування версіями файлів. Проект був створений Лінусом Торвальдсом для управління розробкою ядра Linux. На сьогоднішній день підтримується Джуном Хама. Система спроектована як набір програм, спеціально розроблених з урахуванням їх використання в скриптах. Це дозволяє зручно створювати спеціалізовані системи контролю версій на базі Git або користувацькі інтерфейси. Наприклад, Cogito є саме таким прикладом фронтенда до репозиторіїв Git, а StGit використовує Git для управління колекцією патчів. Git підтримує швидке розділення і злиття версій, включає інструменти для візуалізації та навігації по нелінійній історії розробки. Git надає кожному розробнику локальну копію всієї історії розробки; зміни копіюються з одного репозиторію в іншій.

**Індивідуальне завдання:**

1. Ознайомитись з принципами роботи систем контролю версій.

2. Отримати індивідуальне завдання у викладача.

3. Написати програму згідно з індивідуальним завданням.

4. Створити репозиторій та завантажити туди свій програмний код.

5. Внести зміни в програму, при необхідності провести вирішення конфліктів.

**Аналіз отриманих результатів:**

Для початку було створено репозиторій на github.com на рис. 1.

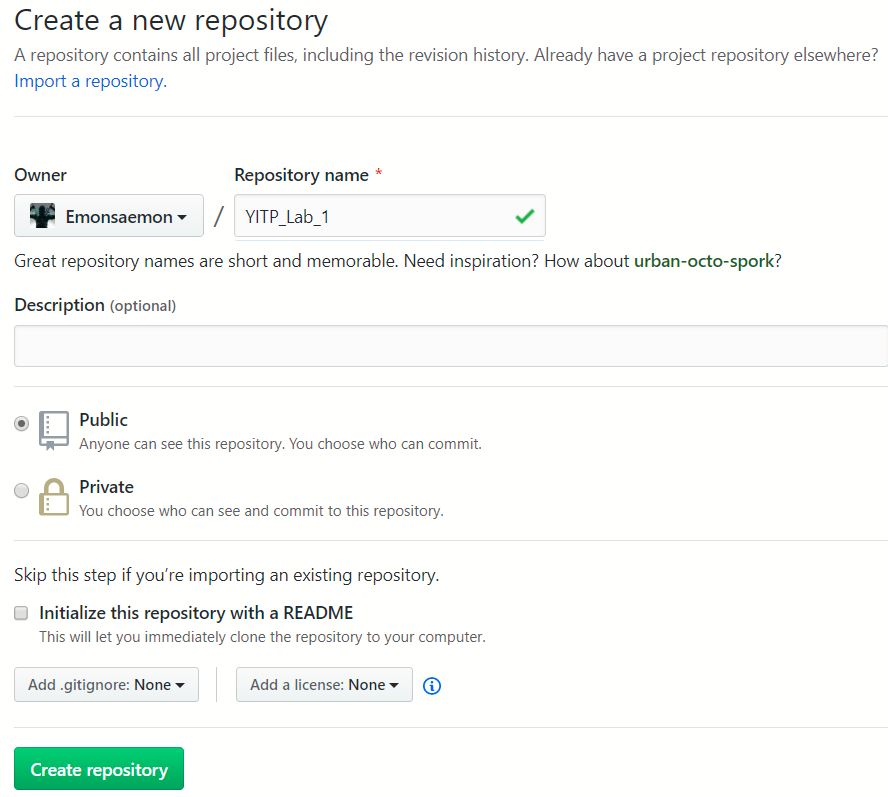
****

Рис. 1. Створений репозиторій на гітхабі

Далі створено папку на комп’ютері та за допомогою команди git init створюємо репозиторій на комп’ютері. Додаємо віддалений репозиторій за посиланням на гітхабі командою git remote add на рис. 2.

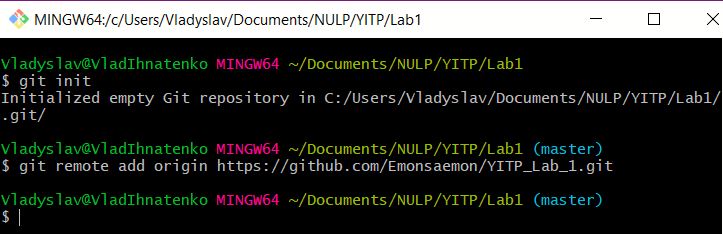
****

Рис. 2. Створення репозиторію на комп’ютері та додавання віддаленого репозиторія

Далі створюємо локальний коміт для внесення локальних змін в репозиторії та згодом відправляємо їх на гілку мастер у віддаленому репозиторії на рис. 3.

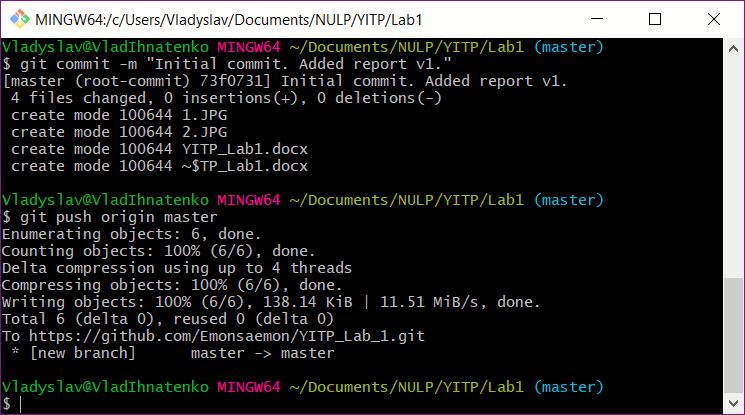


Рис. 3. Внесення змін в локальний та віддалений репозиторій

Результатом виконаних дій є створені файли в репозиторії гітхабу на рис. 4.

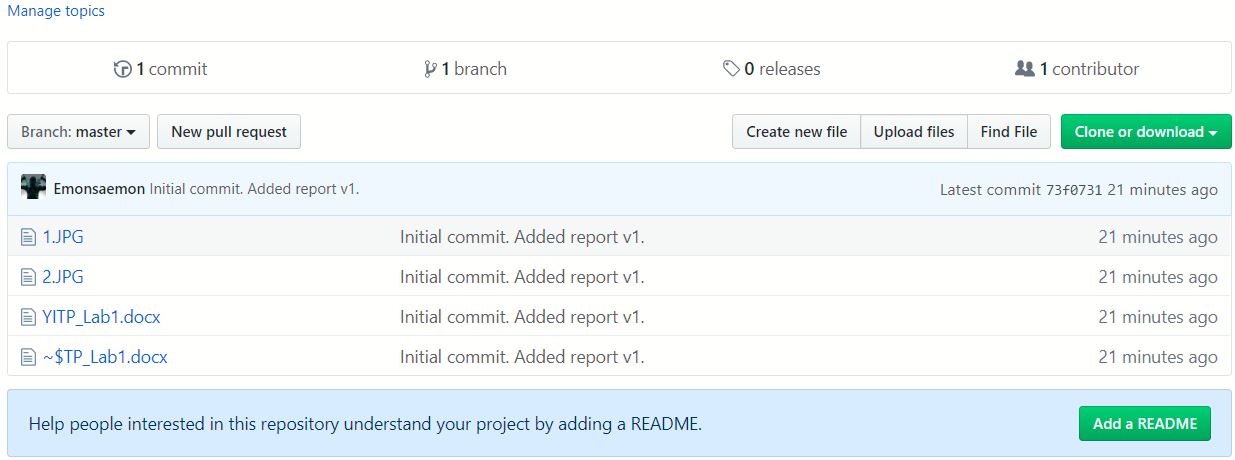


Рис. 4. Результат виконання команд

**Висновки:**