**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ В.Н. КАРАЗІНА**

**ФАКУЛЬТЕТ КОМП’ЮТЕРНИХ НАУК**

**КАФЕДРА БЕЗПЕКИ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ**

**Лабораторна робота № 1**

**з дисципліни навчальної дисципліни**

**«Математичні методи та технології тестування та верифікації програмного забезпечення»**

Виконала:

студентка групи КС-23

Рузудженк Сабіна

Перевірив:

Доцент

Нарєжній О.П

Харків – 2019

**Лабораторная работа № 1**

з навчальної дисципліни

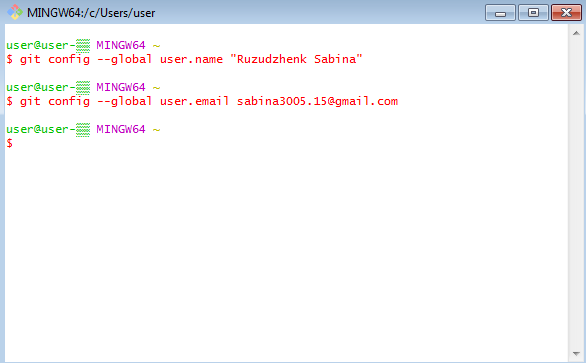
«Математичні методи та технології тестування та верифікації програмного забезпечення»

**Тема роботи:** Вивчення системи контролю версії на прикладі Git

**Мета роботи:** Отримання базових знань про Git та його відмінності від централізованих систем контролю версій. Отримати робочу версію Git на комп’ютері, налаштувати та персоналізувати її.

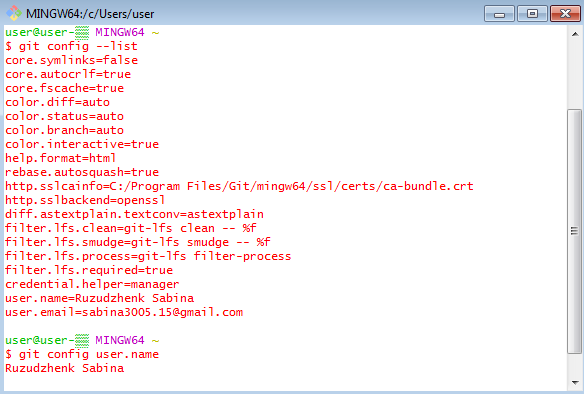
**Хід роботи**

Для початку роботи необхідно встановити Git на комп’ютер. Після вдалого завершення процесу встановлення має з’явитися відповідне вікно консолі. У ньому ми проводимо первісне налаштування версії Git для її персоналізації (зокрема, вводимо прізвище та ім’я, а також адресу електронної пошти **(Рис. 1)**)**.**



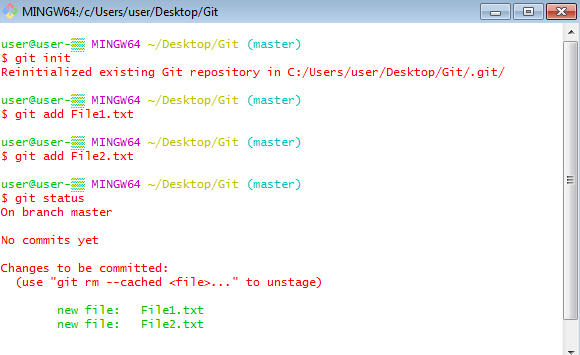
**Рис.1.** Первісне налаштування Git

Для перевірки налаштувань вводимо команду *$ git config –list* **(Рис. 2).**

Для перевірки імені користувача необхідно ввести команду *$ git config user.name* **(Рис. 2).**

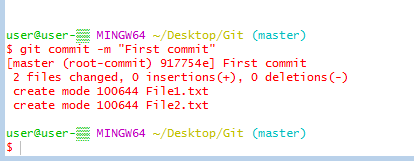
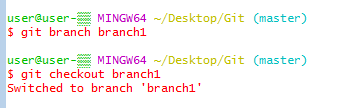
**Рис. 2.** Перевірка налаштувань Git

Далі створюємо нову папку та ініціалізуємо в ній Git–репозиторій. Для подальшої роботи додаємо до проекту делька текстових файлів **(Рис. 3).**

**Рис. 3.** Створення Git–репозиторію

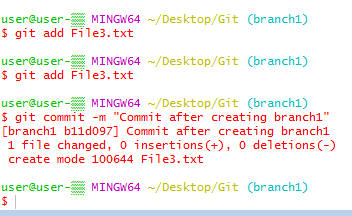
Створюємо перший комміт шляхом введення команди *$ git commit -m "First commit"***(Рис. 4).**

**Рис.4.** Створення першого комміту

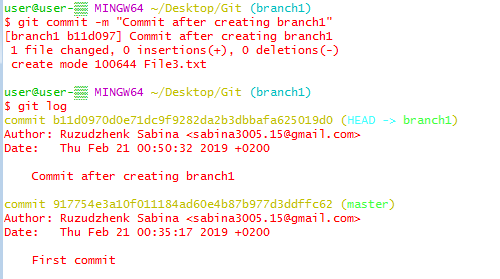
Для створення гілок, які відокремлюють користувача від основної лінії розробки, що дозволяє незалежну роботу виконаємо такі дії: створюємо нову гілку за допомогою команди *$ git branch branch1*; перемикаємося на іншу гілку за допомогою команди *$ git checkout branch1* **(Рис. 5).**

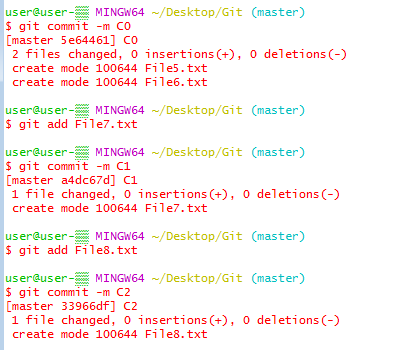
**Рис. 5.** Створення нової гілки та перемикання на неї

Також створюємо необхідні комміти на новій гілці **(Рис. 6).**

Рис. 6. Створення комміту на новій гілці

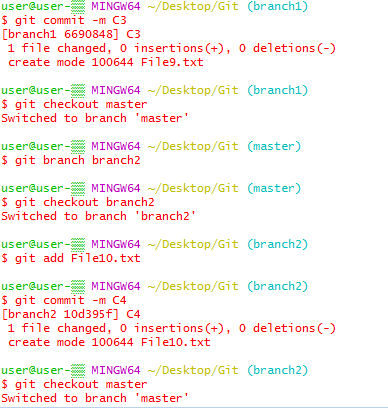
Доступ до історії створених коммітів отримуємо за допомогою команди *$ git log* **(Рис. 7).**

Рис. 7. Історія створених коммітів

Далі знову переключаємося на основну гілку за допомогою команди *$ git checkout master* та створюємо там комміти С0, С1, С2 **(Рис. 8).**

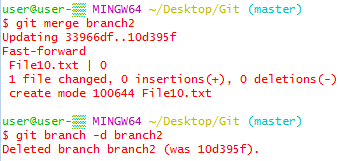
**Рис. 8.** Створення нових коммітів

На гілці branch1 створюємо комміт С3, фіксуємо зміни на даній гілці, повертаємося до основної гілки та створюємо нову гілку branch2, а на ній – комміт С4. А потім повертаємося до гілки master **(Рис. 9).**

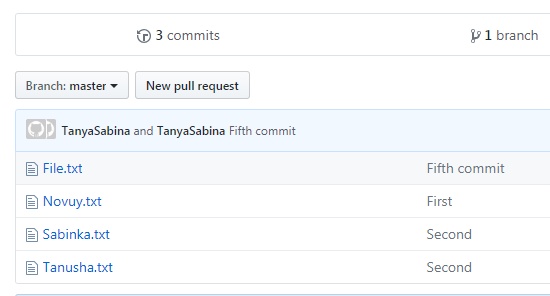
****

**Рис. 9.** Фіксація змін та створення нової гілки

Виконуємо злиття гілок та видаляємо гілку branch2 **(Рис. 10).**

**Рис. 10.** Злиття гілок

Проводимо реєстрацію на <https://github.com/>. Для переміщення проекту на Github–репозиторій виконуємо команди: *$ git remote add origin* [*https://github.com/sabina30r/QualityAssurance*](https://github.com/sabina30r/QualityAssurance)та *$ git push -u origin master.* Після виконання цих команд створений проект, його файли завантажені до репозиторію на Github.com. **(Рис. 11).**

Рис. 11. Завантаження проекту на Github–репозиторій

**Висновки**

Отже, Git – це система контролю версій, яка відстежує зміни у файлі, або файлах протягом певного часу, а також дозволяє через будь-який час повернутися до окремої попередньої версії. Тому, користувач даної системи має можливість відстежувати дії, проведені з файлом, або набором файлів, незалежно від того, ким вони були здійснені. Під час роботи з Git користувач має змогу повернути файл у початковий стан, отримати попередні версії.

Завдяки гіллястій структурі Git забезпечує безперебійну роботу без втручання на інші лінії.

Також, однією з переваг Git є можливість змінити налаштування за потреби поточного користувача, а також критеріями, що вимагає вид його діяльності.

Таким чином, у ході виконання лабораторної роботи було проведено ознайомлення з централізованою системою контролю версій Git, його налаштування та персоналізація. У результаті роботи було вдало створено цілком робочий проект, що наразі переміщений на Github-репозиторій.