Міністерство освіти і науки України

Національний Університет «Запорізька Політехніка»

Кафедра програмних засобів

**ЗВІТ**

з лабораторної роботи №1

з дисципліни «Верифікація цифрових систем» на тему:

«ОСНОВИ GIT»

Виконала:

студентка групи КНТ-227 А.О.Колташева

Прийняв:

доцент Т.І.Каплієнко

**Мета роботи**

Вивчити основні команди по роботі з локальним репозиторієм GIT.

**Завдання на лабораторну роботу**

1. Ознайомитися з теоретичними відомостями щодо основних команд GIT.

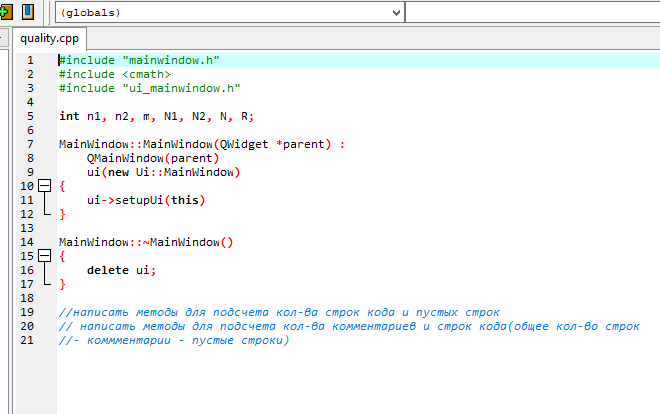
2. Виконати свій варіант завдання, аналогічно нульовому варіанту:

**Варіант 2**

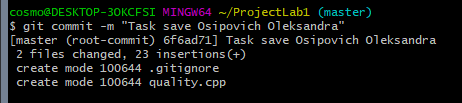
Ініціалізувати каталог з майбутнім проектом в git. Створити файл .gitignore з описом ігнорованих файлів (\*. [oa], \* ~). Користувач вводить свої дані, створює папку проекту, створює в ній файл із завданнями. Виконує комміт для фіксації внесених змін. Створює нову гілку, відкриває файл і виконує завдання, виконує комміт. Повертається в гілку майстер, додає аліаси для branch, checkout, status, history. За допомогою аліасів переглядає всі гілки проекту, його поточний статус та історію коммітів.

**Виконання роботи**

1. Створюємо каталог з майбутнім проектом та переходимо у нього.
2. Створюємо файл .gitignore з описом ігнорованих файлів.
3. Користувач вводить свої дані.
4. Створюємо репозиторій для проекта.
5. Додаємо у нього файли.



1. Робимо комміт.



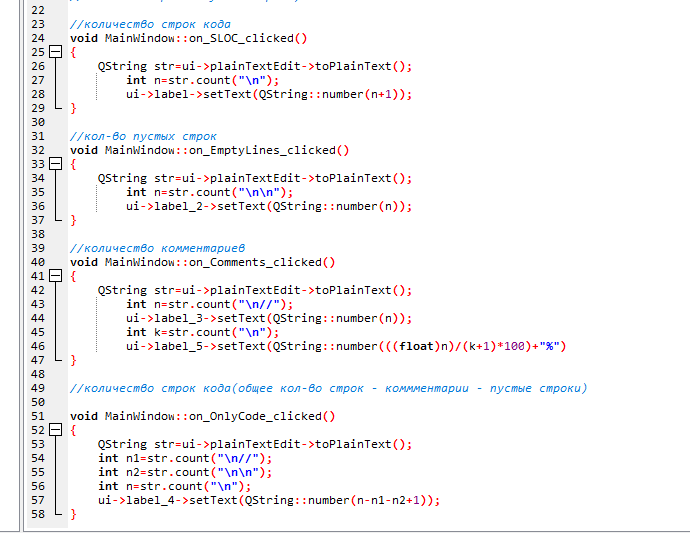
1. Створюємо нову гілку.



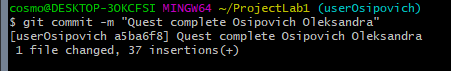
1. Переходимо в нову гілку.



1. Виконуємо завдання.



1. Робимо комміт.



1. Повертаємось в гілку master.



1. Додаємо аліаси для branch, checkout, status, history.



а) встановлюємо аліас br для branch



б) встановлюємо аліас co для checkout

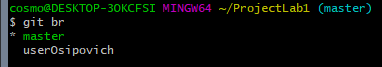


в) встановлюємо аліас st для status

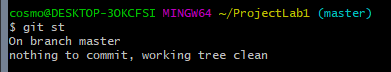


г) встановлюємо аліас hs для history

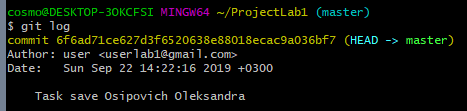
1. Переглядаємо всі гілки проекту



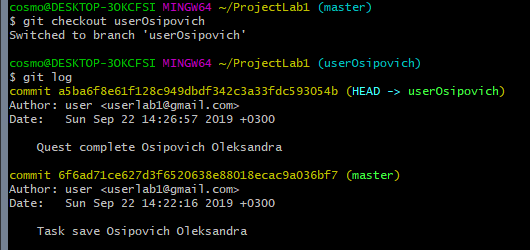
1. Переглядаємо поточний статус



1. Переглядаємо історію коммітів

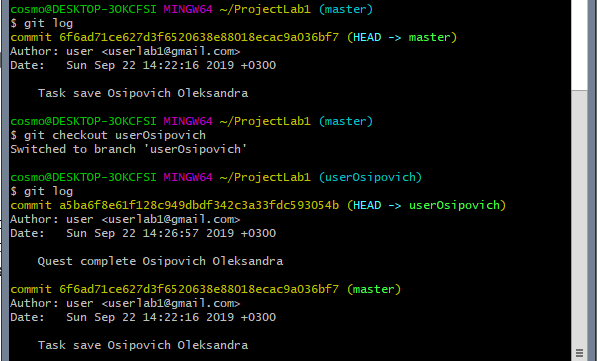
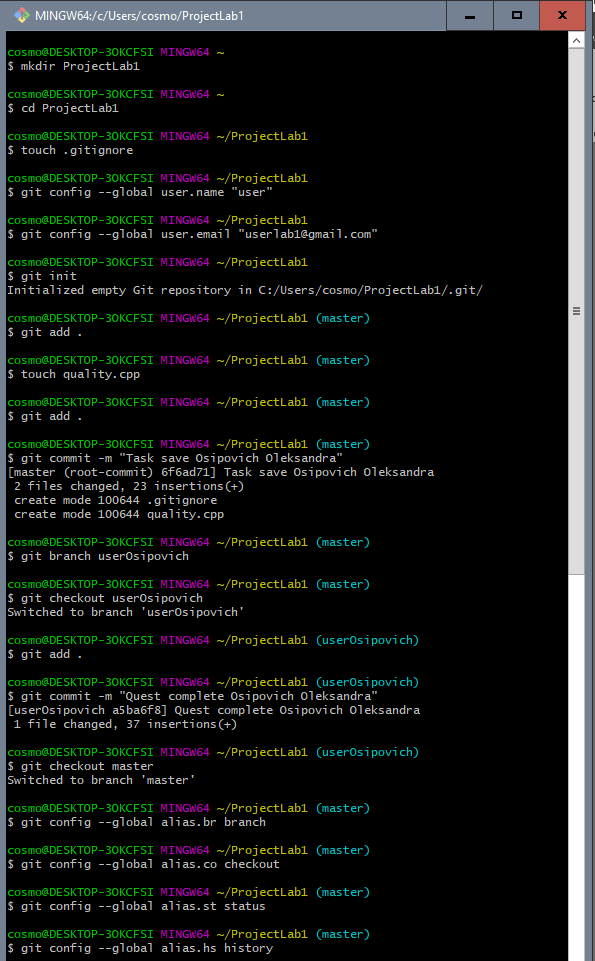
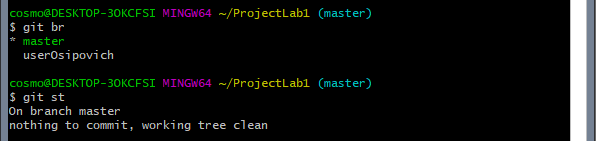


а) комміти в гілці master



б) комміти в гілці userOsipovich

**Повний хід роботи**



**Висновки**

В результаті виконання лабораторної роботи були вивчені основні команди по роботі з локальним репозиторієм GIT.

GIT-репозиторій створюється за допомогою команди **git init**.

Для того щоб почати відстежувати (додати під версійний контроль) новий файл, використовується команда **git add**.

У git файли можуть перебувати в одному з **трьох** станів:

- зафіксованому (**git directory**)- файл вже збережено у вашій локальній базі;

- зміненому (**working directory**) – файли змінилися, але ще не були зафіксовані;

- підготовленому (**staging area**) - змінені файли, відмічені для включення в наступний комміт.

Для того, щоб перевірити поточний стан репозиторію використовується команда **git status**.

**.gitignore** працює для вказівки в ньому файлів і папок, які необхідно приховати від системи контролю версій git. Як правило, приховують файли або папки, які автоматично генерують свій вміст, або мають конфігураційні параметри, які можуть відрізнятися у тих, хто співпрацює над проектом.

Команда **git diff** показує непроіндексовані зміни зроблені після останнього комміту.

Найпростіший спосіб зафіксувати зміни - команда **git commit –m «Коментар до комміту».**

Для простого видалення файлу з репозиторію використовується команда **git rm.**

Для того, щоб видалити файл з репозиторію, залишивши його при цьому в робочому каталозі використовується команда **git rm --cached "Назва файлу"**

Команда **git mv "Назва файлу"** **"Каталог призначення"** дозволяє переміщувати файли.

Переглянути історію коммітів дозволяє команда **git log.** За замовчуванням, без аргументів, **git log** виводить список коммітів створених в даному репозиторії в зворотному хронологічному порядку.

Змінити повідомлення останнього комміту дозволить команда

**git commit --amend.**

Ця команда відкриє в текстовому редакторі повідомлення останнього комміту для виправлення.

Щоб створити гілку і відразу ж перейти на неї, використовується команда **git checkout -b**.

Для створення гілки без переходу в неї виконується команда **git branch.**

Для об’єднання створених гілок використовується команда **git merge**.