

Лабораторна робота №8

Тема: Створення та видалення тегів.

Хід роботи

1. Перевірити глобальні налаштування Git for Windows.
2. У випадку невідповідності вашим даним налаштувати глобальну конфігурацію Git for Windows.
3. Клонувати створений раніше репозиторій як локальну робочу копію.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git (main)
$ git clone https://github.com/Julia139/Work.git
Cloning into 'Work'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
Receiving objects: 52% (11/21), 52.00 KiB | 7.00 KiB/s
```

4. За допомогою консольного інструменту git tag додати теги для поточної ревізії.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git tag v1.0

Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git show v1.0
commit a08ae8a257afd72c67292aee7b4792cc6b6a6ca1 (HEAD -> main, tag: v1.0, origin/main)
Author: Julia139 <juliyaaa13a@gmail.com>
Date: Tue Oct 17 10:59:46 2023 +0300

    first commit

diff --git a/README.md b/README.md
index e42978c..2abac14 100644
--- a/README.md
+++ b/README.md
@@ -1,2 @@
 # project-practicum
+# project-practicum


Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$
```

5. Зробити зміни в локальній робочій копії, зафіксувати зміни та додати новий тег для нової ревізії.
6. Перевірити наявні теги за допомогою інструменту git tag.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git add LP8_3air.pdf
git commit -m "new laba"
git tag v2.0
[main b399d4f] new laba
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "Work/\320\233\320\2408_\320\227\320\262\321\226\321\202.pdf"

Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git tag
v1.0
v2.0

Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$
```

					Лабораторна робота №8			
Зм	Лист	№ докум	Підпис	Дата	Створення та видалення тегів.	Літ.	Лист.	Листів
Розробив	Недашковська Ю.В.					У	1	3
Перевірів	Левицький В.М.							
Оцінка						Група 451		
Затв.								

7. Перемкнутися за допомогою тега на попередій раніше створений комміт.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git checkout v1.0
Note: switching to 'v1.0'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

    git switch -c <new-branch-name>

Or undo this operation with:

    git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at a08ae8a first commit
D    README.md
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work ((v1.0))
$ |
```

8. Повернутися на останню версію репозиторію.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work ((v1.0))
$ git checkout main
Previous HEAD position was a08ae8a first commit
Switched to branch 'main'
D    README.md
Your branch is ahead of 'origin/main' by 1 commit.
(use "git push" to publish your local commits)
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ |
```

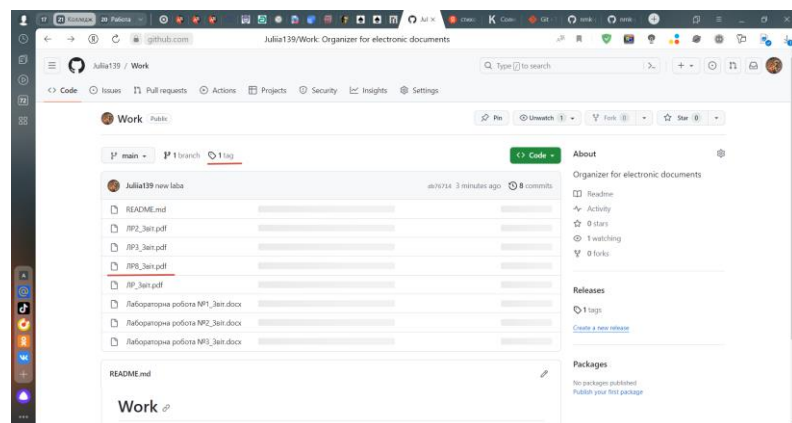
9. Вивантажити зафіксовані зміни на Github.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 633.59 KiB | 30.17 MiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
```

10. Вивантажити створені теги на Github.

```
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Julial139/Work.git
 * [new tag]         v2.0 -> v2.0
Admin@DESKTOP-9L3NP54 MINGW64 ~/dir_for_git/Work (main)
$ |
```

11. Перевірити наявність нових файлів на Github через web-інтерфейс.

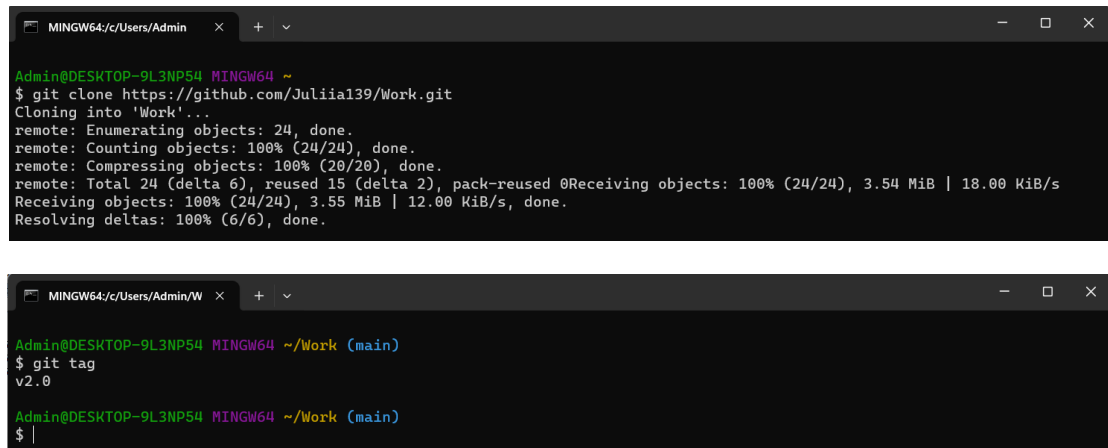


Зм.	Арк.	№ докум.	Підпис	Дата

Лабораторна робота № 8

Ар
2

12. Клонувати репозиторій в інше місце та перевірити наявність тегів.



```
Admin@DESKTOP-9L3NP54 MINGW64 ~
$ git clone https://github.com/Julia139/Work.git
Cloning into 'Work'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 24 (delta 6), reused 15 (delta 2), pack-reused 0Receiving objects: 100% (24/24), 3.54 MiB | 18.00 KiB/s
Receiving objects: 100% (24/24), 3.55 MiB | 12.00 KiB/s, done.
Resolving deltas: 100% (6/6), done.

Admin@DESKTOP-9L3NP54 MINGW64 ~/Work (main)
$ git tag
v2.0

Admin@DESKTOP-9L3NP54 MINGW64 ~/Work (main)
$ |
```

13. Для кожного етапу роботи зробити знімки екрану та додати їх у звіт з описом кожного скріншота.

14. Дати відповіді на контрольні запитання.

15. Зберегти звіт у форматі PDF.

16. Завантажити збережений PDF у репозиторій для лабораторних робіт.

Контрольні питання

1. За допомогою якої команди можливо переглянути список наявних тегів в Git?

Для перегляду списку наявних тегів в Git можна використати команду `git tag`.

2. Які два основних типи тегів використовує Git?

Git використовує два основних типи тегів: легкі та анотовані. Легкі теги є простими покажчиками на конкретний комміт, без додаткової інформації. Анотовані теги є публічними і містять додаткову метадану, таку як ім'я, електронна адреса та дата авторства.

3. Розкажіть про кожен тип тегів Git

Легкі теги – це простий покажчик на конкретний комміт, без додаткової інформації. Вони корисні для швидких або внутрішніх посилань, але не мають переваг, які надають анотовані теги. **Анотовані теги** – це публічні і мають додаткову метадану, таку як ім'я, електронна адреса та дата авторства. Вони зберігаються як повноцінні об'єкти в базі даних Git.

4. Чи можливо позначати вже пройдені комміти?

Так, можливо позначати вже пройдені комміти за допомогою команди `git tag`. Наприклад, `git tag v1.0 <commit>` позначить комміт з хешем `<commit>` як версію 1.0.

5. Для видалення тега в локальному репозиторії необхідно виконати команду?

Для видалення тега в локальному репозиторії необхідно виконати команду `git tag -d <tag>`. Ця команда видалить локальний тег з назвою `<tag>`.

					Лабораторна робота № 8	Ар
						2
Зм.	Арк.	№ докум.	Підпис	Дата		