

Лабораторна робота № 8

Тема: Створення та видалення тегів.

Мета: Отримати навички додавання тегів до ревізій проекту в СКВ Git.

Обладнання та ПЗ: Персональний комп'ютер.

Хід роботи та завдання

Пункт 1.

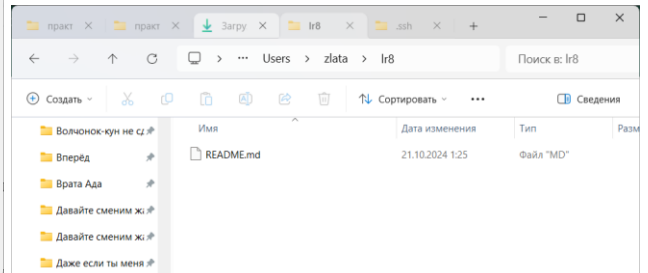
Перевірка глобальних налаштувань Git for Windows.

```
MINGW64/c/Users/zlata
zlata@Zlata MINGW64 ~
$ git config --global --list
user.name=zlata
user.email=alwaysssmile981@gmail.com
zlata@Zlata MINGW64 ~
$
```

Пункт 3.

Клоновано створений раніше репозиторій як локальну робочу копію.

```
MINGW64/c/Users/zlata
zlata@Zlata MINGW64 ~
$ git config --global --list
user.name=zlata
user.email=alwaysssmile981@gmail.com
zlata@Zlata MINGW64 ~
$ mkdir lr8
zlata@Zlata MINGW64 ~
$ cd lr8
zlata@Zlata MINGW64 ~/lr8
$ git clone https://github.com/Zlata451/lr8
Cloning into 'lr8'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
zlata@Zlata MINGW64 ~/lr8
$ cd -
/c/Users/zlata
```



					Лабораторна робота № 8			
Зм	Лист	№ докум	Підпис	Дата	Створення та видалення тегів	Літ.	Лист.	Листів.
Розробив		Ізмestьєва				У	3	5
Перевірів		Левицкий				Група 451		
Оцінка								
Затв								

Пункт 4.

Створено новий тег для поточної ревізії.

```
MINGW64/c/Users/zlata/lr8
fatal: not a git repository (or any of the parent directories): .git

zlata@Zlata MINGW64 ~
$ AC

zlata@Zlata MINGW64 ~
$ cd lr8

zlata@Zlata MINGW64 ~/lr8 (master)
$ git tag -a v1.0 -m "Перша версія"

zlata@Zlata MINGW64 ~/lr8 (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean

zlata@Zlata MINGW64 ~/lr8 (master)
$ git tag
v1.0

zlata@Zlata MINGW64 ~/lr8 (master)
$
```

Пункт 5, 6.

Внесено зміни в локальній робочій копії, зафіксовано ці зміни, а також додано новий тег для нової ревізії; перевірено наявні теги.

```
MINGW64/c/Users/zlata/lr8

zlata@Zlata MINGW64 ~
$ git config --global --list
user.name=Zlata
user.email=alwayssmile981@gmail.com

zlata@Zlata MINGW64 ~
$ cd lr8

zlata@Zlata MINGW64 ~/lr8 (master)
$ git tag -a teg7 -m "Друга версія"

zlata@Zlata MINGW64 ~/lr8 (master)
$ git tag
teg7
v1.0

zlata@Zlata MINGW64 ~/lr8 (master)
$
```

Пункт 7.

Перемкнувся тега на попередню раніше створений коміт.

```
MINGW64/c/Users/zlata/lr8

zlata@Zlata MINGW64 ~/lr8 (master)
$ 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 2bd81df Initial commit

zlata@Zlata MINGW64 ~/lr8 ((v1.0))
$ |
```

					Лабораторна робота № 8	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

Пункт 8.

Повернулася на останню версію репозиторію.

```
MINGW64/c/Users/zlata/lr8

git switch -
Turn off this advice by setting config variable advice.detachedHead to false
HEAD is now at 2bd81df Initial commit

zlata@Zlata MINGW64 ~/lr8 ((v1.0))
$ git checkout master
Previous HEAD position was 2bd81df Initial commit
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

zlata@Zlata MINGW64 ~/lr8 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

zlata@Zlata MINGW64 ~/lr8 (master)
$ |
```

Пункт 9-11.

Вивантажено зафіксовані зміни та створені теги на Github.

```
MINGW64/c/Users/zlata/lr8

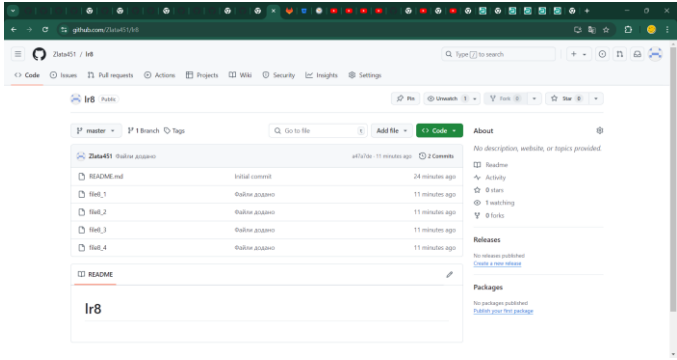
nothing to commit, working tree clean

zlata@Zlata MINGW64 ~/lr8 (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

zlata@Zlata MINGW64 ~/lr8 (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 311 bytes | 311.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Zlata451/lr8
  2bd81df..a47a7de master -> master

zlata@Zlata MINGW64 ~/lr8 (master)
$
```



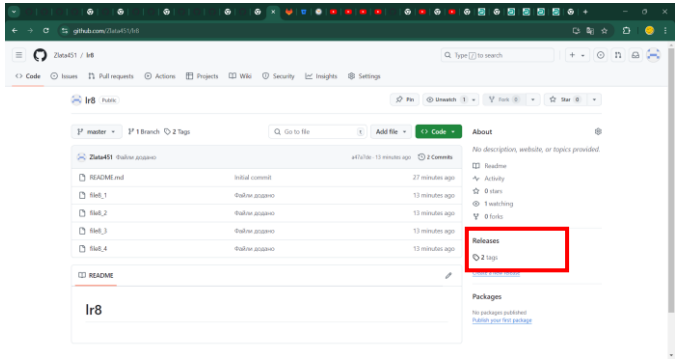
```
MINGW64/c/Users/zlata/lr8

zlata@Zlata MINGW64 ~/lr8 (master)
$ git tag
tag7
v1.0

zlata@Zlata MINGW64 ~/lr8 (master)
$ git push origin master
Everything up-to-date

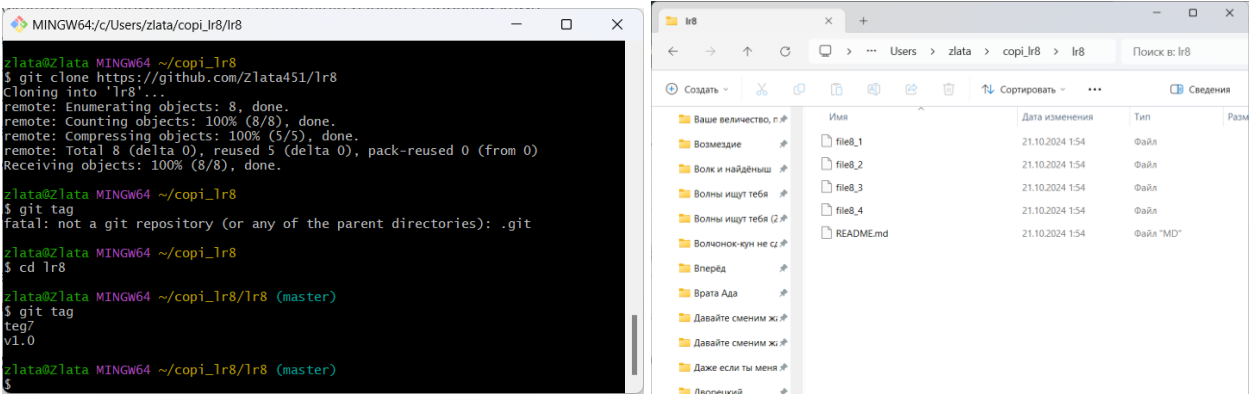
zlata@Zlata MINGW64 ~/lr8 (master)
$ git push origin --tags
Enumerating objects: 2, done.
Counting objects: 100% (2/2), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 336 bytes | 336.00 KiB/s, done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Zlata451/lr8
 * [new tag]    tag7 -> tag7
 * [new tag]    v1.0 -> v1.0

zlata@Zlata MINGW64 ~/lr8 (master)
$
```



Пункт 12.

Клоновано репозиторій в папку з ім'ям copi_lr8, де перевірено наявність тегів. Всі створені теги є ц новій папці.



Контрольні питання

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

Для перегляду списку наявних тегів в Git використовується команда **git tag**. Ця команда виведе список усіх тегів, що існують у поточному репозиторії.

2. Які два основних типи тегів використовує Git?

Git використовує два основних типи тегів:
- М'які теги (lightweight tags)
Анотаційні теги (annotated tags)

3. Розкажіть про кожен тип тегів Git.

М'які теги (lightweight tags)

Це прості, легкі теги, які є просто покажчиками на конкретний коміт. Вони не містять ніякої додаткової інформації (як-от дата створення, автор тощо) і використовуються для простого маркування комітів. М'які теги створюються командою **git tag <tag_name>**

Анотаційні теги (annotated tags)

Це більш розгорнуті теги, які містять додаткову інформацію, таку як ім'я автора, електронна пошта, дата створення та повідомлення. Анотаційні теги корисні для того, щоб дати більше контексту та інформації про певну версію. Анотаційні теги створюються командою **git tag -a <tag_name> -m "Текст повідомлення"**.

4. Чи можливо позначати вже пройдені комміти?

Так, в Git можливо позначати (тегувати) вже пройдені комміти. Для цього

					Лабораторна робота № 8	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		

необхідно вказати хеш комміту (можна використовувати частину хешу) разом з назвою тегу. Наприклад:

```
git tag <tag_name> <commit_hash>
```

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

Для видалення м'якого тегу в локальному репозиторії використовують таку команду **git tag -d <tag_name>**. Це дозволить видалити зазначений тег з локального репозиторію.

Висновок: У ході лабораторної роботи я отримала навички додавання тегів до ревізій проекту в СКВ Git.

					<i>Лабораторна робота № 8</i>	Арк.
						5
Зм.	Арк.	№ докум.	Підпис	Дата		