

## Лабораторна робота №8

### Створення та видалення тегів

#### Хід роботи

За допомогою консольного інструменту git tag додати теги для поточної ревізії

```
matador@Mat MINGW64 /lab_my (master)
$ git tag v1.0

matador@Mat MINGW64 /lab_my (master)
$ git commit -m "add tag"
On branch master
Your branch is ahead of 'pb/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

matador@Mat MINGW64 /lab_my (master)
$ git tag
v1.0
```

Зробити зміни в локальній робочій копії, зафіксувати зміни та додати новий тег для нової ревізії

```
matador@Mat MINGW64 /lab_my (master)
$ git add .

matador@Mat MINGW64 /lab_my (master)
$ git tag v1.1

matador@Mat MINGW64 /lab_my (master)
$ git commit -m "tag1.2"
[master 8d8f4cb] tag1.2
1 file changed, 0 insertions(+), 0 deletions(-)
rename 32.txt => tag12.txt (100%)
```

Перевірити наявні теги за допомогою інструменту git tag

```
matador@Mat MINGW64 /lab_my (master)
$ git tag
v1.0
v1.1
```

Перемкнутися за допомогою тега на попередню раніше створений коміт

```
matador@Mat MINGW64 /lab_my (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 b29d5ae tag

matador@Mat MINGW64 /lab_my ((v1.0))
$ ^C
```

					Лабораторна робота № 8										
Зм	Лист	№ докум	Підпис	Дата											
Розробив		Убоженко			Створення та видалення тегів				Літ.		Лист.		Листів.		
Перевірив		Левицький													
Оцінка															
Затв															
										Група 451					

## Повернутися на останню версію репозиторію

```
matador@Mat MINGW64 /lab_my ((v1.0))
$ git checkout -
Previous HEAD position was b29d5ae tag
Switched to branch 'master'
Your branch is ahead of 'pb/master' by 2 commits.
(use "git push" to publish your local commits)

matador@Mat MINGW64 /lab_my (master)
$ |
```

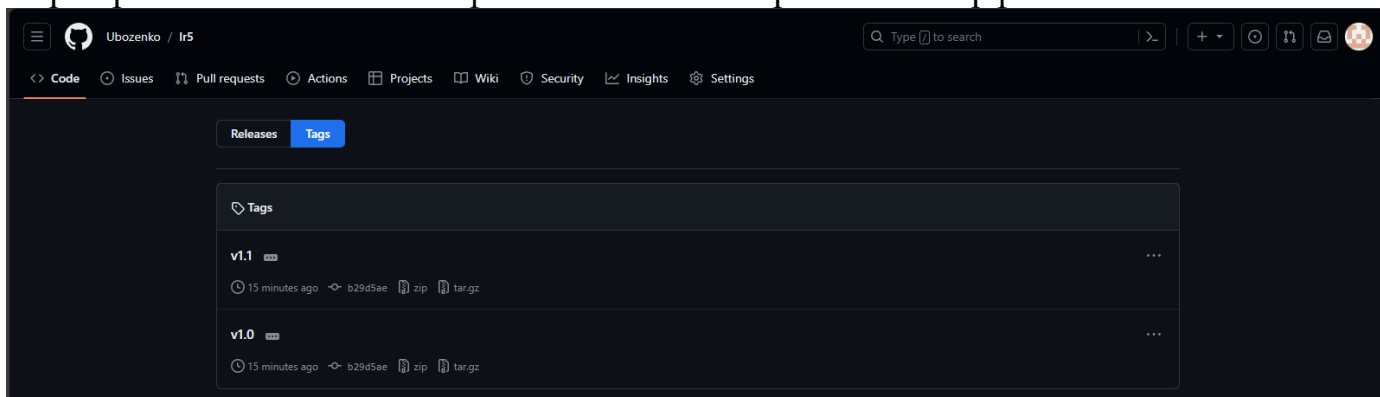
## Вивантажити зафіксовані зміни на Github

```
matador@Mat MINGW64 /lab_my (master)
$ git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 441 bytes | 441.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/Ubozenko/lr5
b3f66b7..8d8f4cb master -> master
```

## Вивантажити створені теги на Github

```
matador@Mat MINGW64 /lab_my (master)
$ git push --tags
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Ubozenko/lr5
* [new tag]          v1.0 -> v1.0
* [new tag]          v1.1 -> v1.1
```

## Перевірити наявність нових файлів на Github через web-інтерфейс



## Клонувати репозиторій в інше місце та перевірити наявність тегів

```
matador@Mat MINGW64 /lr7 (main)
$ git clone https://github.com/Ubozenko/lr5
Cloning into 'lr5'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 26 (delta 6), reused 19 (delta 3), pack-reused 0
Receiving objects: 100% (26/26), done.
Resolving deltas: 100% (6/6), done.

matador@Mat MINGW64 /lr7 (main)
$ git tag

matador@Mat MINGW64 /lr7 (main)
$ cd lr5

matador@Mat MINGW64 /lr7/lr5 (master)
$ git tag
v1.0
v1.1
```

Зм.	Арк.	№ докум.	Підпис	Дата

*Лабораторна робота № 8*

Арк.

## Контрольні питання

1. За допомогою якої команди можливо переглянути список наявних тегів в Git?  
git tag
2. Які два основних типи тегів використовує Git?  
Два основних типи тегів в Git: легковажені (lightweight) та анотовані.
3. Розкажіть про кожен тип тегів Git  
Легковажені теги (lightweight tags) просто вказують на конкретний коміт та є простою посиланням на коміт.  
Анотовані теги (annotated tags) є об'єктами Git, що зберігають багато додаткової інформації, включаючи ім'я тега, електронний підпис, дату створення та повідомлення.
4. Чи можливо позначати вже пройдені комміти?  
Так, можливо позначати вже пройдені коміти в Git.
5. Для видалення тега в локальному репозиторії необхідно виконати команду?  
Команда для видалення тега в локальному репозиторії: git tag -d ім'я\_тега.

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