

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря
Сікорського»

Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

ЛАБОРАТОРНА РОБОТА №1

з дисципліни «Технології розроблення програмного забезпечення»

Тема: «Знайомство з Git»

Виконав:

студент групи IA-33

Якименко Ярослав

Перевірив:

асистент кафедри ICT

Мягкий Михайло Юрійович

Київ - 2025

Тема: Знайомство з Git.

Мета: навчитися користуватися найбільш поширеними Git-командами.

Xід роботи

- Створюємо директорію за допомогою mkdir та переходимо до директорії за допомогою cd:

```
D:\dev>mkdir git_lab
D:\dev>cd git_lab
D:\dev\git_lab>git init
Initialized empty Git repository in D:/dev/git_lab/.git/
D:\dev\git_lab>git status
On branch main
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

- Ініціалізуємо git репозиторій:

```
D:\dev>mkdir git_lab
D:\dev>cd git_lab
D:\dev\git_lab>git init
Initialized empty Git repository in D:/dev/git_lab/.git/
D:\dev\git_lab>git status
On branch main
No commits yet
nothing to commit (create/copy files and use "git add" to track)
```

- Робимо пустий коміт для можливості створення нових гілок:

```
D:\dev\git_lab>git commit -m "git init" --allow-empty
[main (root-commit) 039b118] git init
D:\dev\git_lab>git branch b1
D:\dev\git_lab>git checkout -b b2
Switched to a new branch 'b2'
```

- Створюємо гілки b1-b3 трьома різними способами:

```
D:\dev\git_lab>git commit -m "git init" --allow-empty
[main (root-commit) 039b118] git init
D:\dev\git_lab>git branch b1
D:\dev\git_lab>git checkout -b b2
Switched to a new branch 'b2'
D:\dev\git_lab>git checkout master
error: pathspec 'master' did not match any file(s) known to git
D:\dev\git_lab>git checkout main
Switched to branch 'main'
D:\dev\git_lab>git switch -c b3
Switched to a new branch 'b3'
```

- Створюємо три файли f1-f3 та додаємо їх:

```
D:\dev\git_lab>echo "f1" > f1.txt
D:\dev\git_lab>echo "f2" > f2.txt
D:\dev\git_lab>echo "f3" > f3.txt
```

```
D:\dev\git_lab>git status
On branch main
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1.txt
    f2.txt
    f3.txt

nothing added to commit but untracked files present (use "git add" to track)

D:\dev\git_lab>git add .

D:\dev\git_lab>git status
On branch main
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1.txt
    new file:   f2.txt
    new file:   f3.txt
```

- Перемикаємось на b1 та робимо коміт з файлом f1.txt:

```
D:\dev\git_lab>git checkout b1
A      f1.txt
A      f2.txt
A      f3.txt
Switched to branch 'b1'

D:\dev\git_lab>git commit -m "f1.txt" f1.txt
[b1 5c55c35] f1.txt
  1 file changed, 1 insertion(+)
  create mode 100644 f1.txt

D:\dev\git_lab>git checkout main
A      f2.txt
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b2
A      f2.txt
A      f3.txt
Switched to branch 'b2'

D:\dev\git_lab>git commit -m "f2.txt" f2.txt
[b2 53f4b6f] f2.txt
  1 file changed, 1 insertion(+)
  create mode 100644 f2.txt

D:\dev\git_lab>git checkout main
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b3
A      f3.txt
Switched to branch 'b3'

D:\dev\git_lab>git commit -m "f3.txt" f3.txt
[b3 689b5a9] f3.txt
  1 file changed, 1 insertion(+)
```

- Перемикаємось на b2 та робимо коміт з файлом f2.txt:

```

D:\dev\git_lab>git checkout b1
A      f1.txt
A      f2.txt
A      f3.txt
Switched to branch 'b1'

D:\dev\git_lab>git commit -m "f1.txt" f1.txt
[bl 5c55c35] f1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

D:\dev\git_lab>git checkout main
A      f2.txt
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b2
A      f2.txt
A      f3.txt
Switched to branch 'b2'

D:\dev\git_lab>git commit -m "f2.txt" f2.txt
[b2 53f4b6f] f2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 f2.txt

D:\dev\git_lab>git checkout main
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b3
A      f3.txt
Switched to branch 'b3'

D:\dev\git_lab>git commit -m "f3.txt" f3.txt
[b3 689b5a9] f3.txt
 1 file changed, 1 insertion(+)

```

- Перемикаємось на b3 та робимо коміт з файлом f3.txt:

```

D:\dev\git_lab>git checkout b1
A      f1.txt
A      f2.txt
A      f3.txt
Switched to branch 'b1'

D:\dev\git_lab>git commit -m "f1.txt" f1.txt
[bl 5c55c35] f1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

D:\dev\git_lab>git checkout main
A      f2.txt
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b2
A      f2.txt
A      f3.txt
Switched to branch 'b2'

D:\dev\git_lab>git commit -m "f2.txt" f2.txt
[b2 53f4b6f] f2.txt
 1 file changed, 1 insertion(+)
 create mode 100644 f2.txt

D:\dev\git_lab>git checkout main
A      f3.txt
Switched to branch 'main'

D:\dev\git_lab>git checkout b3
A      f3.txt
Switched to branch 'b3'

D:\dev\git_lab>git commit -m "f3.txt" f3.txt
[b3 689b5a9] f3.txt
 1 file changed, 1 insertion(+)

```

- Створюємо, додаємо та комітимо файл f1.txt на гілці b3:

```

D:\dev\git_lab>echo "hello from b3" > f1.txt

D:\dev\git_lab>git status
On branch b3
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    f1.txt

nothing added to commit but untracked files present (use "git add" to track)

D:\dev\git_lab>git add f1.txt

D:\dev\git_lab>git status
On branch b3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   f1.txt

D:\dev\git_lab>git commit -m "f1.txt" f1.txt
[b3 3978a17] f1.txt
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

D:\dev\git_lab>

```

- Мерджимо b1 в гілку b2, отримуємо конфлікт, так як у b1 та b2 є файли f2.txt з різним наповненням:

```
D:\dev\git_lab>git merge b1
Auto-merging f1.txt
CONFLICT (add/add): Merge conflict in f1.txt
Automatic merge failed; fix conflicts and then commit the result.

D:\dev\git_lab>nano f1.txt
D:\dev\git_lab>
```

- Редагуємо файл f1.txt, тим самим вирішуючи конфлікт, додаємо цей відредагований файл та продовжуємо merge

The image consists of three parts. The top part shows two terminal windows. The left terminal is running on D:\dev\git_lab and displays the command 'git merge b1' followed by a merge conflict message for 'f1.txt'. The right terminal is also on D:\dev\git_lab and shows the command 'nano f1.txt'. The bottom part shows a screenshot of a terminal window displaying a git log --graph output. The graph shows several commits from different authors (e.g., 'AMERDOP', 'ANARDO', 'ANARDO', 'ANARDO') and branches ('HEAD', 'b1', 'b2', 'b3'). A green line indicates the 'result merge' commit where multiple branches were combined.

- Через git log --graph дивимось, які коміти та які гілки було злито:

```
D:\dev\git_lab>git add f1.txt

D:\dev\git_lab>git commit -m "result merge"
[b3 896f650] result merge

D:\dev\git_lab>git log --graph
* commit 896f65039313789f7c76071ee31d7ed55e711ab4 (HEAD -> b3)
| \
|   Merge: 3978a17 5c55c35
|   Author: Андрій Грицай <hastarotimmortal@gmail.com>
|   Date:   Sun Sep 7 01:01:14 2025 +0300
|
|       result merge
|
* commit 5c55c351a7746461ce3aae38360bfc67d4f2773f (b1)
| Author: Андрій Грицай <hastarotimmortal@gmail.com>
| Date:   Sun Sep 7 00:30:28 2025 +0300
|
|   f1.txt
|
* commit 3978a17b03ca05d94bb986e5b5f821707e82836b
| Author: Андрій Грицай <hastarotimmortal@gmail.com>
| Date:   Sun Sep 7 00:54:36 2025 +0300
|
|   f1.txt
|
* commit 689b5a913ef155e2410f2b828d8cf7bffd906641
| Author: Андрій Грицай <hastarotimmortal@gmail.com>
| Date:   Sun Sep 7 00:33:08 2025 +0300
|
|   f3.txt
|
* commit 039b1180316ce6c31f41809ea3e72307fad05f85 (main)
| Author: Андрій Грицай <hastarotimmortal@gmail.com>
| Date:   Sat Sep 6 23:57:46 2025 +0300
|
|   git init
D:\dev\git_lab>
```

Висновок: під час виконання лабораторної роботи я навчився основам роботи з системою контролю версій git, навчився створювати гілки, робити коміти, зливати гілки в одну та вирішувати конфлікти при злитті.