Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційні систем та технологій

Лабораторна робота № 1

із дисципліни «Технології розроблення програмного забезпечення»

Тема: «Розподілена система контролю версій «Git»

Виконав
Студент групи ІА-31:
Олея М. С.

Перевірив:
Мягкий М. Ю.

Київ 2025

1. **Мета:** Навчитися виконувати основні операції в роботі з децентралізованими системами контролю версій на прикладі роботи з сучасною системою Git.

2. Теоретичні відомості:

Git та його призначення:

Git — це розподілена система контролю версій, яка дозволяє відстежувати зміни у файлах, координувати роботу кількох розробників і зберігати історію проекту.

Гілки:

Гілка в Git — це окрема лінія розвитку проекту. Вона дозволяє розробляти нові функції або виправляти помилки, не впливаючи на основну версію проекту master

Коміти:

Коміт — це збережений стан файлів у репозиторії**.** Він містить інформацію про зміни і коментар.

Merge:

Merge — це процес злиття однієї гілки з іншою**,** що дозволяє об’єднати зміни декількох гілок.

Конфлікти та їх вирішення:

Конфлікти виникають, коли зміни в одних і тих же рядках файлу зійшлися в різних гілках**..**

Щоб вирішити конфлікт: редагуємо файл, видаляємо позначки, додаємо файл (git add) і завершуємо merge або rebase.

3. Хід роботи:

В терміналі

- створити порожній репозиторій

```
C:\Users\User>mkdir laba1

C:\Users\User>cd laba1
```

```
C:\Users\User\laba1>git init
Initialized empty Git repository in C:/Users/User/laba1/.git/
```

- з гілки main створити 3 гілки, використавши 3 різні способи

```
C:\Users\User\laba1>git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Перевіряємо дії в нашому репозиторії

```
C:\Users\User\laba1>echo "A1" > A1.txt

C:\Users\User\laba1>git add A1.txt

C:\Users\User\laba1>git commit -m "addingA1"
[main (root-commit) 6ac82ea] addingA1
 1 file changed, 1 insertion(+)
 create mode 100644 A1.txt
```

Робимо перший commit

```
C:\Users\User\laba1>git branch branch1

C:\Users\User\laba1>git branch
  branch1
* main

C:\Users\User\laba1>git checkout -b branch2
Switched to a new branch 'branch2'

C:\Users\User\laba1>git branch
  branch1
* branch2
  main

C:\Users\User\laba1>git checkout main
Switched to branch 'main'

C:\Users\User\laba1>git switch -c branch3
Switched to a new branch 'branch3'

C:\Users\User\laba1>git branch
  branch1
  branch2
* branch3
  main
```

Створюємо 3 гілки через «branch» «checkout» «switch»

- створити 3 файли та відрядити їх по 3 гілках

```
C:\Users\User\laba1>echo "1" > F1.txt

C:\Users\User\laba1>echo "2" > F2.txt

C:\Users\User\laba1>echo "3" > F3.txt

C:\Users\User\laba1>git add F3.txt

C:\Users\User\laba1>git commit -m "adding F3"
[branch3 1818c92] adding F3
 1 file changed, 1 insertion(+)
 create mode 100644 F3.txt

C:\Users\User\laba1>git checkout branch2
Switched to branch 'branch2'

C:\Users\User\laba1>git add F2.txt

C:\Users\User\laba1>git commit -m "addingF2"
[branch2 388406e] addingF2
 1 file changed, 1 insertion(+)
 create mode 100644 F2.txt

C:\Users\User\laba1>git checkout branch1
Switched to branch 'branch1'

C:\Users\User\laba1>git add F1.txt

C:\Users\User\laba1>git commit -m "addingF1"
[branch1 97aed31] addingF1
 1 file changed, 1 insertion(+)
 create mode 100644 F1.txt
```

Створюємо файли F (1, 2, 3) та коммітиммо їх

```
C:\Users\User\laba1>git log --all
commit 97aed3175656f1053c58a3d32188222cde3b9cd2 (HEAD -> branch1)
Author: Hoityk <oleya.michail@gmail.com>
Date:   Sat Nov 1 10:01:13 2025 +0200

    addingF1

commit 388406efb172b44f36c5744b242dbfe3fd360982 (branch2)
Author: Hoityk <oleya.michail@gmail.com>
Date:   Sat Nov 1 10:00:22 2025 +0200

    addingF2

commit 1818c9223c055e7e6384520ac10b8a8b9c9c1b26 (branch3)
Author: Hoityk <oleya.michail@gmail.com>
Date:   Sat Nov 1 09:59:00 2025 +0200

    adding F3

commit 6ac82ead7903f5f0ec206ea590a34196d68f7cdb (main)
Author: Hoityk <oleya.michail@gmail.com>
Date:   Sat Nov 1 09:53:58 2025 +0200

    addingA1
```

```
C:\Users\User\laba1>git log --all --graph
* commit 97aed3175656f1053c58a3d32188222cde3b9cd2 (HEAD -> branch1)
| Author: Hoityk <oleya.michail@gmail.com>
| Date:   Sat Nov 1 10:01:13 2025 +0200
|
|     addingF1
|
| * commit 388406efb172b44f36c5744b242dbfe3fd360982 (branch2)
|/  Author: Hoityk <oleya.michail@gmail.com>
|   Date:   Sat Nov 1 10:00:22 2025 +0200
|
|       addingF2
|
| * commit 1818c9223c055e7e6384520ac10b8a8b9c9c1b26 (branch3)
|/  Author: Hoityk <oleya.michail@gmail.com>
|   Date:   Sat Nov 1 09:59:00 2025 +0200
|
|       adding F3
|
* commit 6ac82ead7903f5f0ec206ea590a34196d68f7cdb (main)
  Author: Hoityk <oleya.michail@gmail.com>
  Date:   Sat Nov 1 09:53:58 2025 +0200

      addingA1
```

Переглядаємо наші комміти

- об’єднати другу та третю гілки, використовуючи “rebase”

```
C:\Users\User\laba1>git branch
* branch1
  branch2
  branch3
  main

C:\Users\User\laba1>git checkout branch2
Switched to branch 'branch2'

C:\Users\User\laba1>git rebase branch3
Successfully rebased and updated refs/heads/branch2.

C:\Users\User\laba1>git log --all --graph
* commit 8cf1309262ee01859fe2b9478f6514950861a7c6 (HEAD -> branch2)
| Author: Hoityk <oleya.michail@gmail.com>
| Date:   Sat Nov 1 10:00:22 2025 +0200
|
|     addingF2
|
* commit 1818c9223c055e7e6384520ac10b8a8b9c9c1b26 (branch3)
| Author: Hoityk <oleya.michail@gmail.com>
| Date:   Sat Nov 1 09:59:00 2025 +0200
|
|     adding F3
|
| * commit 97aed3175656f1053c58a3d32188222cde3b9cd2 (branch1)
|/  Author: Hoityk <oleya.michail@gmail.com>
|   Date:   Sat Nov 1 10:01:13 2025 +0200
|
|       addingF1
|
* commit 6ac82ead7903f5f0ec206ea590a34196d68f7cdb (main)
  Author: Hoityk <oleya.michail@gmail.com>
  Date:   Sat Nov 1 09:53:58 2025 +0200

      addingA1
```

4. Висновок

Лабораторна робота поглибила мої знання Git: від створення репозиторіїв і роботи з комітами до управління гілками та злиттям змін. Практика показала, як контроль версій оптимізує командну роботу над проєктами.