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

Лабораторна робота №1

Технології розроблення програмного забезпечення

Тема: «Основи роботи з Git»

Виконала:
Студентка групи ІА-34
Дригант А.С
Перевірив:
Мягкий Михайло Юрійович

Київ-2025

Тема: Основи роботи з Git

Мета: Ознайомитися з базовими можливостями системи контролю версій Git. Навчитися створювати новий репозиторій, ініціалізувати коміт, створювати та перемикатися між гілками, додавати й фіксувати зміни у файлах, а також виконувати злиття гілок із розв'язанням конфліктів. У процесі роботи закріпити практичні навички роботи з різними командами

Теоретичні відомості

Git — це розподілена система контролю версій, яка дозволяє відслідковувати зміни у файлах, працювати над проектами в команді та зберігати історію розробки. Вона використовується для зручного управління програмним кодом і дозволяє розробникам повертатися до попередніх версій, створювати паралельні гілки розвитку та об'єднувати результати роботи.

Основні поняття Git:

- Репозиторій (Repository) - сховище, в якому зберігається весь проект та історія його змін. Репозиторій може бути локальним (на комп'ютері користувача) та віддаленим (на сервері, наприклад GitHub).
- Коміт (Commit) - збереження поточного стану файлів у репозиторії. Кожен коміт має унікальний ідентифікатор (хеш) і опис змін.
- Гілка (Branch) - незалежна лінія розробки. За замовчуванням створюється гілка master (або main). Нові гілки дозволяють працювати над різними функціональними можливостями окремо.
- Merge (злиття) - об'єднання змін з однієї гілки в іншу. Якщо зміни конфліктують (змінюють один і той самий рядок у файлі), виникає конфлікт злиття, який потрібно вирішити вручну.
- Індекс (Staging area) - проміжна область, куди додаються файли командою git add перед комітом.
- HEAD - покажчик на поточний коміт або активну гілку.

Основні команди Git:

- git init - створення нового локального репозиторію.
- git status - перевірка стану файлів у репозиторії (які змінені, які додані до індексу).
- git add <file> - додавання файлів у staging area.
- git commit -m "повідомлення" - створення коміту з повідомленням.
- git branch - перегляд або створення нових гілок.
- git checkout <branch> / git switch <branch> - перемикання між гілками.
- git merge <branch> - об’єднання вказаної гілки з поточною.
- git log --graph - перегляд історії комітів у вигляді графа.

Хід роботи

1. **Створення папки та ініціалізація репозиторію**

- У командному рядку була створена нова папка new_folder.
- Виконано перехід у цю папку та ініціалізацію Git-репозиторію командою git init.
- Перевірено поточну гілку та стан репозиторію командами git branch та git status.

2. **Створення початкового коміту та гілок**

- Створено перший пустий коміт за допомогою git commit --allow-empty -m "init".
- Створено нові гілки:
    - b1 (команда git branch b1),
    - b2 (команда git checkout -b b2),
    - b3 (команда git switch -c b3).

3. **Створення файлів та додавання їх у репозиторій**

- У гілці b3 створено три текстові файли: f1.txt, f2.txt, f3.txt.
- Виконано додавання файлів у staging area командою git add ..

- Створено коміт з додаванням файлу f3.txt та зафіксовано зміни.

4. **Робота з гілкою b1**

- Перехід у гілку b1.
- Зафіксовано зміни з файлом f1.txt (створено коміт).

5. **Робота з гілкою b2**

- Перехід у гілку b2.
- Створено та закомічено файл f2.txt.
- Додатково змінено файл f1.txt (додано рядок "a"), після чого зміни було додано в індекс (git add .) і виконано коміт.

6. **Перегляд історії комітів**

- Виконано команду git log --graph, щоб переглянути всі коміти у вигляді графа та визначити, які зміни знаходяться у різних гілках.

7. **Злиття гілок та вирішення конфлікту**

- Виконано злиття гілки b1 у гілку b2 (git merge b1).
- Під час злиття виник конфлікт у файлі f1.txt, оскільки він був змінений у двох гілках.
- Конфлікт вирішено вручну (позначено файл як готовий за допомогою git add f1.txt).
- Завершено процес злиття командою git merge --continue.

8. **Перевірка результатів злиття**

- Виконано git log --graph, де видно всі коміти з гілок b1 і b2, а також коміт з повідомленням **“Merge branch 'b1' into b2”**, що підтверджує успішне злиття.

```
Microsoft Windows [Version 10.0.19045.6216]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.

C:\Users\ndryg>mkdir new_folder

C:\Users\ndryg>cd new_folder

C:\Users\ndryg\new_folder>git init
Initialized empty Git repository in C:/Users/ndryg/new_folder/.git/

C:\Users\ndryg\new_folder>git branch

C:\Users\ndryg\new_folder>git status
On branch master

No commits yet

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

C:\Users\ndryg\new_folder>git branch b1
fatal: not a valid object name: 'master'

C:\Users\ndryg\new_folder>git commit --allow-empty -m "init"
[master (root-commit) f72bf9b] init

C:\Users\ndryg\new_folder>git branch b1

C:\Users\ndryg\new_folder>git checkout master
Already on 'master'

C:\Users\ndryg\new_folder>git checkout -b b2
Switched to a new branch 'b2'

C:\Users\ndryg\new_folder>git checkout

C:\Users\ndryg\new_folder>git checkout master
Switched to branch 'master'

C:\Users\ndryg\new_folder>git switch -c b3
Switched to a new branch 'b3'

C:\Users\ndryg\new_folder>echo "1" > f1.txt

C:\Users\ndryg\new_folder>echo "2" > f2.txt

C:\Users\ndryg\new_folder>echo "3" > f3.txt

C:\Users\ndryg\new_folder>git status
On branch b3
Untracked files:
```

```
C:\Users\ndryg\new_folder>git status
On branch b3
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)

C:\Users\ndryg\new_folder>git add .

C:\Users\ndryg\new_folder>git status
On branch b3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   f1.txt
        new file:   f2.txt
        new file:   f3.txt


C:\Users\ndryg\new_folder>git commit -m "add file f3" f3.txt
[b3 c1d2432] add file f3
 1 file changed, 1 insertion(+)
 create mode 100644 f3.txt

C:\Users\ndryg\new_folder>git status
On branch b3
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   f1.txt
        new file:   f2.txt


C:\Users\ndryg\new_folder>git checkout master
A       f1.txt
A       f2.txt
Switched to branch 'master'

C:\Users\ndryg\new_folder>git co b1
git: 'co' is not a git command. See 'git --help'.

The most similar commands are
        commit
        clone
        log

C:\Users\ndryg\new_folder>git chcekout b1
git: 'chcekout' is not a git command. See 'git --help'.
```

```
C:\Users\ndryg\new_folder>git chcekout b1
git: 'chcekout' is not a git command. See 'git --help'.

The most similar command is
        checkout

C:\Users\ndryg\new_folder>git checkout b1
A       f1.txt
A       f2.txt
Switched to branch 'b1'

C:\Users\ndryg\new_folder>git commit -m "f1" f1.txt
[b1 7ac0e20] f1
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt

C:\Users\ndryg\new_folder>git checkout master
A       f2.txt
Switched to branch 'master'

C:\Users\ndryg\new_folder>git checkout b2
A       f2.txt
Switched to branch 'b2'

C:\Users\ndryg\new_folder>git commit -m "f2" f2.txt
[b2 5e8ff82] f2
 1 file changed, 1 insertion(+)
 create mode 100644 f2.txt

C:\Users\ndryg\new_folder>echo "a" > f1.txt

C:\Users\ndryg\new_folder>git status
On branch b2
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)

C:\Users\ndryg\new_folder>git add .

C:\Users\ndryg\new_folder>git commit
[b2 bfe7c0c] added f1 to b2
 1 file changed, 1 insertion(+)
 create mode 100644 f1.txt
```

```
C:\Users\ndryg\new_folder>git log --graph
* commit bfe7c0cc35c19206ad312c6e425c48ac79e47113 (HEAD -> b2)
| Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| Date:   Sat Sep 6 14:51:54 2025 +0300
|
|     added f1 to b2
|
* commit 5e8ff824fa39a09b241d2aee4604a85421fa4abc
| Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| Date:   Sat Sep 6 14:49:35 2025 +0300
|
|     f2
|
* commit f72bf9b34aea0c25650e8bdf4e9eb0733de9968b (master)
  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
  Date:   Sat Sep 6 14:33:50 2025 +0300

      init

C:\Users\ndryg\new_folder>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.

C:\Users\ndryg\new_folder>git status
On branch b2
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both added:      f1.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\ndryg\new_folder>nano f1.txt
"nano" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\ndryg\new_folder>vi f1.txt
"vi" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

C:\Users\ndryg\new_folder>git add .

C:\Users\ndryg\new_folder>git merge --continue
[b2 bf2bd86] Merge branch 'b1' into b2
```

```
C:\Users\ndryg\new_folder>git merge --continue
[b2 bf2bd86] Merge branch 'b1' into b2

C:\Users\ndryg\new_folder>git log --graph
*   commit bf2bd86c9656e0ad0043309f7f9a0e6f0c83369d (HEAD -> b2)
|\  Merge: bfe7c0c 7ac0e20
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:58:06 2025 +0300
| |
| |     Merge branch 'b1' into b2
| |
| * commit 7ac0e20d6f50e9d5903c10dd1af06c77b6fb3fd4 (b1)
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:48:24 2025 +0300
| |
| |     f1
| |
* | commit bfe7c0cc35c19206ad312c6e425c48ac79e47113
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:51:54 2025 +0300
| |
| |     added f1 to b2
| |
* | commit 5e8ff824fa39a09b241d2aee4604a85421fa4abc
|/  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
|   Date:   Sat Sep 6 14:49:35 2025 +0300
|
|       f2
|
* commit f72bf9b34aea0c25650e8bdf4e9eb0733de9968b (master)
  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
  Date:   Sat Sep 6 14:33:50 2025 +0300
```

```
:
*   commit bf2bd86c9656e0ad0043309f7f9a0e6f0c83369d (HEAD -> b2)
|\  Merge: bfe7c0c 7ac0e20
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:58:06 2025 +0300
| |
| |     Merge branch 'b1' into b2
| |
| * commit 7ac0e20d6f50e9d5903c10dd1af06c77b6fb3fd4 (b1)
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:48:24 2025 +0300
| |
| |     f1
| |
* | commit bfe7c0cc35c19206ad312c6e425c48ac79e47113
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:51:54 2025 +0300
| |
| |     added f1 to b2
| |
* | commit 5e8ff824fa39a09b241d2aee4604a85421fa4abc
|/  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
|   Date:   Sat Sep 6 14:49:35 2025 +0300
|
|       f2
|
* commit f72bf9b34aea0c25650e8bdf4e9eb0733de9968b (master)
  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
  Date:   Sat Sep 6 14:33:50 2025 +0300
```

```
:
*   commit bf2bd86c9656e0ad0043309f7f9a0e6f0c83369d (HEAD -> b2)
|\  Merge: bfe7c0c 7ac0e20
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:58:06 2025 +0300
| |
| |     Merge branch 'b1' into b2
| |
| * commit 7ac0e20d6f50e9d5903c10dd1af06c77b6fb3fd4 (b1)
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:48:24 2025 +0300
| |
| |     f1
| |
* | commit bfe7c0cc35c19206ad312c6e425c48ac79e47113
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:51:54 2025 +0300
| |
| |     added f1 to b2
| |
* | commit 5e8ff824fa39a09b241d2aee4604a85421fa4abc
|/  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
|   Date:   Sat Sep 6 14:49:35 2025 +0300
|
|       f2
|
* commit f72bf9b34aea0c25650e8bdf4e9eb0733de9968b (master)
  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
  Date:   Sat Sep 6 14:33:50 2025 +0300
```

```
:...skipping...
*   commit bf2bd86c9656e0ad0043309f7f9a0e6f0c83369d (HEAD -> b2)
|\  Merge: bfe7c0c 7ac0e20
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:58:06 2025 +0300
| |
| |     Merge branch 'b1' into b2
| |
| * commit 7ac0e20d6f50e9d5903c10dd1af06c77b6fb3fd4 (b1)
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:48:24 2025 +0300
| |
| |     f1
| |
* | commit bfe7c0cc35c19206ad312c6e425c48ac79e47113
| | Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
| | Date:   Sat Sep 6 14:51:54 2025 +0300
| |
| |     added f1 to b2
| |
* | commit 5e8ff824fa39a09b241d2aee4604a85421fa4abc
|/  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
|   Date:   Sat Sep 6 14:49:35 2025 +0300
|
|       f2
|
* commit f72bf9b34aea0c25650e8bdf4e9eb0733de9968b (master)
  Author: Nastenaaa <145150122+Nastenaaa@users.noreply.github.com>
  Date:   Sat Sep 6 14:33:50 2025 +0300

      init
```

Висновок: у ході виконання лабораторної роботи я ознайомилася з основами роботи системи контролю версій Git. Я повторила як створювати локальний репозиторій, додавати файли та фіксувати зміни за допомогою комітів. Також я виконала створення та перемикання між гілками, внесла зміни у різних гілках та провела їх злиття. Під час злиття гілок зіткнулася з конфліктом у файлі та успішно вирішила його, що дало змогу зрозуміти, як Git працює з одночасними змінами в одному і тому ж файлі. Я закріпила практичні навички роботи з основними командами.