

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**

із дисципліни «*ТПЗ*»

Тема: Системи контролю версій. Розподілена система контролю версій  
«Git».

**Виконав:**

Студент групи ІА-34

Ковальчук Станіслав

**Перевірив:**

Мягкий М.Ю.

### Завдання:

- Ознайомитись із короткими теоретичними відомостями.
- Створити Git репозиторій.
- Клонувати Git репозиторій.
- Продемонструвати базову роботу з репозиторієм: створення версій, додавання тегів, робіт з гілками (створення та злиття), робота з комітами, вирішення конфліктів, а також робота з віддаленим репозиторієм.

### Виконання завдання:

1. Створений локальний репозиторій:

```
C:\Users\stasw> git init lab1
Initialized empty Git repository in C:/Users/stasw/lab1/.git/

C:\Users\stasw> cd lab1
```

2. Створений і закомічений в репозиторій файл.

```
C:\Users\stasw\lab1> echo "test" > hello.txt

C:\Users\stasw\lab1> git add hello.txt
```

```
C:\Users\stasw\lab1> git commit -m "Hello added"
[master (root-commit) 925a4ee] Hello added
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
```

3. Створена в проєкті директорія.

```
C:\Users\stasw\lab1> mkdir child_dir

C:\Users\stasw\lab1> cd child_dir
```

```
C:\Users\stasw\lab1\child_dir> git add child_dir
fatal: pathspec 'child_dir' did not match any files

C:\Users\stasw\lab1\child_dir> cd ..

C:\Users\stasw\lab1> git add child_dir

C:\Users\stasw\lab1> git commit -m "Add child directory with content"
[master d4caffa] Add child directory with content
 1 file changed, 1 insertion(+)
 create mode 100644 child_dir/inner.txt
```

4. Створено дві гілки (master і feature\_branch)

```
C:\Users\stasw\lab1>git branch feature_branch
```

```
C:\Users\stasw\lab1>git checkout feature_branch  
Switched to branch 'feature_branch'
```

5. У гілці feature\_branch видалено директорію і закомічено зміни

```
C:\Users\stasw\lab1>rmdir /s /q child_dir
```

```
C:\Users\stasw\lab1>dir  
Volume in drive C has no label.  
Volume Serial Number is 4419-23E7
```

```
Directory of C:\Users\stasw\lab1
```

```
23.09.2025  10:37    <DIR>          .  
23.09.2025  10:26    <DIR>          ..  
23.09.2025  10:23                9 hello.txt  
                1 File(s)                9 bytes  
                2 Dir(s)  727 848 988 672 bytes free
```

```
C:\Users\stasw\lab1>git add .
```

```
C:\Users\stasw\lab1>git commit -m "Remove child_dir"  
[feature_branch d5b5a49] Remove child_dir  
1 file changed, 1 deletion(-)  
delete mode 100644 child_dir/inner.txt
```

6. Виконано злиття змін з основною гілкою

```
C:\Users\stasw\lab1>git checkout master  
Switched to branch 'master'
```

```
C:\Users\stasw\lab1>git merge feature_branch  
Updating d4caffa..d5b5a49  
Fast-forward  
child_dir/inner.txt | 1 -  
1 file changed, 1 deletion(-)  
delete mode 100644 child_dir/inner.txt
```

7. GIT\_LOG

```
C:\Users\stasw\lab1>git log
commit d5b5a494ce34679ece474a0f4d37d0d932830a14 (HEAD -> master, feature_branch)
Author: Stas Kovalchuk <staswwwkovalchuk@gmail.com>
Date: Tue Sep 23 10:38:32 2025 +0300

    Remove child_dir

commit d4cafface184bcad61fcc1f60d9290bc214f5313
Author: Stas Kovalchuk <staswwwkovalchuk@gmail.com>
Date: Tue Sep 23 10:31:27 2025 +0300

    Add child directory with content

commit 925a4ee0d1984013cb37e61fbd2113968cf3633c
Author: Stas Kovalchuk <staswwwkovalchuk@gmail.com>
Date: Tue Sep 23 10:26:36 2025 +0300

    Hello added
```

Висновок: Під час виконання лабораторної роботи я детально розглянув принципи функціонування систем контролю версій та набув практичного досвіду роботи з Git. У процесі роботи було створено локальний репозиторій, додано та зафіксовано зміни у файлах, виконано операції з гілками, а також проведено їх злиття. Виконані дії дали змогу краще зрозуміти підхід до керування версіями та структурування змін у проєкті. Отже, поставлені цілі лабораторної роботи досягнуто - я отримав важливі навички, які стануть у пригоді під час подальшої розробки програмних продуктів та командної роботи.

### 1. Що таке система контролю версій (СКВ)?

СКВ – це програмне забезпечення, яке дозволяє зберігати історію змін у файлах, відслідковувати, хто і коли вносив правки, повертатися до попередніх версій, працювати над проєктом командою та об'єднувати результати роботи.

### 2. Поясніть відмінності між розподіленою та централізованою СКВ.

- Централізована (CVS, SVN): один центральний сервер, усі зміни зберігаються на ньому, робота без підключення до сервера майже неможлива.
- Розподілена (Git, Mercurial): у кожного розробника є повна копія репозиторію, можна працювати офлайн, зміни синхронізуються при підключенні.

### 3. Поясніть різницю між stage та commit в Git.

- Stage (індексація): підготовка файлів до фіксації (git add).
- Commit: збереження підготовлених змін у локальний репозиторій з коментарем (git commit -m "...").

#### 4. Як створити гілку в Git?

Команда:

```
git branch new_branch
```

Або одразу створити і перейти:

```
git checkout -b new_branch
```

#### 5. Як створити або скопіювати репозиторій Git з віддаленого серверу?

Команда:

```
git clone <url>
```

#### 6. Що таке конфлікт злиття, як створити конфлікт, як вирішити конфлікт?

- Конфлікт злиття: виникає, коли двоє розробників змінили один і той самий рядок у файлі в різних гілках.
- Створити конфлікт: змінити один і той самий рядок у файлі у двох різних гілках і спробувати з'єднати (git merge).
- Вирішення: вручну виправити файл (залишити потрібний варіант), позначити зміни як вирішені (git add), зробити git commit.

#### 7. В яких ситуаціях використовуються команди: merge, rebase, cherry-pick?

- merge – об'єднання двох гілок в одну (зберігає історію).
- rebase – "переписування" історії, щоб застосувати коміти на основі іншої гілки (історія лінійна).

- cherry-pick – застосування окремого конкретного коміту з іншої гілки.

8. Як переглянути історію змін Git репозиторію в консолі?

Команда:

`git log`

9. Як створити гілку в Git не використовуючи команду `git branch`?

Використати:

`git checkout -b new_branch`

або

`git switch -c new_branch`

10. Як підготувати всі зміни в поточній папці до коміту?

`git add .`

11. Як підготувати всі зміни в дочірній папці до коміту?

`git add child_dir/`

12. Як переглянути перелік наявних гілок в репозиторії?

`git branch`

13. Як видалити гілку?

- Локально:

`git branch -d branch_name`

або примусово:

`git branch -D branch_name`

- Віддалено:

`git push origin --delete branch_name`

14. Які є способи створення гілки та в чому між ними різниця?

- `git branch new_branch` – створює гілку, але не переключає на неї.
- `git checkout -b new_branch` або `git switch -c new_branch` – створює гілку і одразу переходить у неї.