



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**  
«Команди Git»

Виконала:

студентка групи ІА-23

Єрмак Д. Р

Перевірив:

Мягкий М. Ю.

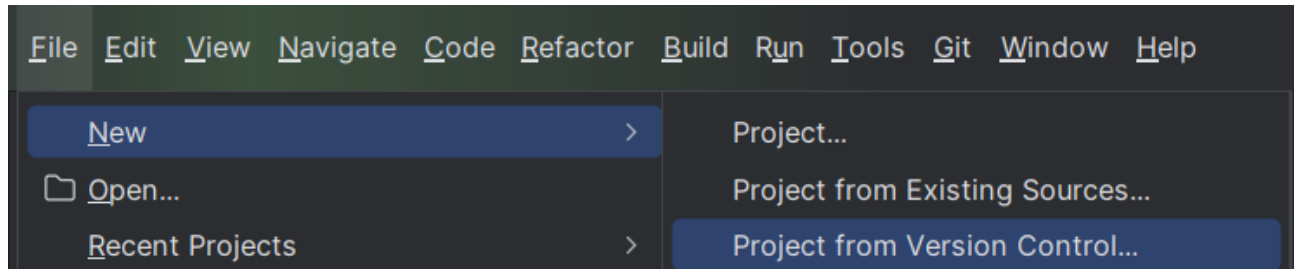
Київ 2024

## Хід роботи

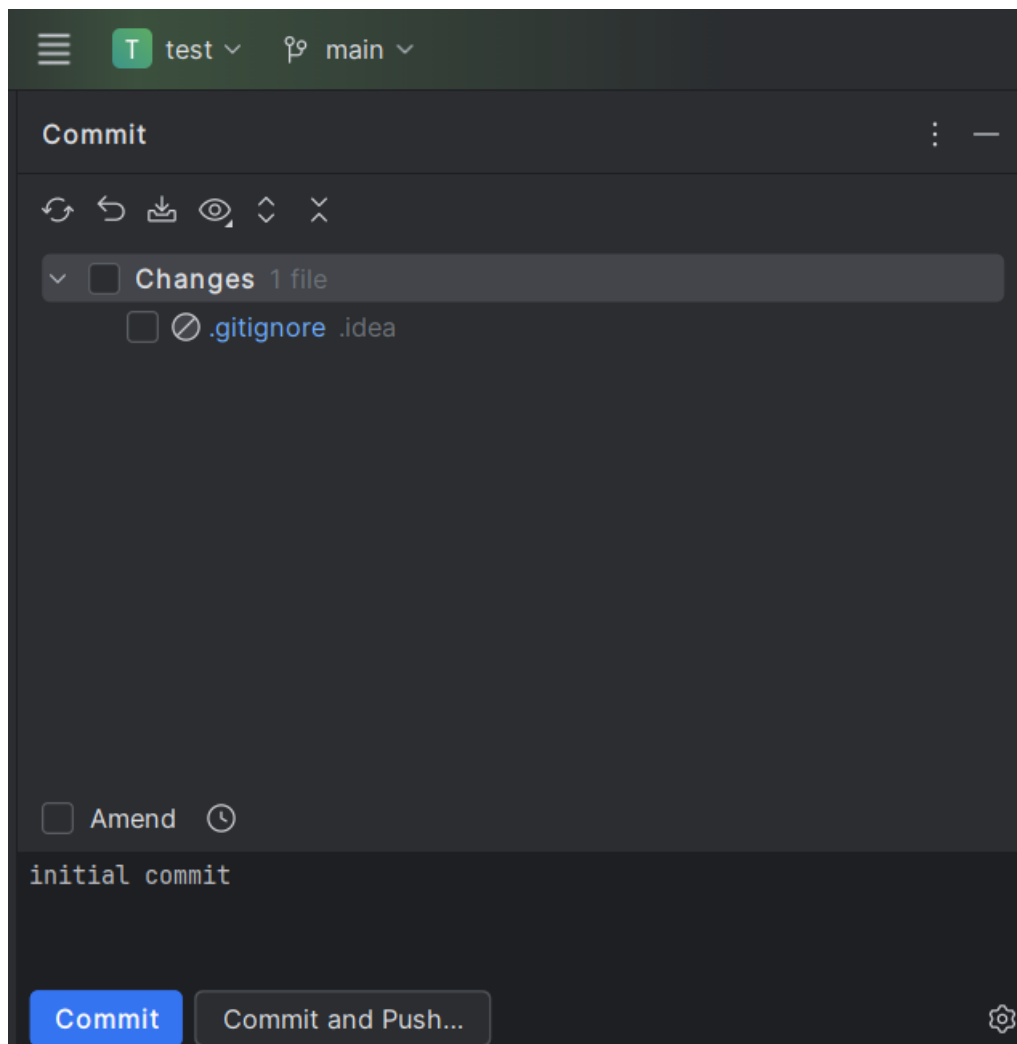
### 1. Створюємо репозиторій на Github



### 2. Клонуємо собі репозиторій



### 3. За допомогою інтерфейсу створюємо перший коміт на гілці main



### 4. Створюємо 3 гілки з основної

```
PS D:\kpi3\test> git checkout -b "branch1"
```

```
PS D:\kpi3\test> git checkout -b "branch2"
```

```
PS D:\kpi3\test> git branch "branch3"
```

5. Додаємо в гілку main файли “main.txt”, “1.txt”, “2.txt”, “3.txt” з текстовим вмістом «1»

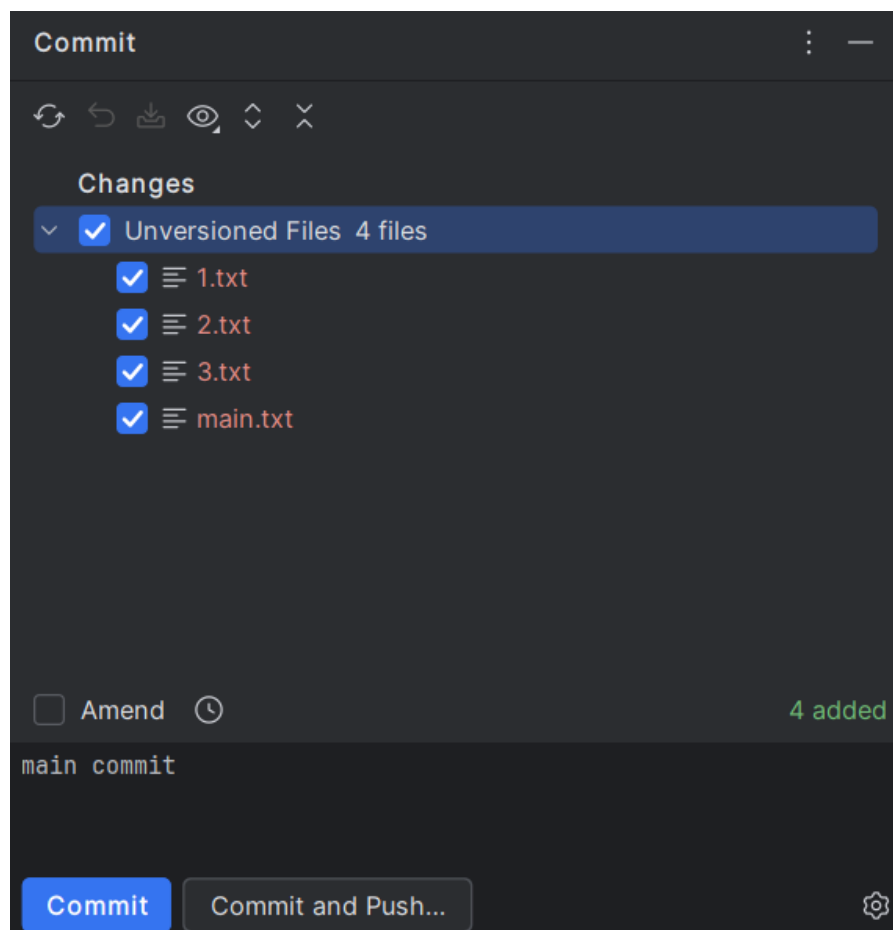
```
PS D:\kpi3\test> echo "1" > main.txt
```

```
PS D:\kpi3\test> echo "1" > 1.txt
```

```
PS D:\kpi3\test> echo "1" > 2.txt
```

```
PS D:\kpi3\test> echo "1" > 3.txt
```

6. Комітимо файли у відповідні гілки



7. Перемикаємось на гілку main та змінюємо вміст файлів

```
PS D:\kpi3\test> git checkout main
```

```
PS D:\kpi3\test> echo "a" > 1.txt
```

```
PS D:\kpi3\test> echo "a" > 2.txt
```

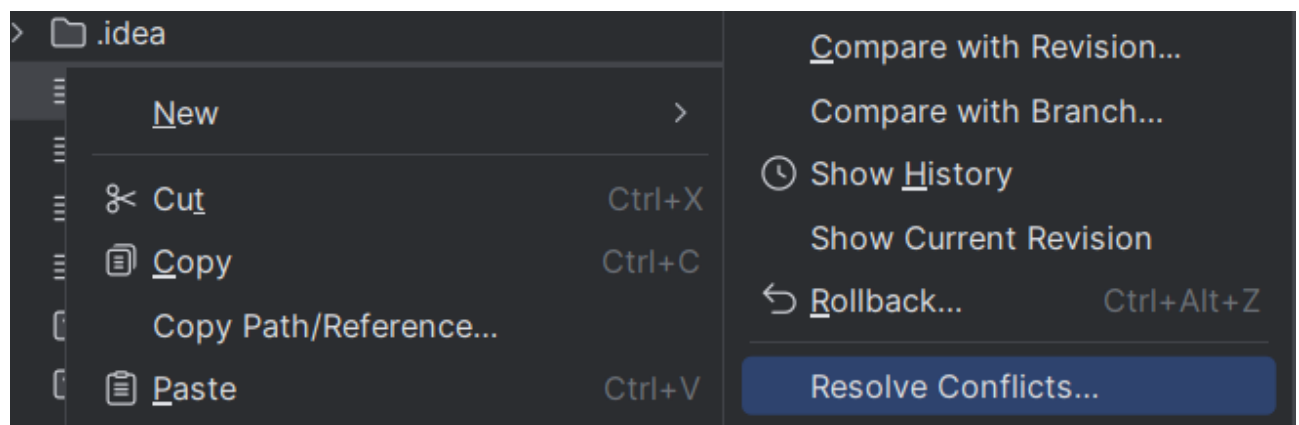
```
PS D:\kpi3\test> echo "a" > 3.txt
```

8. Перемикаємось на гілку 1 та мерджимо з основної гілки

```
PS D:\kpi3\test> git checkout branch1
```

```
PS D:\kpi3\test> git merge main
```

9. При виконанні виникає конфлікт, оскільки вміст файлу відрізняється, вирішуємо конфлікти за допомогою інтерфейсу



10. Створюємо відповідний коміт

```
PS D:\kpi3\test> git commit -m "merge main + res conf"
```

11. Перевіряємо коміти відповідної гілки

```
PS D:\kpi3\test> git log --oneline
3a99717 (HEAD -> branch1) merge main + res conf
a4919c7 (main) add a to txts
7779580 main commit
d145fd3 main commit
1f137db initial commit
```

12. Перемикаємось на 2 гілку та робимо ребейз, виникає конфлікт, вирішуємо його за прикладом попереднього варіанту

```

PS D:\kpi3\test> git rebase main
warning: Cannot merge binary files: 2.txt (HEAD vs. 67f1dfd (main commit))
Auto-merging 2.txt
CONFLICT (add/add): Merge conflict in 2.txt
error: could not apply 67f1dfd... main commit
main commit
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".

```

13. Додаємо змінений файл

```

PS D:\kpi3\test> git add .

```

14. Продовжуємо ребейз

```

PS D:\kpi3\test> git rebase --continue
[detached HEAD 135f984] main commit
Successfully rebased and updated refs/heads/branch2.
PS D:\kpi3\test> git log branch2 --oneline
135f984 (HEAD -> branch2) main commit
a4919c7 (main) add a to txts
d145fd3 main commit
1f137db initial commit

```

15. Робимо чері-пик з 2 гілки на 3

```

PS D:\kpi3\test> git checkout branch3
Switched to branch 'branch3'
PS D:\kpi3\test> git log branch3 --oneline
2528df6 (HEAD -> branch3) main commit
add a to txts
d145fd3 main commit
1f137db initial commit
PS D:\kpi3\test> git cherry-pick a4919c7 d145fd3
warning: Cannot merge binary files: 3.txt (HEAD vs. a4919c7 (add a to txts))
Auto-merging 3.txt
CONFLICT (add/add): Merge conflict in 3.txt
error: could not apply a4919c7... add a to txts
hint: After resolving the conflicts, mark them with
hint: "git add/rm <paths>", then run
hint: You can instead skip this commit with "git cherry-pick --skip".
hint: To abort and get back to the state before "git cherry-pick",
hint: run "git cherry-pick --abort".

```

16. Стикаємось з конфліктом, вирішуємо його та продовжуємо, після чого перевіряємо історію гілок

```
PS D:\kpi3\test> notepad "3.txt"
PS D:\kpi3\test> git add .
```

```
PS D:\kpi3\test> git cherry-pick --continue
[branch3 5a43048] add a to txts
Date: Thu Oct 17 13:37:23 2024 +0300
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 2.txt
[branch3 7f4c061] main commit
Date: Thu Oct 17 13:32:36 2024 +0300
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 main.txt
PS D:\kpi3\test> git log --all --oneline
7f4c061 (HEAD -> branch3) main commit
5a43048 add a to txts
135f984 (branch2) main commit
3a99717 (branch1) merge main + res conf
a4919c7 (main) add a to txts
2528df6 main commit
7779580 main commit
d145fd3 main commit
1f137db initial commit
```

```
PS D:\kpi3\test> git log --all --oneline --graph
* 7f4c061 (HEAD -> branch3) main commit
* 5a43048 add a to txts
* 2528df6 main commit
| * 135f984 (branch2) main commit
| | * 3a99717 (branch1) merge main + res conf
| | | \
| | | /
| | /|
| * | a4919c7 (main) add a to txts
| * | d145fd3 main commit
| / /
| * 7779580 main commit
| /
* 1f137db initial commit
PS D:\kpi3\test>
```

Висновок: у ході виконання даної лабораторної я спробувала на практиці основні команди для локального контролю версій Git