МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Інститут **КНІТ**
Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 1
З дисципліни: *“Моделювання та аналіз програмного забезпечення”*
На тему: *“Налаштування Git”*

Лектор:
доц. каф. ПЗ
Сердюк П.В.

Виконав:
ст. гр. ПЗ-22
Чаус Олег

Прийняв:
викл. каф. ПЗ
Микуляк А. В.

« ____ » _____ 2023 р.

∑= _____ .....

Львів – 2023

Тема роботи: Налаштування Git.

Мета роботи: Навчитися працювати з системою керування репозиторіями програмного коду GitLab.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Git є розподіленою системою контролю версій, що дозволяє відстежувати зміни у файлах та коді програмного забезпечення. Вона зберігає історію змін, що дозволяє повернутися до попередніх версій файлів або повернутися до конкретного стану проекту.

У Git знімки станів системи в репозиторії зберігаються у вигляді комітів, що можуть належати до різних гілок. Гілки дозволяють відокремити розробку різних функціональних частин проекту та об'єднувати їх пізніше.

Git має низку команд, які можна використовувати для роботи з ним. Для початку роботи з Git потрібно створити репозиторій, додати файли до стеження та зробити перший коміт. Далі можна створювати гілки, переходити між гілками та об'єднувати їх.

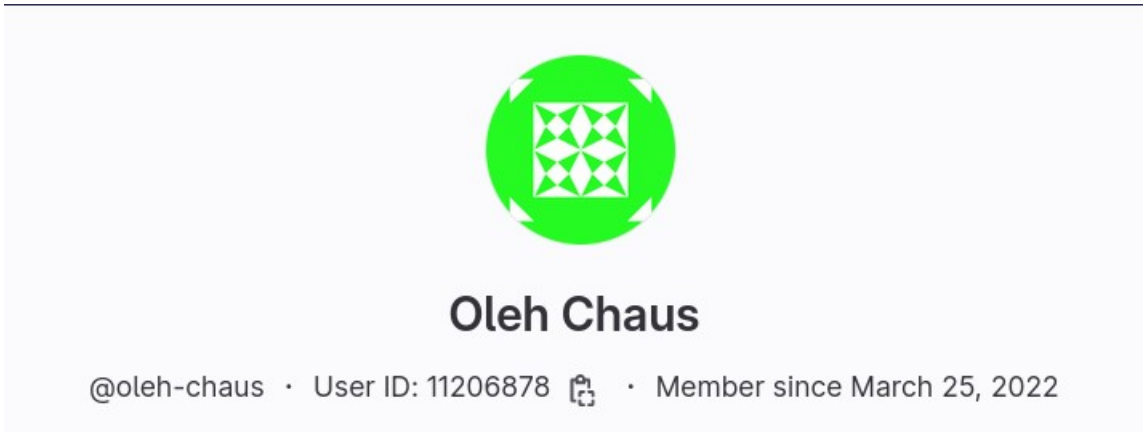
Git дозволяє працювати з репозиторієм як локально, так і в мережі, що дозволяє працювати в команді та обмінюватися змінами між учасниками проекту.

ЗАВДАННЯ

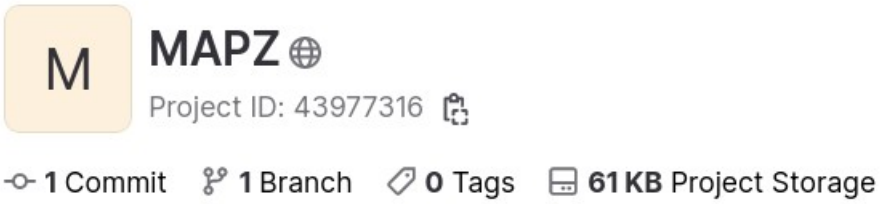
1. Створити акаунт на GitLab.
2. Створити репозиторій з назвою MAPZ.
3. Створити папку для лабораторних на комп’ютері та клонувати репозиторій.
4. Створити локальну внутрішню структуру папок (Lab0, Lab1 ... Lab6).
5. Надати доступ з правами читання коду для користувача pavlo_serdyuk.
6. Додати посилання на репозиторій у Excel-файл.
7. Пройти 4 рівні на веб-ресурсі https://learngitbranching.js.org.
8. Повторити усі дії в папці Lab0.

ХІД ВИКОНАННЯ

1. Створив акаунт на GitLab.



2. Створив репозиторій та назвав його MAPZ.



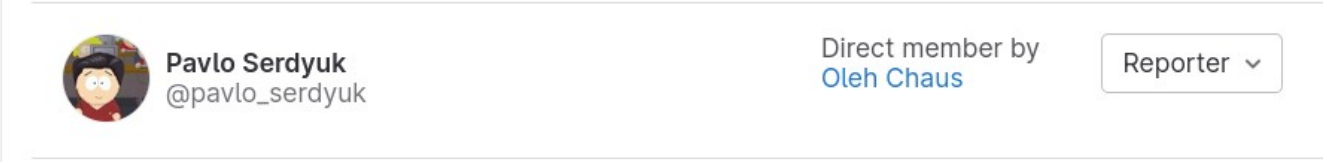
3. Клонував репозиторій на локальну машину.

```
[peachy@fedora Uni]$ git clone https://gitlab.com/oleh-chaus/mapz.git
Cloning into 'mapz'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

4. Створив папки для лабораторних робіт.

```
[peachy@fedora mapz]$ mkdir Lab0 Lab1 Lab2 Lab3 Lab4 Lab5 Lab6
[peachy@fedora mapz]$ ls -l
total 8
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab0
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab1
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab2
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab3
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab4
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab5
drwxr-xr-x. 1 peachy peachy    0 Mar  4 17:14 Lab6
-rw-r--r--. 1 peachy peachy 6187 Mar  2 13:29 README.md
[peachy@fedora mapz]$
```

5. Надав доступ для pavlo_serdyuk.



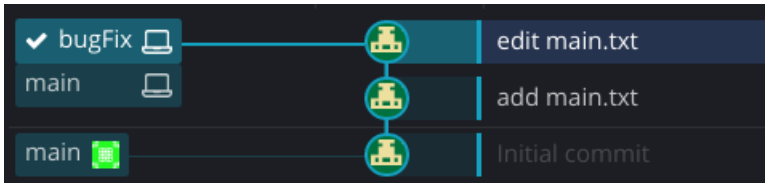
6. Проробив усі дії на сайті https://learngitbranching.js.org та повторив їх на комп’ютері.

```
[peachy@fedora Lab0]$ nano main.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "add main.txt"
[main 67ce1a5] add main.txt
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/main.txt
[peachy@fedora Lab0]$ nano main.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "edit main.txt"
[main 5318f31] edit main.txt
 1 file changed, 1 insertion(+)
```

Завдання 1-1. Створення комітів.

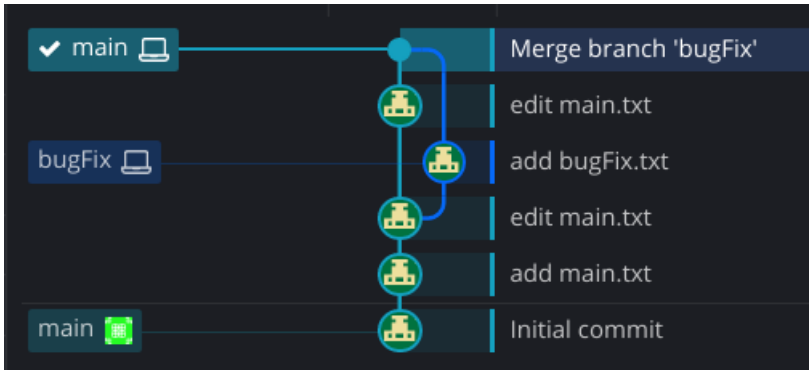
```
[peachy@fedora Lab0]$ git branch bugFix
[peachy@fedora Lab0]$ git checkout bugFix
Switched to branch 'bugFix'
```

Завдання 1-2. Створення гілки “BugFix”.



Візуалізація попередніх дій у програмі GitKraken.

```
Switched to branch 'bugFix'
[peachy@fedora Lab0]$ nano bugFix.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "add bugFix.txt"
[bugFix afd8f4f] add bugFix.txt
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/bugFix.txt
[peachy@fedora Lab0]$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)
[peachy@fedora Lab0]$ nano main.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "edit main.txt"
[main cee3540] edit main.txt
 1 file changed, 1 insertion(+)
[peachy@fedora Lab0]$ git merge bugFix
Merge made by the 'ort' strategy.
 Lab0/bugFix.txt | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/bugFix.txt
```



Завдання 1-3. Мердж гілки bugFix в main.

```
[peachy@fedora Lab0]$ git checkout -b bugFix
Switched to a new branch 'bugFix'
[peachy@fedora Lab0]$ nano bugFix.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "edit bugFix.txt"
[bugFix d90e68d] edit bugFix.txt
 1 file changed, 1 insertion(+)
[peachy@fedora Lab0]$ git checkout main
Switched to branch 'main'
Your branch is ahead of 'origin/main' by 5 commits.
  (use "git push" to publish your local commits)
[peachy@fedora Lab0]$ nano main.txt
[peachy@fedora Lab0]$ git add .
[peachy@fedora Lab0]$ git commit -m "edit main.txt"
[main f04487f] edit main.txt
 1 file changed, 1 insertion(+)
[peachy@fedora Lab0]$ git checkout bugFix
Switched to branch 'bugFix'
[peachy@fedora Lab0]$ git rebase main
Successfully rebased and updated refs/heads/bugFix.
```

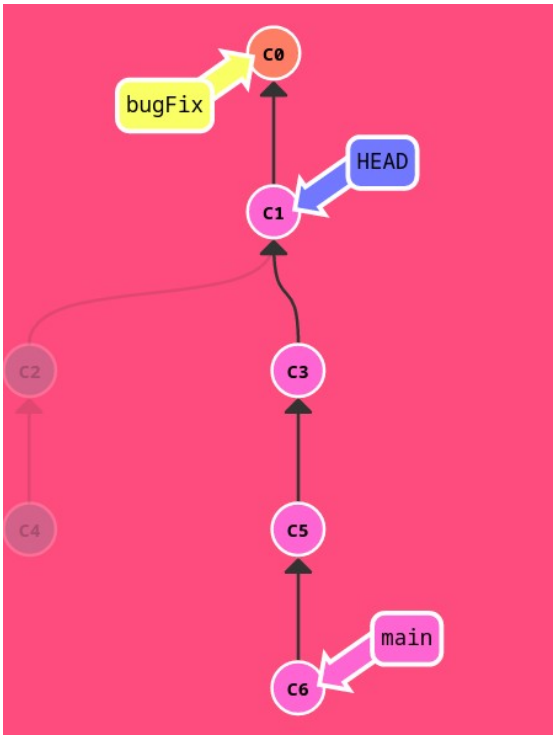
Завдання 1-4. Rebase гілки в main.

```
[peachy@fedora Lab0]$ git checkout 6120
Note: switching to '6120'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at 6120b6e edit bugFix.txt
```

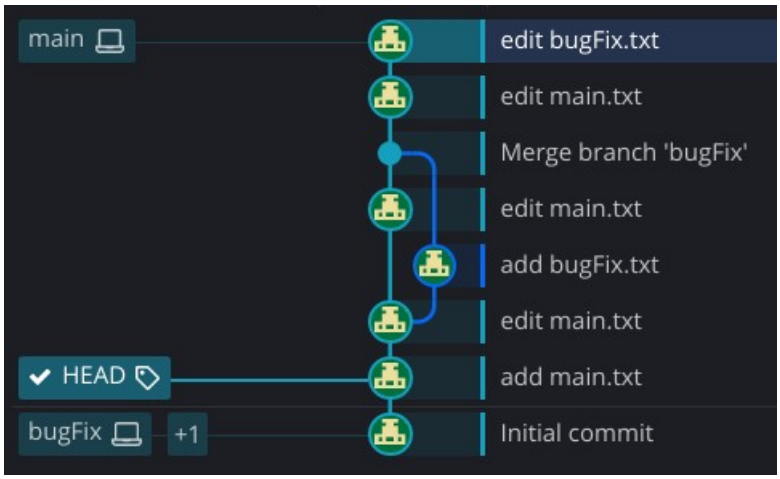
Завдання 2-1. Від’єднання HEAD від поточної гілки.

```
HEAD is now at 6120b6e edit bugFix.txt
[peachy@fedora Lab0]$ git checkout bugFix^
Previous HEAD position was 6120b6e edit bugFix.txt
HEAD is now at f04487f edit main.txt
```

Завдання 2-2. Відносне (^) переміщення HEAD.




```
* 6120b6e (bugFix) edit bugFix.txt
* f04487f (HEAD, main) edit main.txt
*   9e1d7f1 Merge branch 'bugFix'
|\
| * afd8f4f add bugFix.txt
* | cee3540 edit main.txt
|/
* 5318f31 edit main.txt
* 67ce1a5 add main.txt
* 9dd3be6 (origin/main, origin/HEAD) Initial commit
[peachy@fedora Lab0]$ git branch -f bugFix 9dd3
[peachy@fedora Lab0]$ git checkout 67ce
Previous HEAD position was f04487f edit main.txt
HEAD is now at 67ce1a5 add main.txt
[peachy@fedora Lab0]$ git branch -f main 6120
```



Завдання 2-3. Переміщення гілок силоміць.

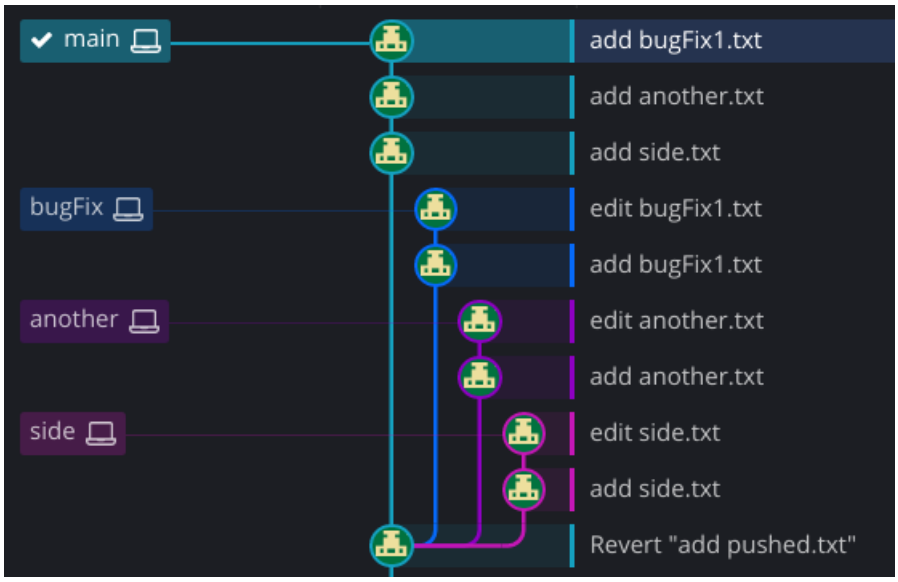
до

```
[peachy@fedora Lab0]$ git checkout local
Switched to branch 'local'
[peachy@fedora Lab0]$ git reset HEAD^
[peachy@fedora Lab0]$ git checkout pushed
Switched to branch 'pushed'
[peachy@fedora Lab0]$ git revert HEAD
[pushed 7f0ed6a] Revert "add pushed.txt"
 2 files changed, 1 insertion(+), 1 deletion(-)
 delete mode 100644 Lab0/pushed.txt
```

після

Завдання 2-4. Скасування змін для гілок local та pushed.

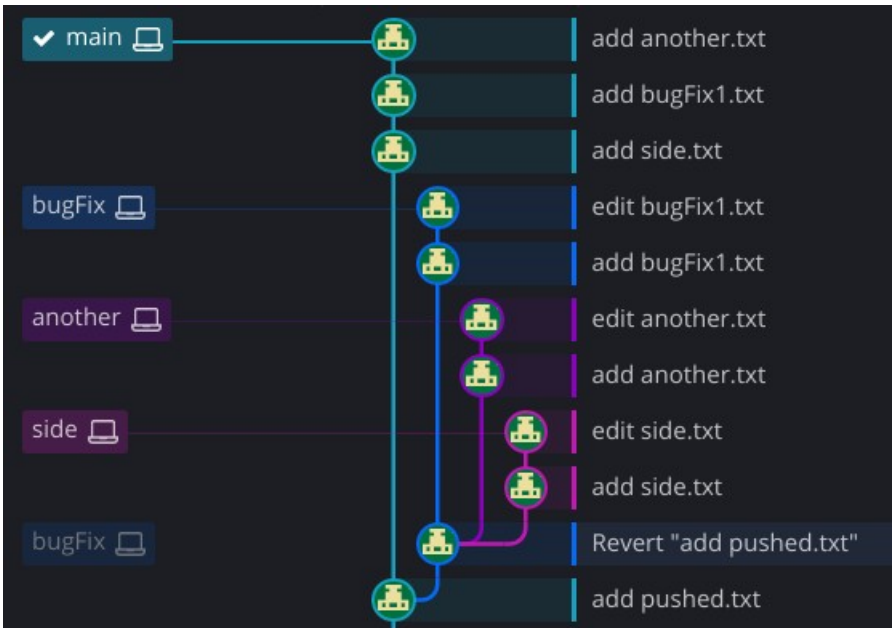
```
[peachy@fedora Lab0]$ git cherry-pick 6a0d aa6c 020a
[main 4b2e045] add side.txt
 Date: Sun Mar 5 16:00:44 2023 +0200
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/side.txt
[main ca1e082] add another.txt
 Date: Sun Mar 5 16:02:19 2023 +0200
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/another.txt
[main 66bdeaf] add bugFix1.txt
 Date: Sun Mar 5 16:05:42 2023 +0200
 1 file changed, 1 insertion(+)
 create mode 100644 Lab0/bugFix1.txt
```



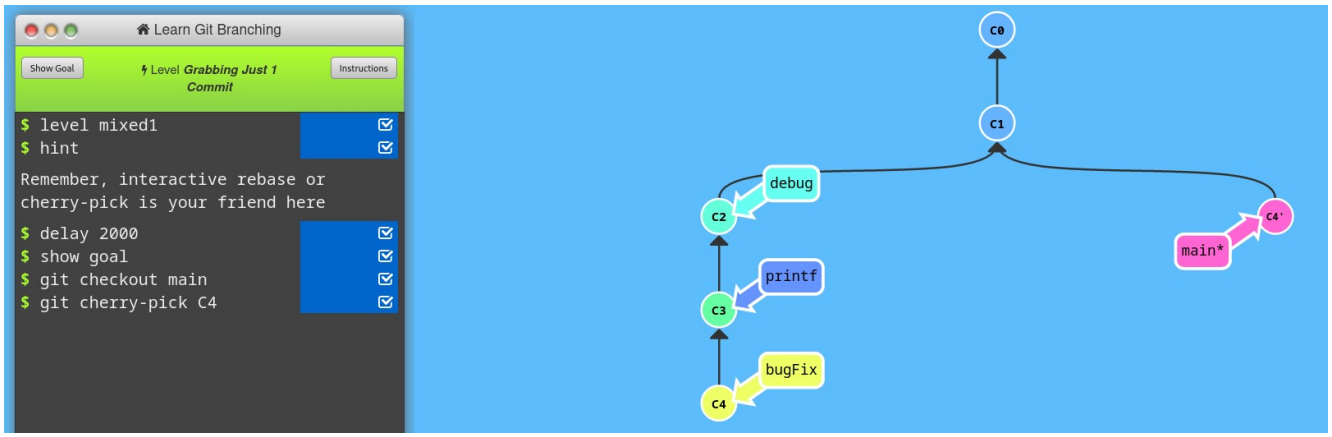
Завдання 3-1. Cherry-pick в main.

```
[peachy@fedora Lab0]$ git rebase -i HEAD~4
Successfully rebased and updated refs/heads/main.
```

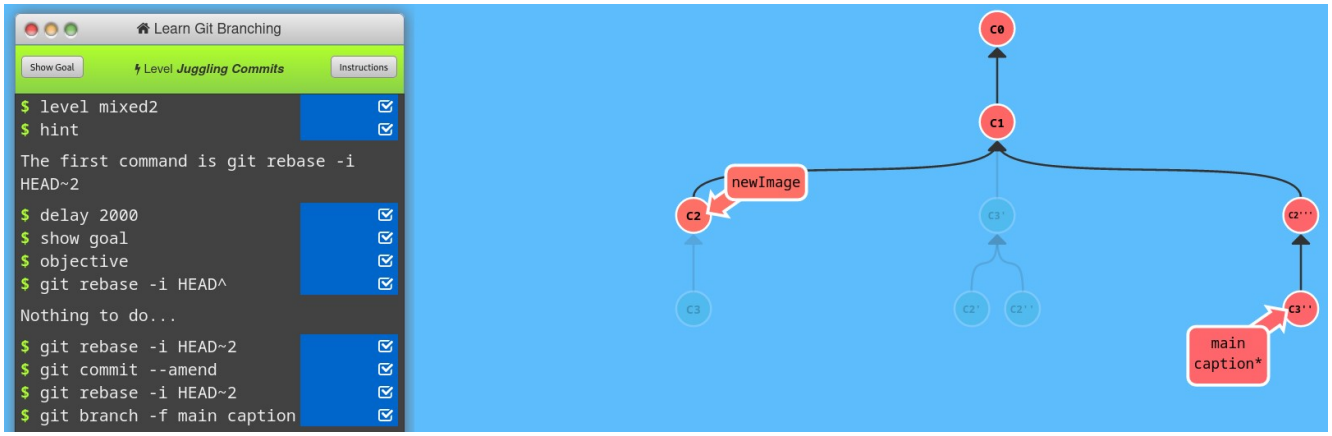
```
drop 7f0ed6a Revert "add pushed.txt"
pick 4b2e045 add side.txt
pick 66bdeaf add bugFix1.txt
pick ca1e082 add another.txt
```



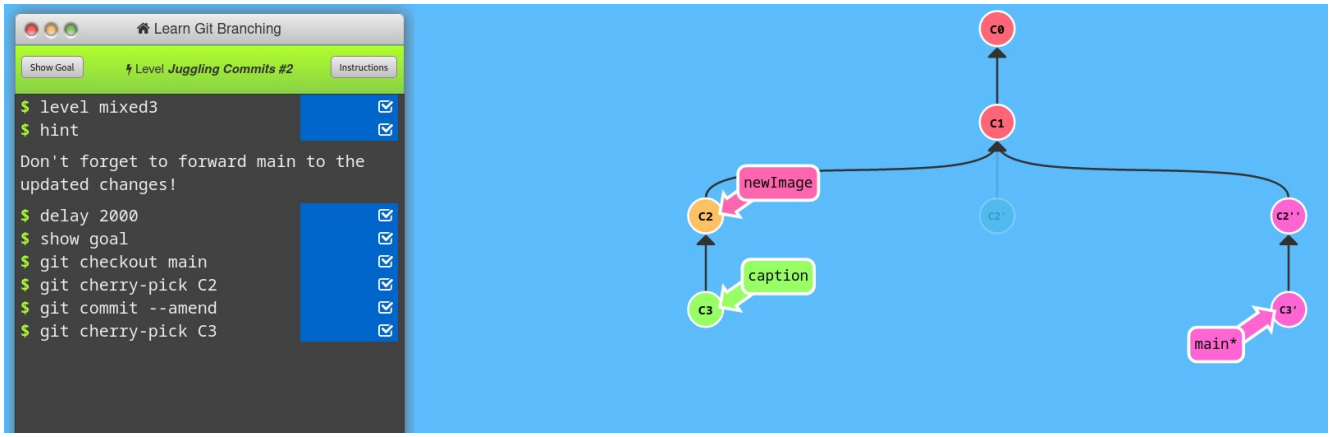
Завдання 3-2. Інтерактивний rebase.



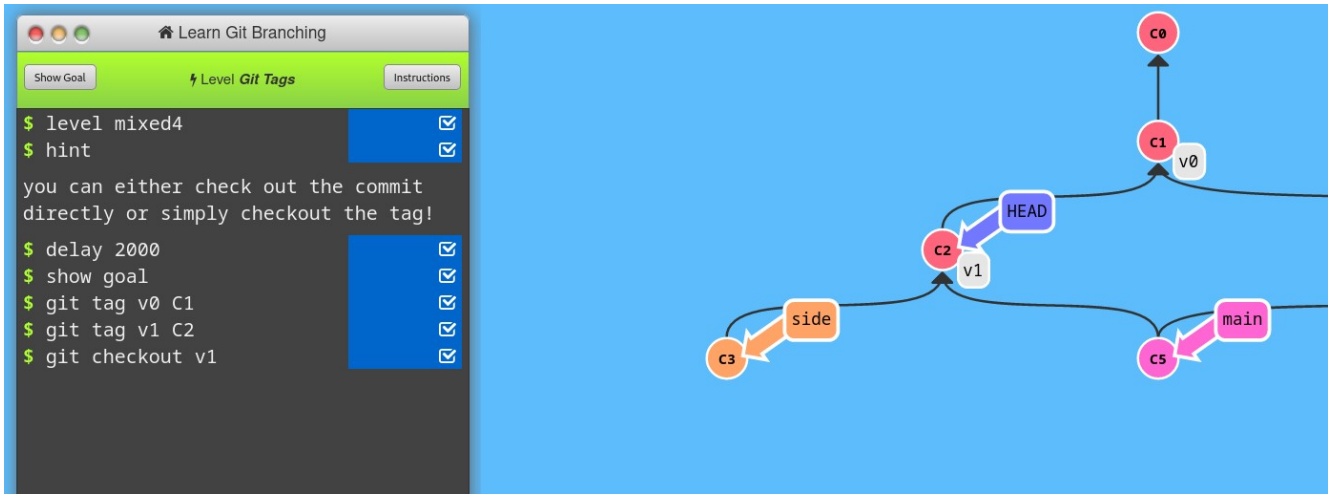
Завдання 4-1.



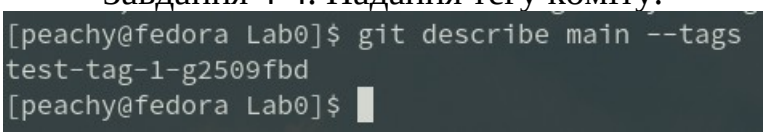
Завдання 4-2.



Завдання 4-3.



Завдання 4-4. Надання тегу коміту.



Завдання 4-5. git describe.

Виконані завдання

ВИСНОВКИ

Під час виконання лабораторної роботи я ознайомився з програмою Git. Навчився створювати коміти та нові гілки. Також розглянув такі команди, як git merge, git rebase, git cherry-pick, які дозволяють керувати проектом.