

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**  
**КАФЕДРА ЕЛЕКТРОННИХ ОБЧИСЛЮВАЛЬНИХ МАШИН**

## **КРОСПЛАТФОРМЕННІ ЗАСОБИ ПРОГРАМУВАННЯ**

### **ВКАЗІВКИ З ОРГАНІЗАЦІЇ РЕПОЗИТОРІЇВ У СИСТЕМІ КОНТРОЛЮ ВЕРСІЙ GIT ТА СЕРВІСІ ХОСТИНГУ ВЕРСІЙ GITHUB**

для студентів базового напрямку «Комп'ютерна інженерія»

## Зміст

Створення віддаленого репозиторію у GitHub.....	3
Система контролю версій Git. Клонування віддаленого репозиторію.....	4
Створення гілок .....	7
Перейменування гілок .....	11
Приклад організації вихідного коду лабораторних робіт у репозиторії Git.....	12
Фіксація змін у гілці з вихідним кодом на локальному і віддаленому репозиторіях .....	14
Злиття гілок що містять вихідні коди з гілкою main.....	16
Видалення гілок .....	19
Перелік використаних джерел .....	20

## Створення віддаленого репозиторію у GitHub

GitHub - сервіс, що дає можливість створювати та завантажувати репозиторії (сховища) проектів програмних продуктів, та використовувати їх у подальшому для спільної віддаленої роботи з іншими розробниками проектів. Репозиторій (сховище) GitHub використовуються для зберігання файлів вихідних кодів програм, документації та інших програмних ресурсів необхідних для роботи над проектами програмних продуктів.

Для створення репозиторію необхідно:

1. Відкрити web-браузер і перейти за посиланням <https://github.com/>
2. Зареєструватись на github.com або увійти в існуючий обліковий запис (рис. 1).

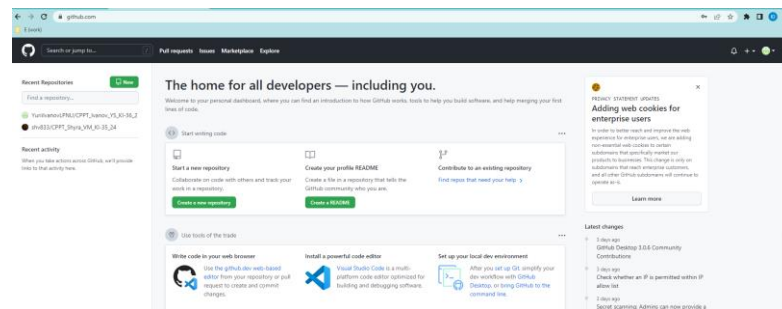


Рис. 1. Сторінка github.

3. Викликати меню користувача і вибрати пункт «Your repositories» («Ваші репозиторії») (рис. 2).

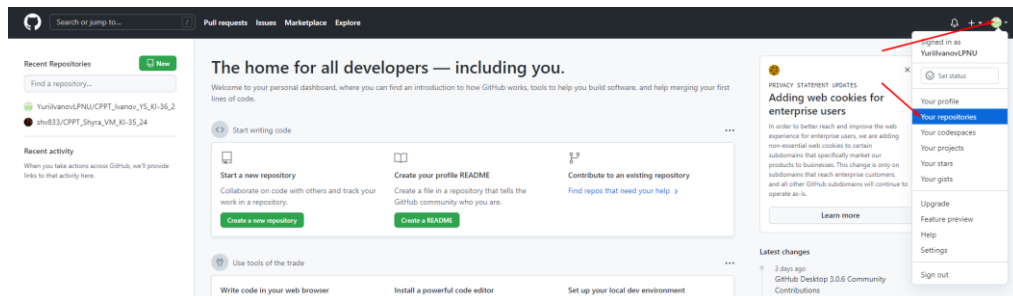


Рис. 2. Меню користувача github.

4. На сторінці репозиторіїв натиснути кнопку «new» (рис. 3)

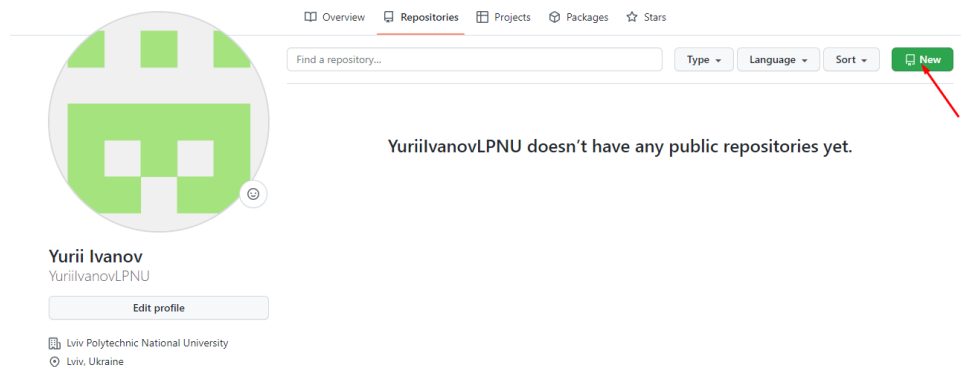


Рис. 3. Сторінка репозиторіїв.

5. На сторінці «Create a new repository» («Створити новий репозиторій») (рис. 4) необхідно:

5.1. У полі «Repository name» («Назва репозиторію») вказати назву репозиторію за таким шаблоном: «CPPT\_LN\_FN\_GN\_SGN», де: LN – прізвище студента (англійською), FN – ініціали студента (англійською), GN – номер академічної групи, SGN – номер підгрупи. Наприклад: «CPPT\_Ivanov\_YS\_KI-34\_1».

5.2. У полі «Description» («Опис») ввести: «Лабораторні роботи з предмету "Кросплатформенні засоби програмування"».

5.3. Відмітити «Add a README file».

5.4. У списку «Add .gitignore» вибрати «.gitignore template: Java».

5.5. Натиснути кнопку «Create repository» («Створити репозиторій»).

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* YuriIvanovLPNU / Repository name \* CPPT\_Ivanov\_YS\_KI-34\_1 ✓

Great repository names are short and memorable. Need inspiration? How about [super-duper-waddle?](#)

Description (optional) Лабораторні роботи з предмету "Кросплатформенні засоби програмування"

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)  
.gitignore template: Java

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)  
License: None

This will set `main` as the default branch. Change the default name in your [settings](#).

① You are creating a public repository in your personal account.

[Create repository](#)

Рис. 4. Сторінка створення репозиторію.

### Система контролю версій Git. Клонування віддаленого репозиторію

Git – розподілена система контролю і керування версіями програмного забезпечення, або іншими проектами що розробляються. Використання Git надає розробникам (або групам розробників) такі основні можливості:

- зберігати різні версії проектів, і, за потреби швидко переходити між потрібними версіями;

- одночасно і незалежно розробляти новий і/або змінювати існуючий функціонал в одному або кількох проектах.

При розробці програмного забезпечення з використанням системи контролю версій Git необхідно встановити систему контролю версій Git. Повний опис процесу завантаження і встановлення Git наведено в [1]. Після завантаження і встановлення системи контролю версій Git, в контекстному меню Windows стануть доступними пункти «Git GUI Here» та «Git Bash Here» (рис. 5).

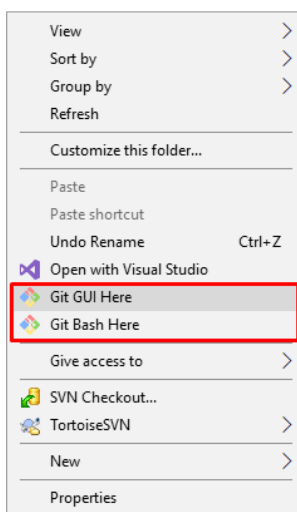


Рис. 5. Приклад контекстного меню Windows після встановлення Git.

Для клонування віддаленого репозиторію на жорсткий диск необхідно:

1. На жорсткому диску створити каталог, в якому будуть розміщуватись репозиторії (наприклад D:\repos\).

2. Перейти в створений каталог і викликати контекстне меню (рис. 5). Для клонування за допомогою Git GUI необхідно:

2.1. Вибрати пункт «Git GUI Here» (рис. 5), в результаті чого на екрані з'явиться вікно застосунку Git Gui (рис. 6). У вікні містяться такі посилання:

- Create New Repository (створити новий локальний репозиторій);
- Clone Existing Repository (клонувати існуючий репозиторій);
- Open Existing Repository (відкрити існуючий репозиторій).

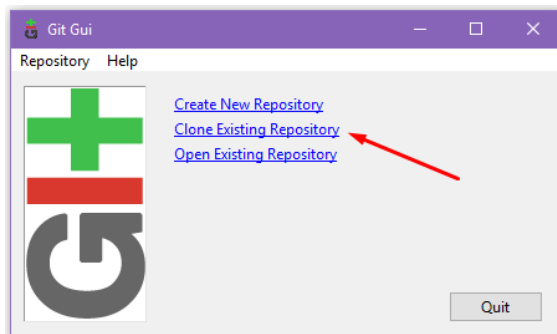


Рис. 6. Вікно застосунку Git Gui.

Після вибору «Clone Existing Repository» на екрані з'явиться вікно в якому вказується місце розташування віддаленого і локального репозиторіїв (рис. 7).

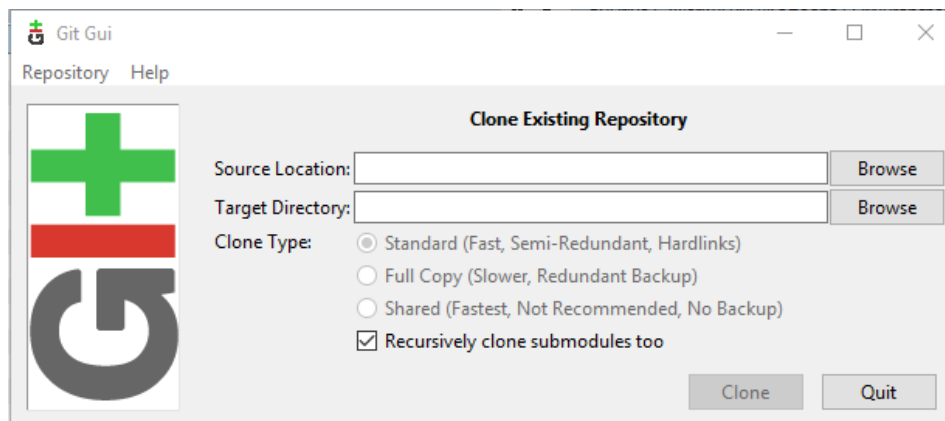


Рис. 7. Вікно клонування репозиторію.

2.2. В браузері перейти на [github.com](https://github.com) і відкрити сторінку створеного репозиторію (рис. 8). На сторінці репозиторію натиснути кнопку «Code» («Код») і скопіювати посилання на віддалений репозиторій (рис. 8). Приклад посилання: [https://github.com/YuriiIvanovLPNU/CPPT\\_Ivanov\\_YS\\_KI-34\\_1.git](https://github.com/YuriiIvanovLPNU/CPPT_Ivanov_YS_KI-34_1.git)

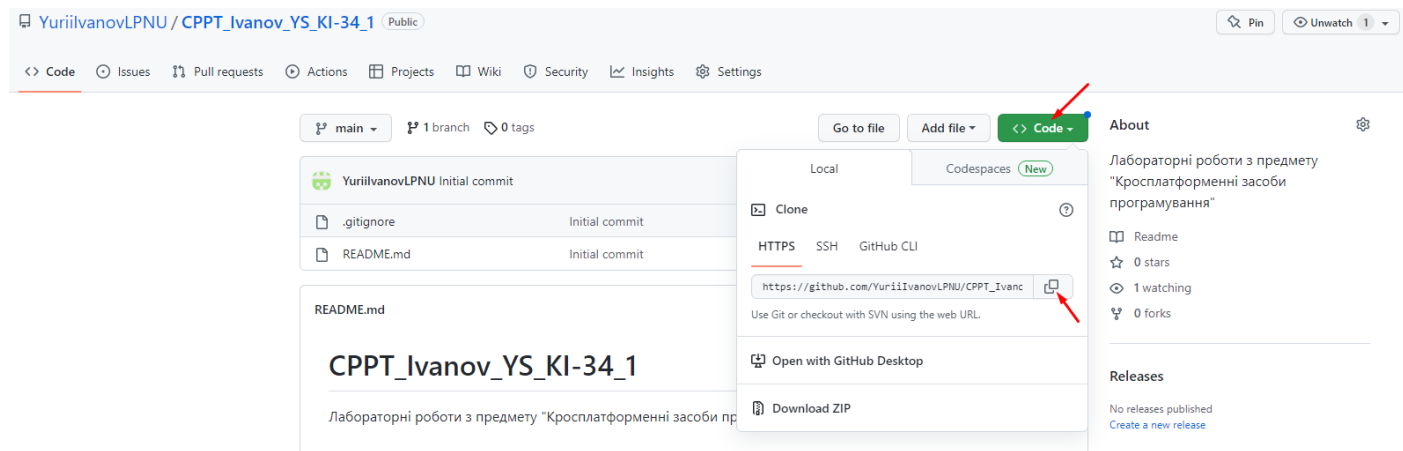


Рис. 8. Копіювання посилання на репозиторій.

2.3. У вікні (рис. 7) у полі «Source Location» («Розташування джерела») вставити (за допомогою комбінації клавіш Shift+Insert) скопійоване в п. 2.2. посилання на віддалений репозиторій, а у полі «Target Directory» («Цільовий каталог») - шлях до каталогу репозиторіїв + підкаталог в якому буде розташовуватись локальна копія репозиторію. Наприклад: «D:\repos\CPPT\_LABS», де «D:\CPPT\_geros» – каталог репозиторіїв. Натиснути кнопку «Clone» («Клонувати»).

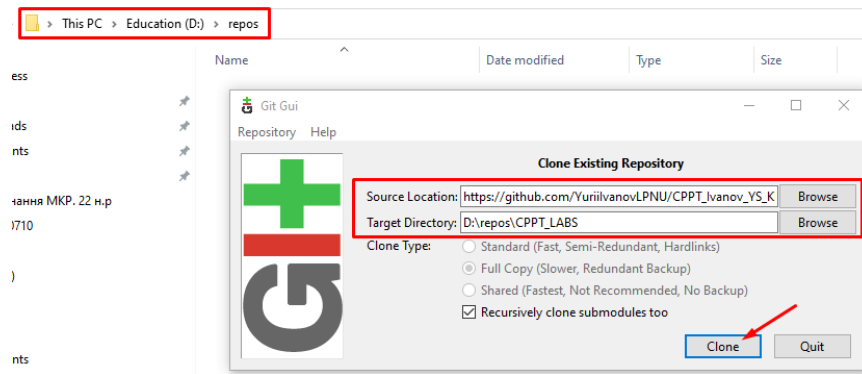


Рис. 9. Приклад заповнення інформації для клонування віддаленого репозиторію.

2.1.4. Після успішного клонування віддаленого репозиторію, в каталозі репозиторіїв буде створено підкаталог, що містить локальну копію віддаленого репозиторію (рис. 10).

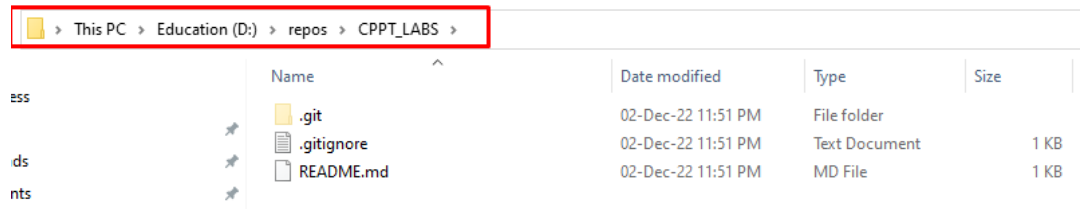


Рис. 10. Приклад локальної копії віддаленого репозиторію.

2.1.5. Скопіювати посилання на репозиторій (рис. 8) і відправити електронною поштою викладачу, який проводить лабораторні заняття.

### Створення гілок

За замовчуванням кожен репозиторій Git містить основну гілку з назвою «main» («головна»). В процесі виконання лабораторних робіт, ця гілка в кожному репозиторії лабораторної роботи повинна бути результуючою, тобто такою, що містить остаточну версію файлів з програмними кодами, які будуть перевірятись викладачем. Під час виконання лабораторних робіт студент може створювати власні гілки, як на основі гілки main, так й інших, попередньо створених власних гілок. Це необхідно наприклад для того, щоб студент мав можливість додати і/або випробувати новий функціонал який додається у вихідний код не зачіпаючи існуючої робочої версії програми. Створювати власні гілки можна як з віддаленого, так і з локального репозиторіїв. Для створення власної гілки з віддаленого репозиторію необхідно:

1. Відкрити сторінку віддаленого репозиторію github (рис. 11) і перейти в розділ «Branch» («Гілка»).

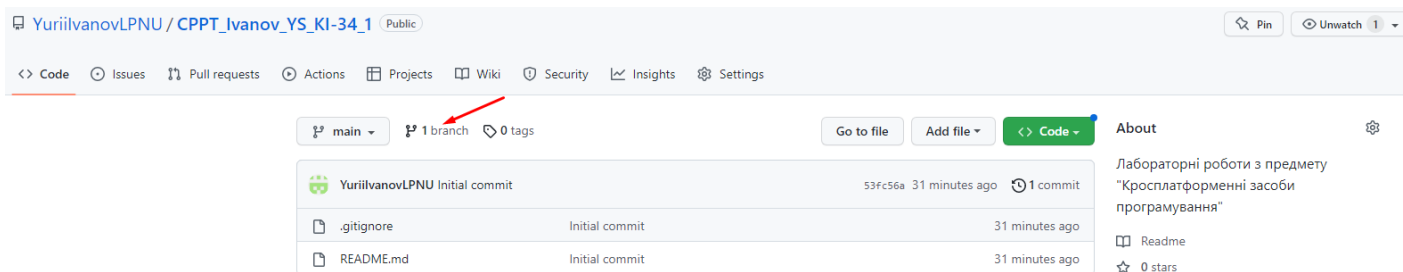


Рис. 11. Сторінка віддаленого репозиторію.

2. Натиснути кнопку «New branch» («Нова гілка») (рис. 12).

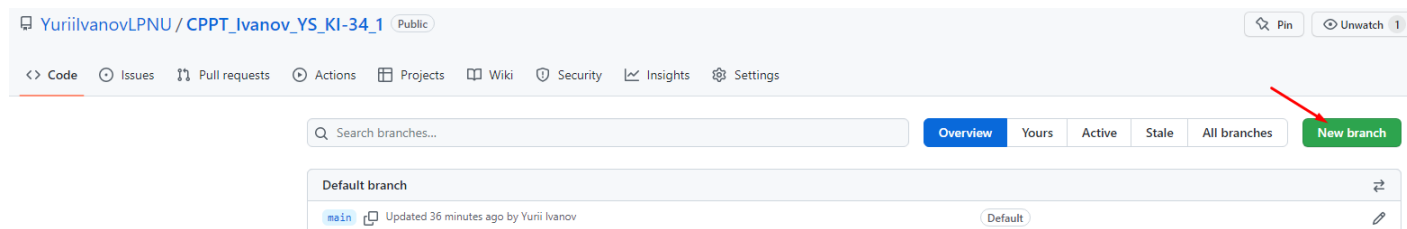


Рис. 12. Сторінка з інформацією про гілки у віддаленому репозиторії.

3. У діалоговому вікні (рис. 13) ввести назву гілки («Branch name»). Назва гілки може бути довільною, але бажано щоб вона відображала короткий опис нового, або модифікації існуючого функціоналу, який додається/змінюється в програмі. Також необхідно вибрати існуючу гілку на основі якої буде створена нова («Branch source»), натиснути кнопку «Create branch» («Створити гілку»).

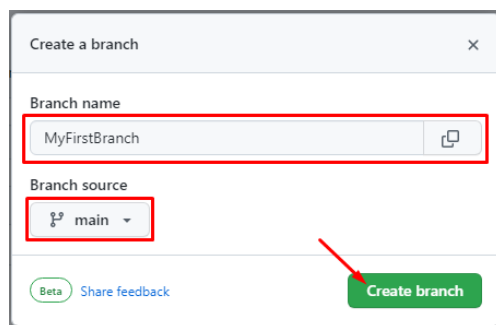


Рис. 13. Діалогове вікно створення нової гілки.

Після чого, на сторінці що містить інформацію про гілки, буде відображена нова створена гілка.

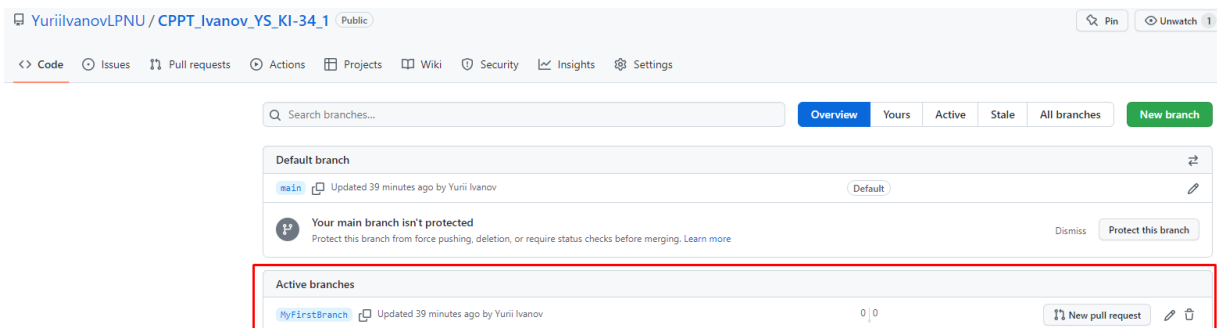


Рис. 14. Вигляд сторінки з інформацією про гілки після створення нової.



4. Перейти в каталог диска де було створено копію віддаленого репозиторію, викликати контекстне меню і вибрати пункт «Git GUI Here» (рис. 5). У головному меню вікна «Git Gui» перейти у підменю «Remote» («Віддалений»), перейти у пункт «Fetch from» («Отримати з») і вибрати підпункт «origin» (рис. 13). В результатів цього з'явиться вікно «Git Gui fetch origin» (рис. 14) і зміни у віддаленому репозиторії стануть доступними в локальному.

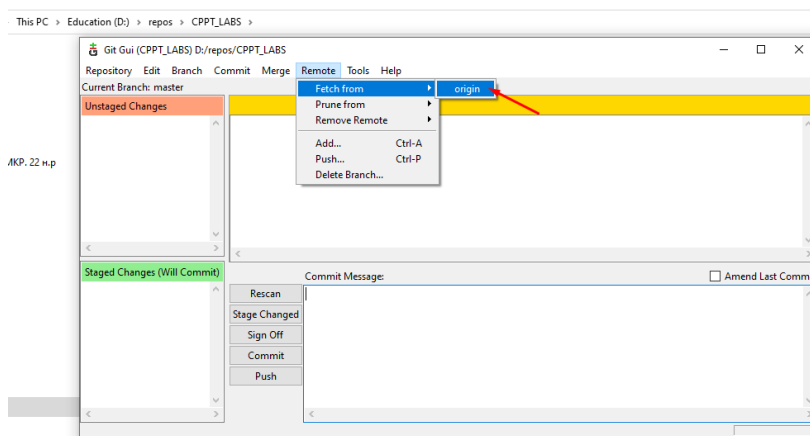


Рис. 13. Отримання змін з віддаленого репозиторію.

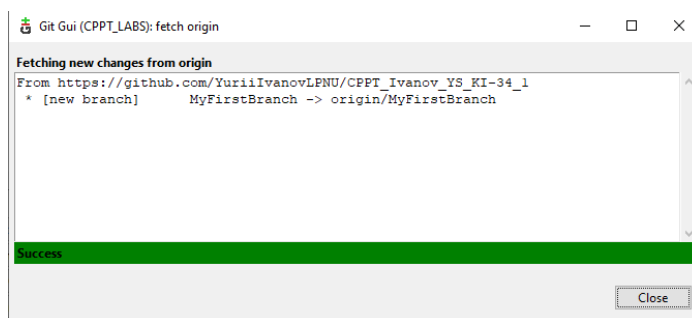


Рис. 14. Вікно «fetch origin».

Для створення власної гілки з локального репозиторію необхідно:

1. У головному меню вікна «Git Gui» перейти у підменю «Branch» («Гілка») і вибрати пункт «Create» («Створити») (рис. 15).

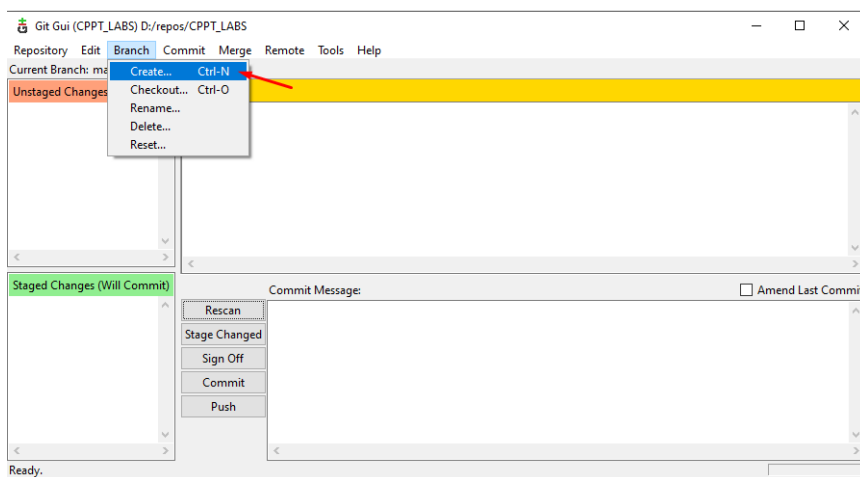


Рис. 15. Створення гілки з Git Gui.

2. У вікні створення гілки (рис. 16) ввести назву гілки, в розділі «Starting Revision» («Початкові ревізії») вибрати «Local Branch», виділити базову гілку на основі якої буде створено нову гілку і натиснути кнопку «Create». В результаті цього, створена гілка стане поточною (рис. 17).

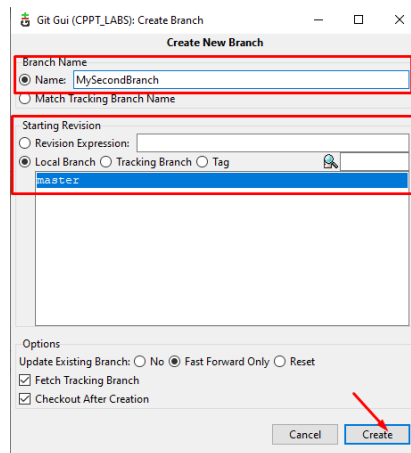


Рис. 16. Вікно створення гілки.

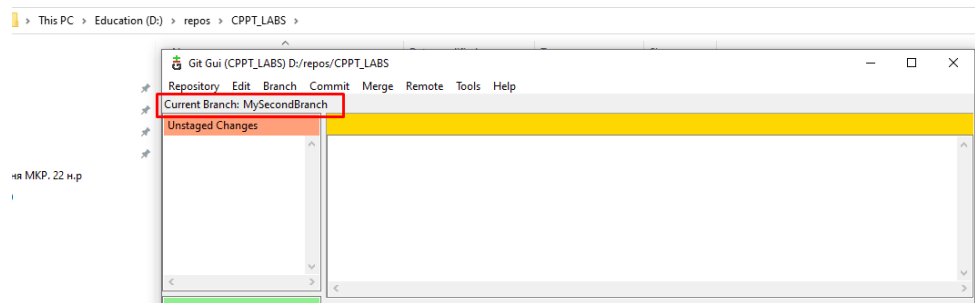


Рис. 17. Відображення поточної гілки.

3. Для того щоб створена локальна гілка потрапила на віддалений репозиторій у вікні «Git Gui» натиснути кнопку «Push», після чого на екрані з'явиться діалогове вікно «Git Gui Push» (рис. 18) в якому необхідно вибрати створену гілку і натиснути кнопку «Push».

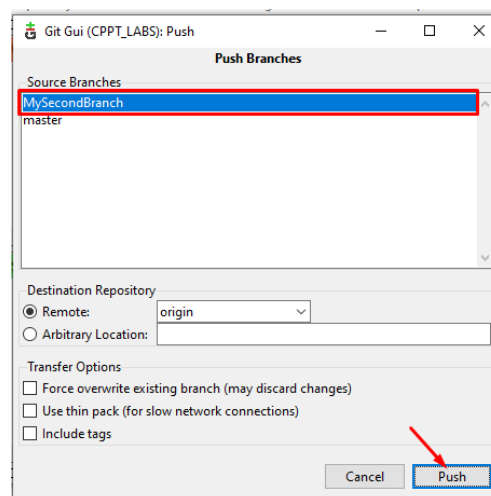


Рис. 18. Відправлення змін на віддалений репозиторій.

Якщо на віддалений репозиторій зміни були внесені успішно, то у вікні «Git Gui Push origin» (Рис. 19) відобразиться відповідний статус «Success», і на сторінці віддаленого репозиторію github (рис. 20) в розділі «Branches» («Гілки») відобразиться гілка що була створена локально.

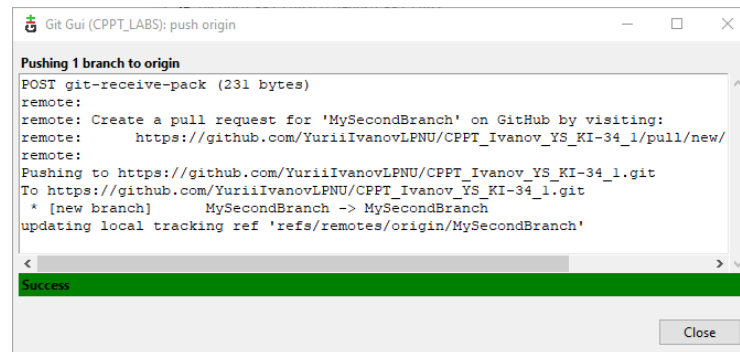


Рис. 19. Вікно «Git Gui Push origin».

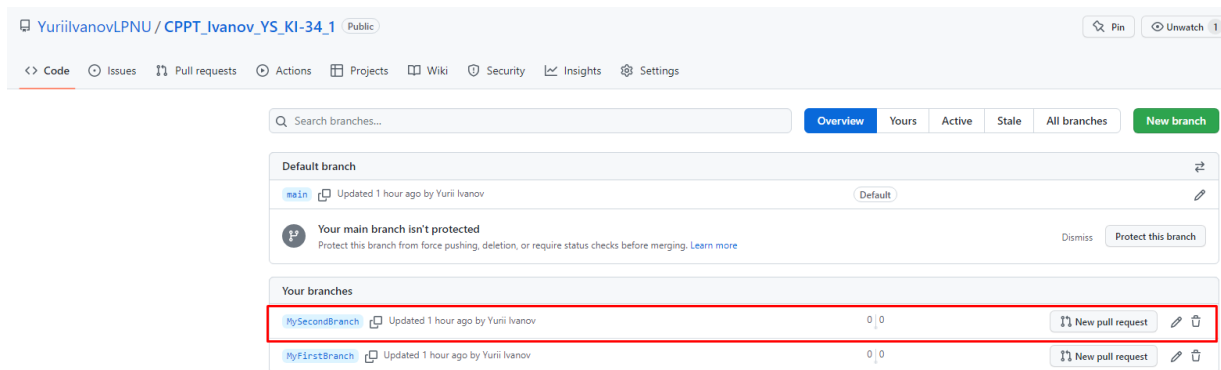


Рис. 20. Сторінка віддаленого репозиторію, що містить гілку, створену з локального репозиторію.

## Перейменування гілок

Для перейменування гілки необхідно:

1. У головному меню вікна «Git Gui» перейти у підменю «Branch» («Гілка») і вибрати пункт «Rename» («Перейменувати») (рис. 21).

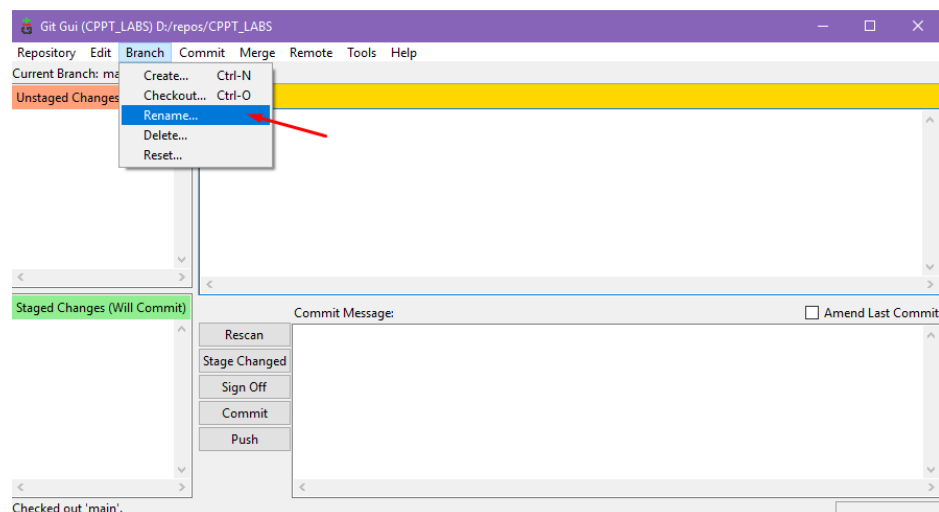


Рис. 21. Перейменування гілки з Git Gui.

2. У вікні «Перейменування гілки» у полі «New Name» («Нова Назва») вказати нову назву гілки (рис. 22) і натиснути кнопку «Rename» («Перейменувати»). Після цього, замість назви поточної гілки (Рис. 17) відобразиться нова назва.

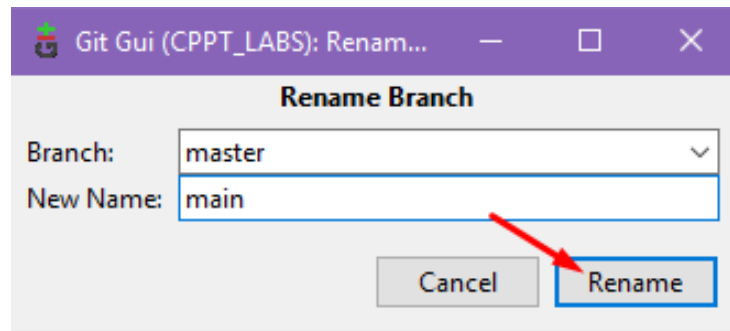


Рис. 22. Перейменування гілки.

### Приклад організації вихідного коду лабораторних робіт у репозиторії Git

Завантаження і встановлення JDK для Java SE та середовища розробки Eclipse наведені в лабораторній роботі № 1 методичних вказівок [2]. Для створення вихідного коду лабораторної роботи у каталозі, що містить локальну копію репозиторію необхідно:

1. Створити гілку, у якій зберігатиметься вихідний код лабораторної роботи (рис. 23).

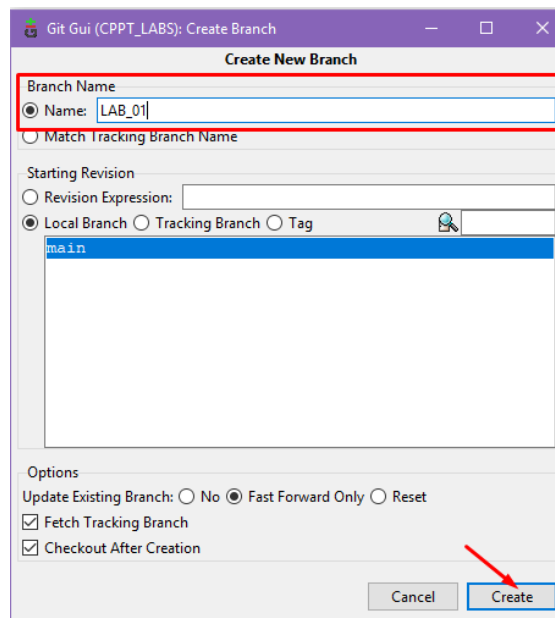


Рис. 23. Приклад створення гілки у якій зберігатиметься вихідний код лабораторної роботи.

2. Переконайтесь, що створена гілка є поточною (рис. 17).
3. У каталозі, що містить локальну копію репозиторію, створити підкаталог у якому зберігатиметься вихідний код лабораторної роботи (рис. 24).

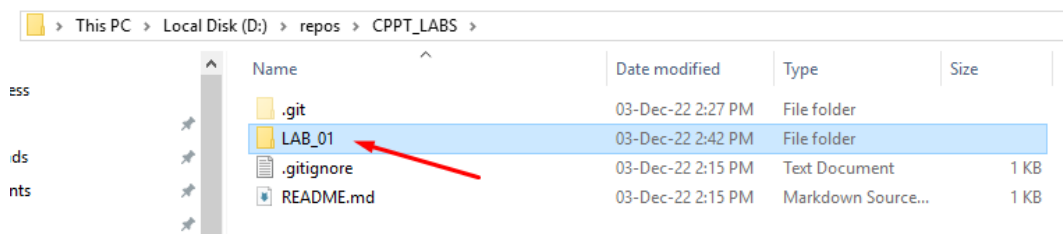


Рис. 24. Каталог, що містить лабораторну роботу.

4. Запустити середовище Eclipse.

5. У вікні «Eclipse IDE Launcher» (рис. 25) у полі «Workspace» («Робочий простір») ввести шлях до каталогу де створено локальний репозиторій і натиснути «Launch» («Запустити»).

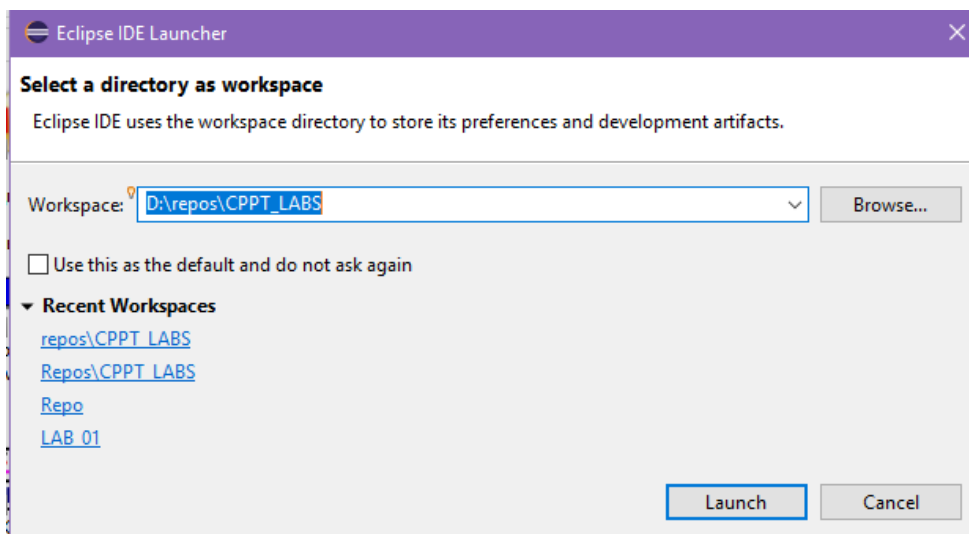


Рис. 25. Вікно «Eclipse IDE Launcher».

6. У середовищі Eclipse створити новий проект, при цьому у полі «Project Name» («Назва проекту») вказати ім'я гілки (рис. 26), яка була створена для зберігання вихідного коду (рис. 23).

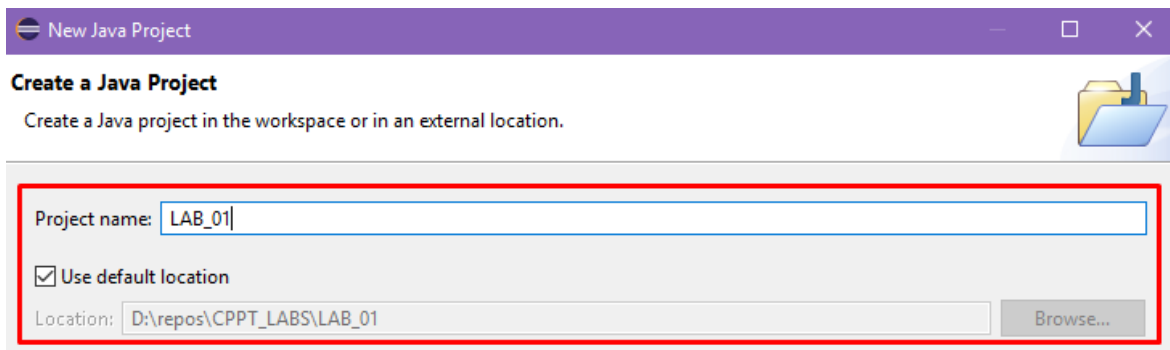


Рис. 26. Назва проекту у середовищі Eclipse.

7. Виконати лабораторну роботу, дотримуючись порядку виконання, що наведений в методичних вказівках [2]. Приклад вмісту каталогу що містить вихідний код лабораторної роботи наведено на рис. 27.

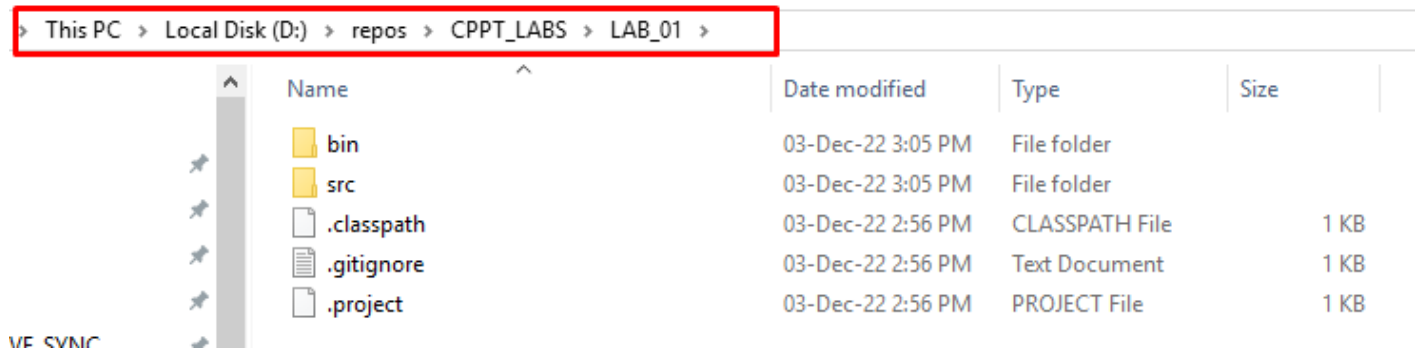


Рис. 27. Каталог, що містить локальну копію репозиторію.

### Фіксація змін у гілці з вихідним кодом на локальному і віддаленому репозиторіях

1. Перейти у каталог з локальним репозиторієм, викликати контекстне меню (рис. 5) і перейти у Git Gui. (рис. 28).

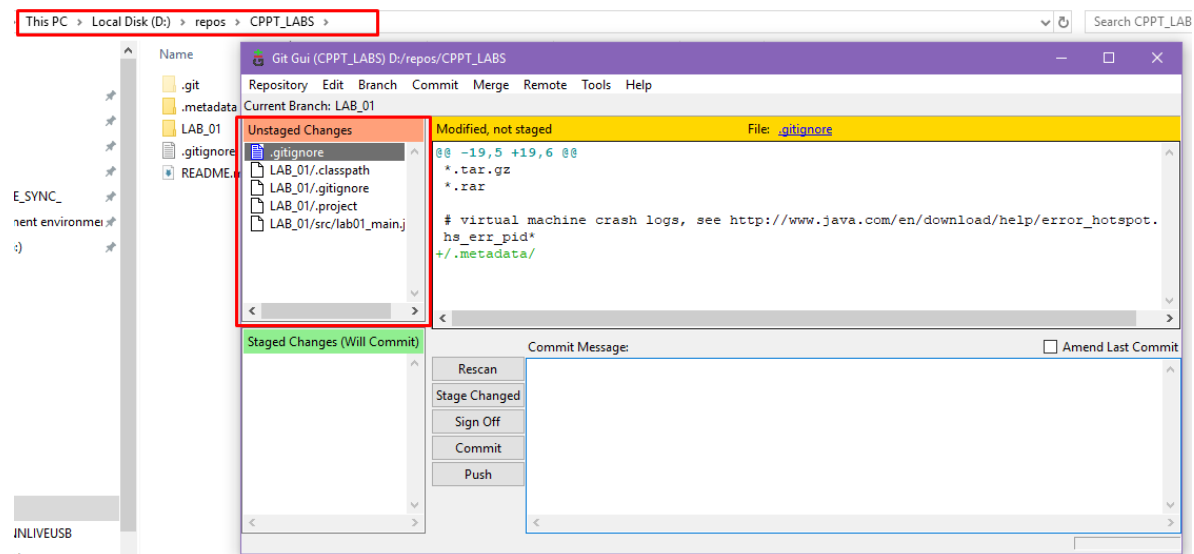


Рис. 28. Приклад вікна Git Gui, що містить невстановлені зміни у вихідному коді.

2. У розділі «Unstaged Changes» («Невстановлені зміни»), відображаються зміни у вихідному коді лабораторної роботи, які були зроблені в результаті її виконання. Переглянути зміни у вихідному коді і натиснути кнопку «Stage Changed» («Встановити зміни»), в результаті чого зміни потрапляють у розділ «Staged Changes» («Встановлені зміни») (рис. 29).

3. У полі «Commit Message» («Повідомлення фіксації») ввести короткий коментар (1-2 речення) який відображає короткий опис змін які були внесені у вихідний код (рис. 30). Натиснути кнопку «Commit» («Зафіксувати») в результаті чого зміни будуть зафіксовані у локальному репозиторії.

4. Натиснути кнопку «Push» («Відправити»), при цьому на екрані відобразиться вікно з опціями відправлення змін на віддалений репозиторій (рис. 31).

5. У вікні відправлення змін (рис. 31) вибрати гілку у яку були зафіксовані зміни у вихідному коді лабораторної роботи, вибрати «Destination Repository» («Репозиторій призначення») – remote origin (віддалений) і натиснути кнопку «Push» («Відправити»). В результаті цього на екрані з'явиться

вікно «Git Gui push origin» в якому відобразиться стан відправлення змін на віддалений репозиторій (рис. 32).

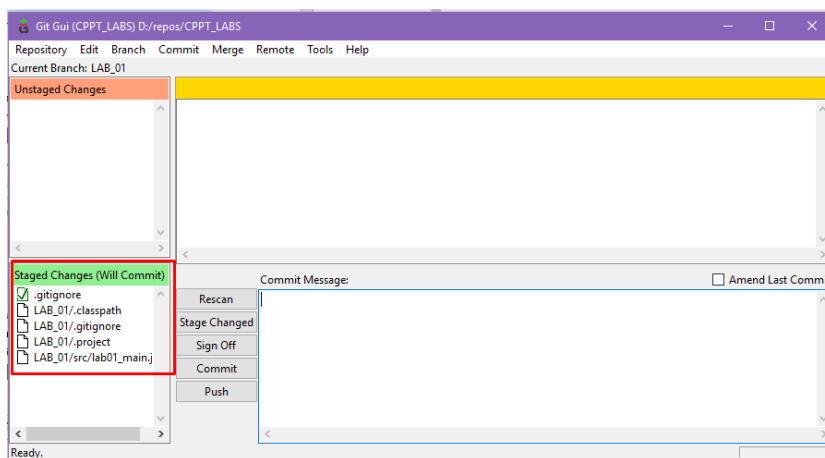


Рис. 29. Приклад вікна Git Gui, що містить встановлені зміни у вихідному коді.

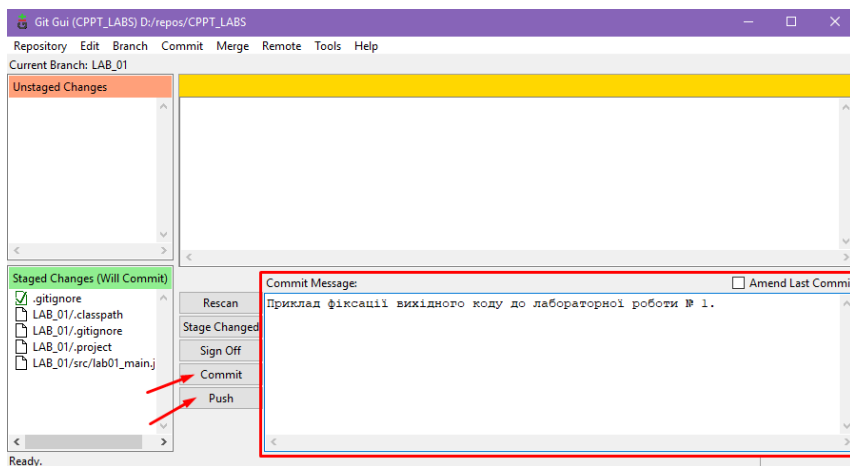


Рис. 30. Приклад коментаря до фіксації у вихідному коді.

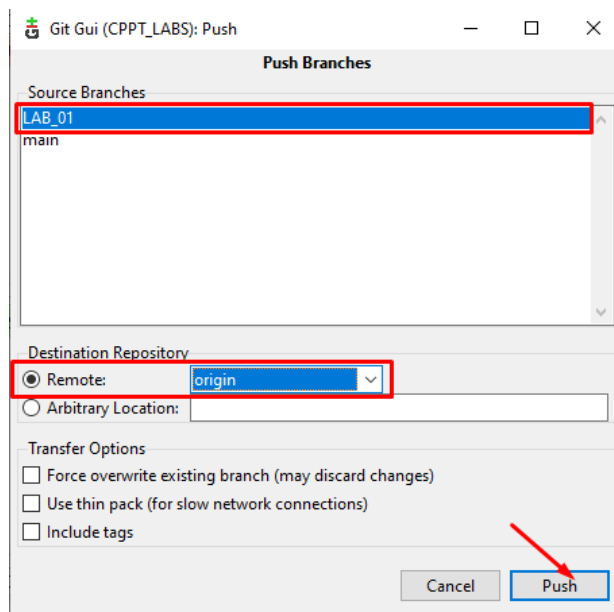


Рис. 31. Вікно з опціями відправлення змін на віддалений репозиторій.

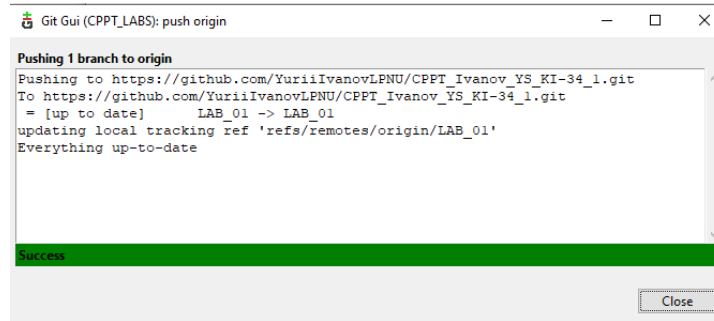


Рис. 32. Вікно стану відправлення змін у віддалений репозиторій.

Щоб переконатись у тому, що зміни були успішно відправлені на віддалений репозиторій необхідно:

1. У віддаленому репозиторії перейти на сторінку з гілками (рис. 11). На сторінці з гілками буде відображено створену гілку до лабораторної роботи (рис. 33).

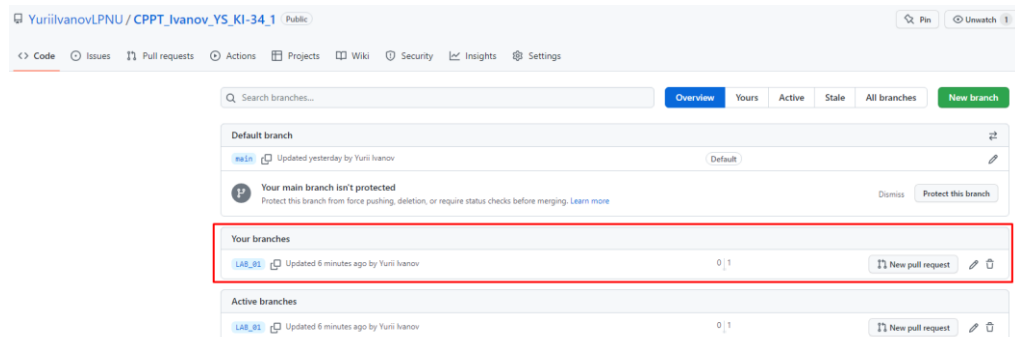


Рис. 33. Сторінка з гілками у віддаленому репозиторії.

2. Перейти до створеної гілки, відкрити каталог «LAB\_01» і перейти у підкаталог «src» (рис. 34).

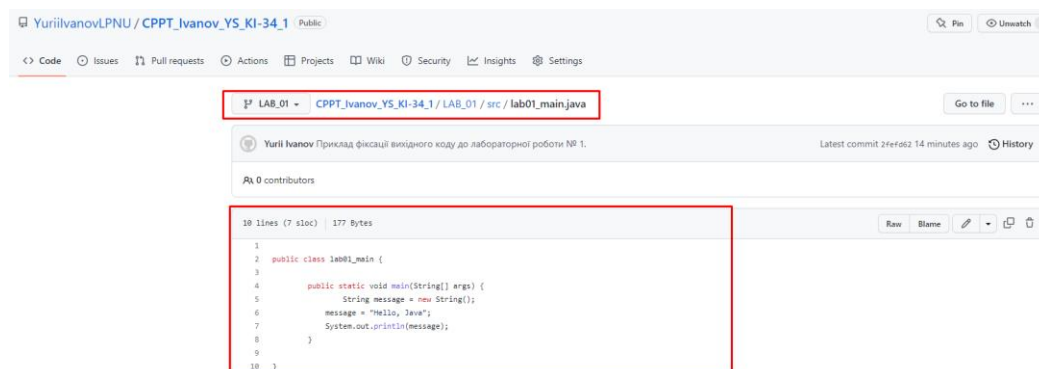


Рис. 34. Приклад відправлених змін у вихідному коді програми на віддалений репозиторій.

### Злиття гілок що містять вихідні коди з гілкою main

Після виконання лабораторної роботи, її вихідний код повинен бути розміщений у гілці main, вміст якої в подальшому перевіряється викладачем. Для розміщення коду у гілці main необхідно виконати злиття вмісту гілки що містить вихідний робочий код лабораторної роботи з гілкою main, а



також виконати фіксацію і відправлення на віддалений репозиторій вмісту гілки main. Для виконання цих дій необхідно:

1. Перейти у каталог з локальним репозиторієм, викликати контекстне меню (рис. 5) і перейти у Git Gui.
2. Зробити поточною гілку main (рис. 17).
3. У меню «Merge» («Злиття») вибрати пункт «Local Merge» («Локальне злиття») (рис. 35), в результаті чого на екрані з'явиться вікно у якому буде відображено перелік гілок які доступні для злиття з гілкою main (рис. 36). У вікні злиття гілок необхідно вибрати гілку з вихідним кодом лабораторної роботи, яку слід злити з гілкою main і натиснути кнопку «Merge» («Злиття»). Після чого, з'явиться вікно у якому буде відображено статус злиття гілок (рис. 37).

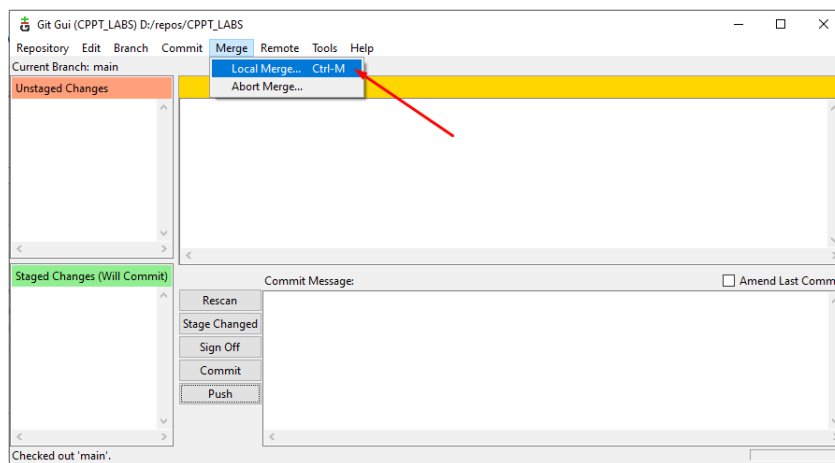


Рис. 35. Виклик меню злиття гілок.

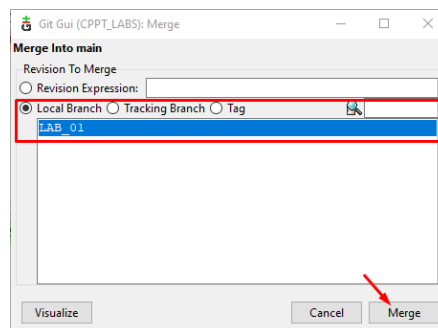


Рис. 36. Вікно злиття гілок.

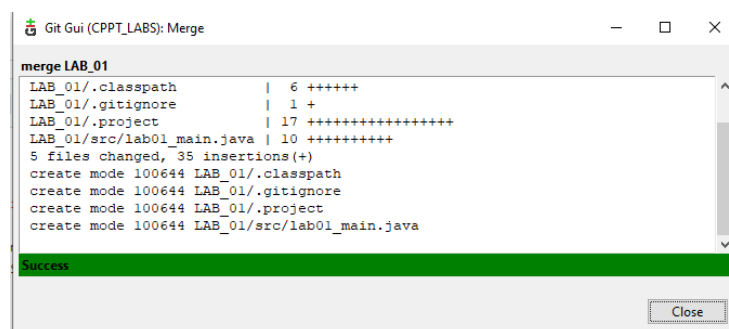


Рис. 37. Вікно статусу злиття гілок.

4. У вікні «Git Gui» натиснути кнопку «Push» («Відправити»), при цьому на екрані відобразиться вікно з опціями відправлення змін на віддалений репозиторій (рис. 38). У вікні, що відобразилось необхідно вибрати гілку main і натиснути «Push» («Відправити»).

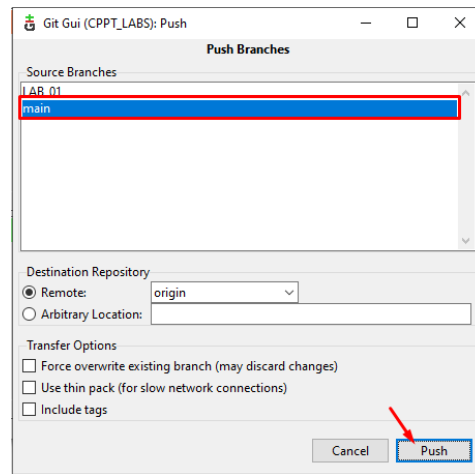


Рис. 38. Вікно відправлення змін на віддалений репозиторій.

В результаті цього на екрані з'явиться вікно «Git Gui push origin» в якому відобразиться стан відправлення змін у гілці main на віддалений репозиторій (рис. 39).

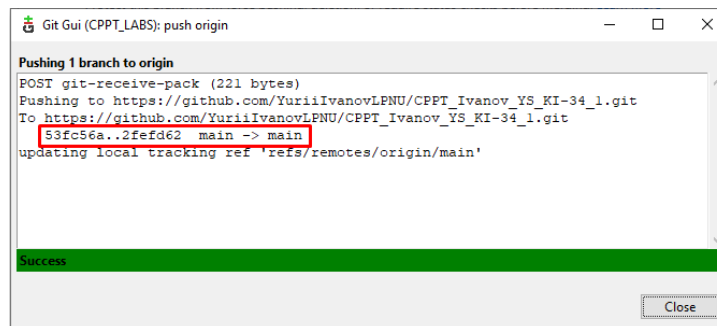


Рис. 39. Вікно стану відправлення змін у гілці main на віддалений репозиторій.

5. Переконайтесь у тому, що зміни були успішно відправлені на віддалений репозиторій. Перейти у розділ «Branches» («Гілки») на віддаленому репозиторії, вибрати гілку main, перейти у каталог «LAB\_01» і підкаталог «src» (рис. 40).

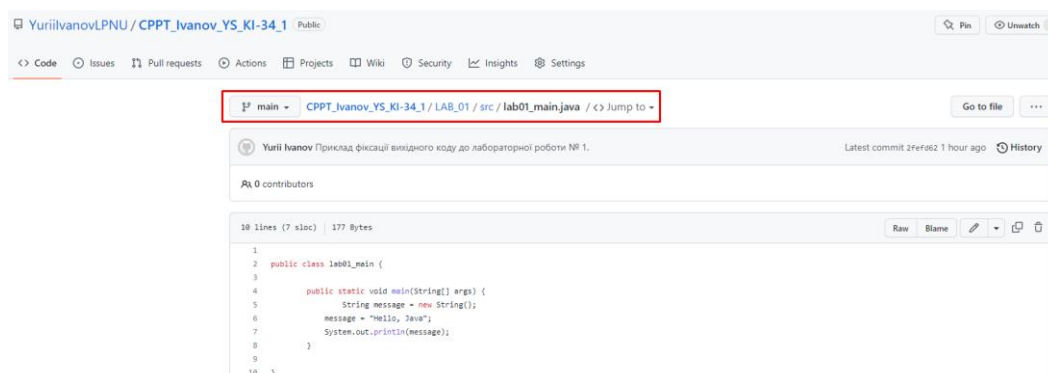


Рис. 40. Приклад відправлених на віддалений репозиторій змін у вихідному коді що міститься у гілці main.

## Видалення гілок

Для видалення гілки на локальному репозиторії необхідно:

1. Перейти у каталог з локальним репозиторієм, викликати контекстне меню (рис. 5) і перейти у Git Gui.
2. У меню «Branch» («Гілка») вибрати пункт «Delete» («Видалити») (рис. 41). На екрані з'явиться вікно у якому буде відображено перелік гілок які доступні для видалення (рис. 42).

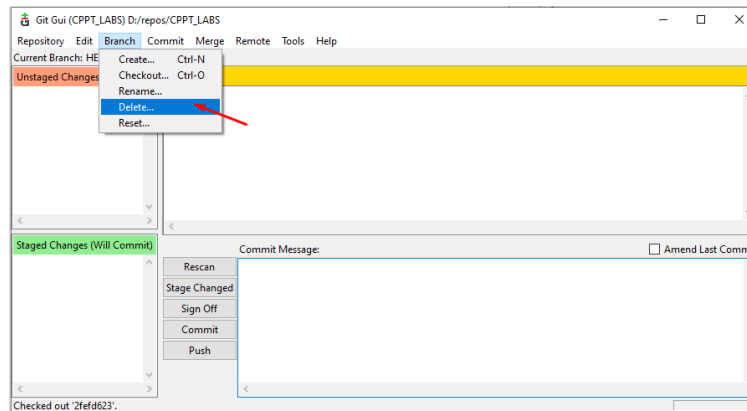


Рис. 41. Меню видалення гілок.

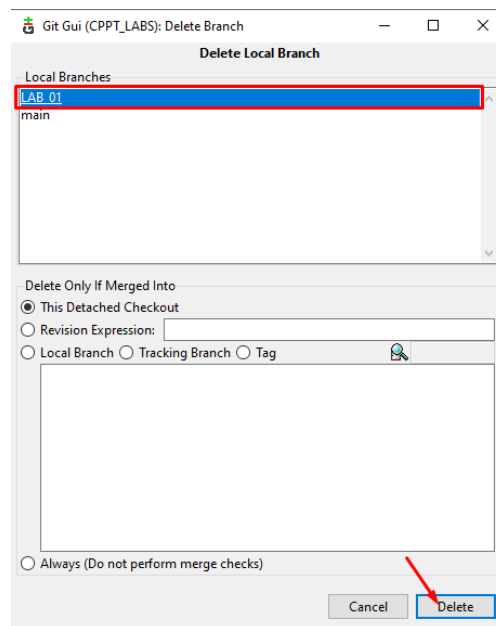


Рис. 42. Вікно з опціями видалення гілок.

3. Вибрати гілку, яку необхідно видалити і натиснути кнопку «Delete» («Видалити»). В результаті цих дій гілку буде видалено, а гілка main стане поточною.

Для видалення гілки на віддаленому репозиторії необхідно:

1. Відкрити сторінку віддаленого репозиторію github (рис. 11) і перейти в розділ «Branch» («Гілка»).

2. У розділі «Your branches» («Ваші гілки») натиснути кнопку видалення (рис. 43) і оновити сторінку відображення гілок натиснувши F5.

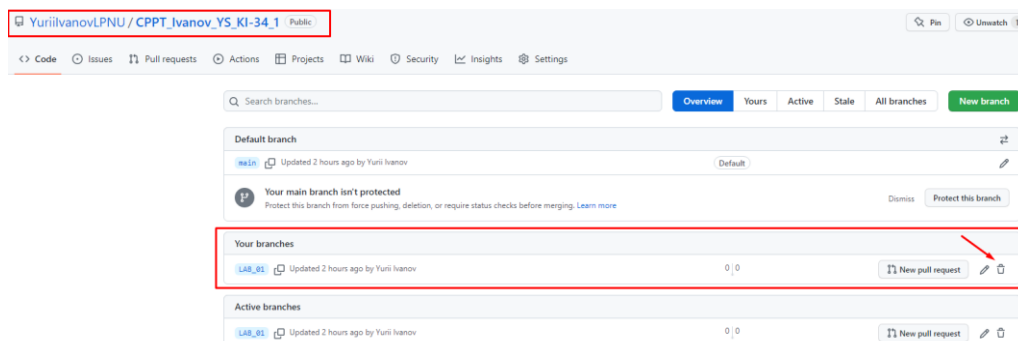


Рис. 43. Видалення гілки на віддаленому репозиторії.

### Перелік використаних джерел

1. Встановлення Git [електронний ресурс]. – Режим доступу: <https://cherto4ka.xyz/2021/11/04/установка-git/>
2. Кросплатформенні засоби програмування: Методичні вказівки до виконання лабораторних робіт для студентів базового напрямку ”Комп’ютерна інженерія”. Укл.: М.В. Олексів – Львів: Видавництво Національного університету “Львівська політехніка”, 2014. – 90 с.