МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Факультет радіотехніки, комп’ютерних систем і інфокомунікацій

Кафедра інформаційно-комунікаційних технологій ім. О. О. Зеленського

1. **Практична робота №1**

з дисципліни «Технології неперервної інтеграції і розгортання ІС»

Виконав: студент 4 курсу групи № 548

напряму підготовки (спеціальності)

126 Штучний інтелект ті інформаційні системи

(шифр і назва напряму підготовки (спеціальності))

Косянчук Максим Олександрович

(прізвище й ініціали студента)

Прийняв:

(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_

Харків – 2024

**Завдання на практичне заняття №1**

**Основи роботи з Git**

**Мета:** Ознайомитися з основами роботи з системою контролю версій Git, навчитися створювати репозиторій, комітити зміни у файлі, працювати з віддаленими репозиторіями, співпрацювати з іншими розробниками.

**Зміст заняття**

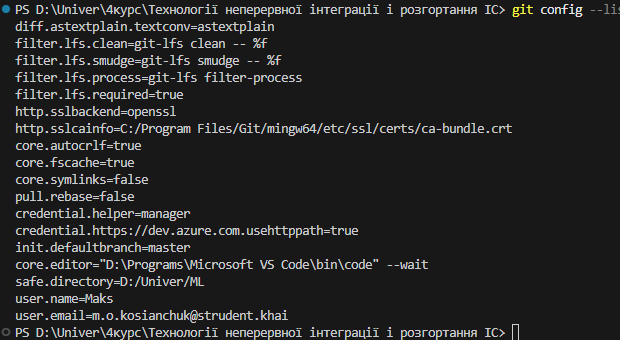
1. Встановлення та налаштування Git.
2. Основні команди роботи з Git.
3. Робота з віддаленими гілками.
4. Вирішення конфліктів.

Для кожного завдання (заданого цифрою) у звіті необхідно навести скрін-шот терміналу (або його частини, але обов'язково, щоб було видно рядок запрошення до введення, наприклад, “**mivanov@pc: ~$**“) з Вашою командою, результатом виконання команди, вмістом файлу (за необхідністю).

**Завдання:**

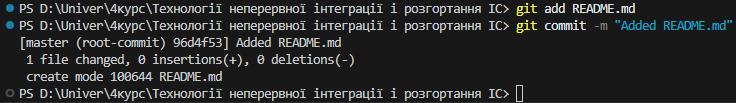
1. 1. Встановіть Git на ваш комп'ютер.
   * Завантажте та встановіть останню версію Git з Офіційного сайту - https://git-scm.com/.
   * Відкрийте командний рядок (термінал, консоль, Command Prompt).
   * Введіть git --version, щоб перевірити, що Git встановлений.



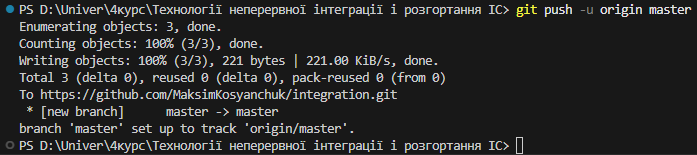
1. Налаштуйте Git.
   * Встановіть своє ім'я та електронну пошту, які будуть використовуватись для підписання комітів
   * Перевірте поточні налаштування:
   * 
2. Створіть локальний репозиторій Git.
   * Створіть нову папку для проекту на вашому комп'ютері.



1. Додайте файли та зробіть коміт.
   * Створіть файл у каталозі вашого проекту, наприклад, README.md.
   * Додайте зміни у файлі до індексу
   * Зафіксуйте зміни у репозиторії



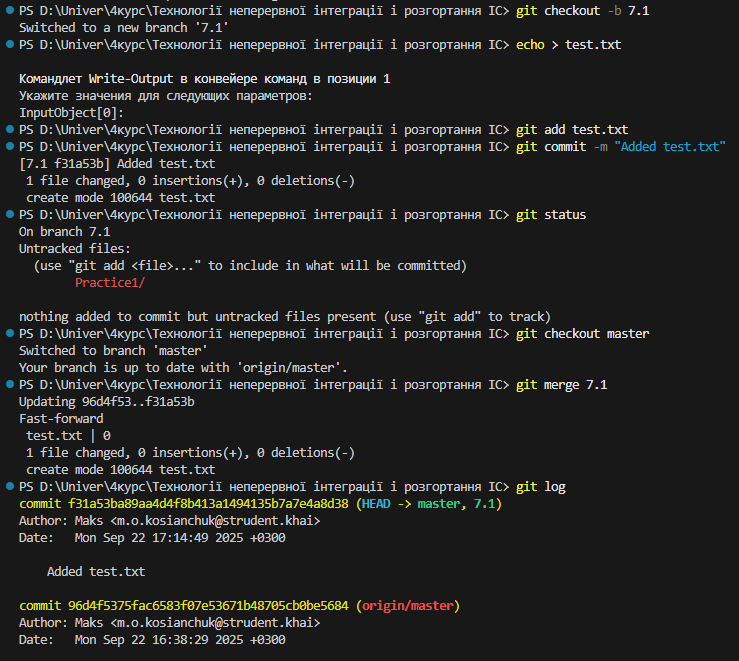
1. Робота з віддаленим репозиторієм.
   * Зареєструйте акаунт на GitHub (https://github.com/), якщо у вас його ще немає.
   * Створіть новий віддалений репозиторій на GitHub.
   * Додайте віддалений репозиторій у ваш локальний проект:
   * Завантажте коміти з локального репозиторію до віддаленого репозиторію на Github:

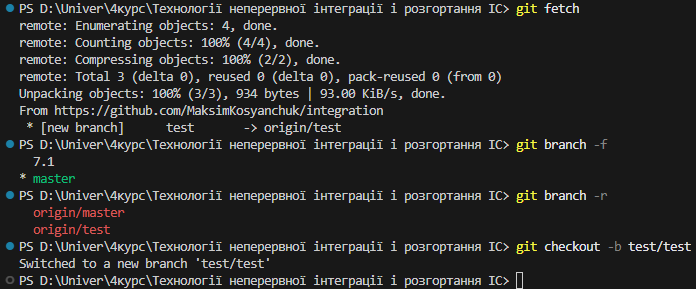
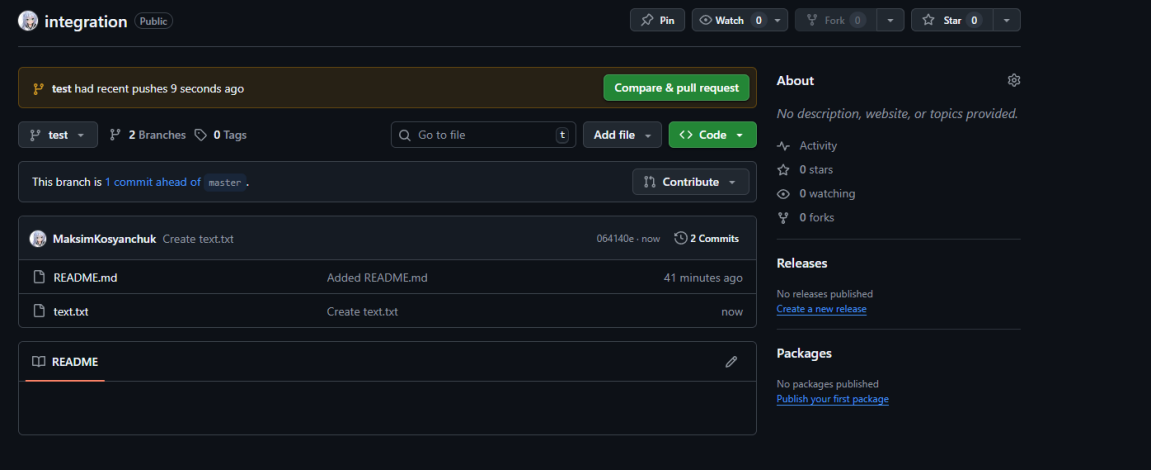


1. Співпраця з іншими розробниками.
   * Ознайомтеся з такими командами Git, як:

* git branch - для створення гілок для роботи над окремими завданнями.
* git checkout - для перемикання між гілками.
* git merge - для зливання гілок та інтеграції змін із різних гілок.
* git pull - для отримання останніх змін з віддаленого репозиторію та додавання їх до вашого локального репозиторію.

1. Робота з гілками.
   * Створіть нову гілку для роботи над однією з функцій вашого проекту:
   * Робіть зміни у файлах, додавайте їх до індексу та комітіть, аналогічно до попередніх кроків. Зверніть увагу, що ці зміни будуть додані тільки до поточної гілки.
   * Поверніться до основної гілки (main або master, залежно від версії git) та злийте дві гілки разом:
   * Тепер зміни з вашої гілки внесені до основної гілки. Ви можете видалити свою робочу гілку, якщо вона вам більше не потрібна:



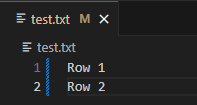
1. Робота з віддаленими гілками.
   * Коли ви працюєте з гілками, неминуче будуть віддалені гілки, які створили ваші колеги.
   * Створіть нову гілку в GitHub за допомогою UI.
   * Завантажте на GitHub файл на ваш вибір, наприклад test3.txt
   * Щоб отримати список всіх віддалених гілок, виконайте:
   * Щоб створити локальну копію віддаленої гілки, виконайте:
   * Тепер ви можете працювати з цією гілкою локально. Коли ви будете готові внести свої зміни до віддаленого репозиторію, виконайте: 
2. Вирішення конфліктів.

Іноді в процесі співпраці виникають конфлікти, коли дві особи вносять зміни до одного блоку коду. Щоб вирішити конфлікт, треба відкрити файл з конфліктом, знайти спірні моменти, зазначені обидвома особами, вирішити, який варіант змін коректний, видалити

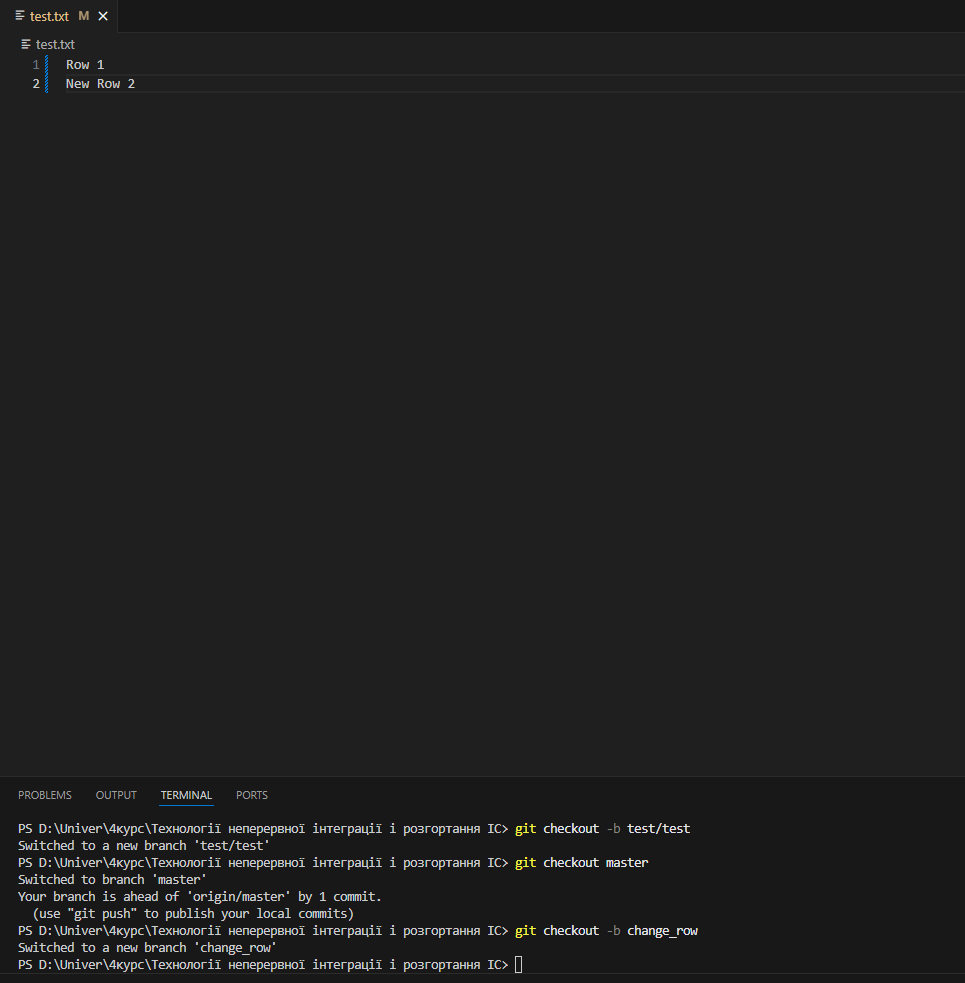
зайве і зробити коміт.

Далі потрібно зберегти зміни, додати їх до індексу та зробити коміт, аналогічно до попередніх кроків. Коміт повинен містити повідомлення про виправлення конфлікту.

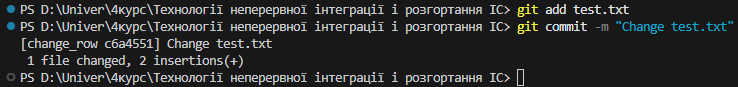
Розглянемо простий практичний приклад створення й вирішення конфлікту в Git.



* + Створіть й переключіться на нову гілку feature\_branch:
  + Зробіть зміни в файлі example.txt. Змініть другий рядок на "Змінений рядок 2".



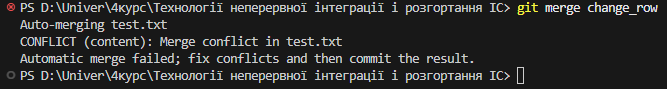
* + Додайте й збережіть зміни у файлі.



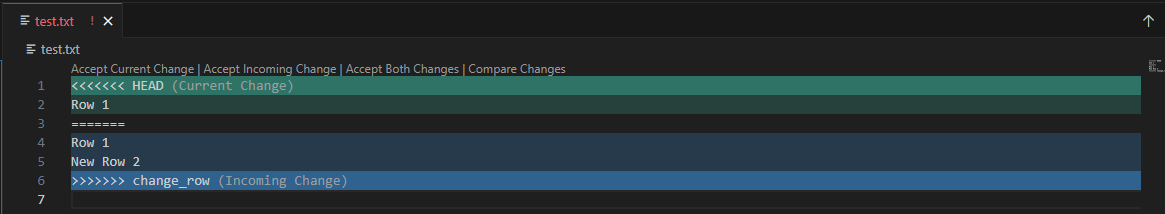
* + Поверніться на головну гілку.
  + Зробіть іншу зміну в файлі example.txt на головній гілці, змінивши другий рядок на "Новий рядок 2".

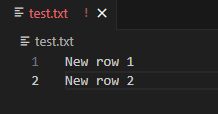
Тепер маємо конфлікт у файлі example.txt, коли спробуємо злити гілку feature\_branch.

* + Спробуйте злити гілки.

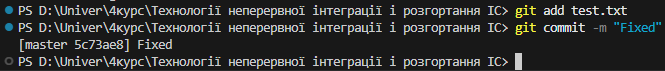


Це повідомлення вказує на конфлікт у файлі "example.txt".

* + Відкрийте файл example.txt у текстовому редакторі
  + Вирішіть конфлікт, домовившись зі своїми колегами, змінивши його на "Узгоджений рядок 2".



* + Збережіть файл та зробіть коміт.



1. Розуміння Git Workflow.
   * Для ефективної роботи з командою дотримуйтесь визначеної стратегії роботи з Git, такими як GitHub Flow або Gitflow Workflow.
   * Ознайомтеся з поняттями pull request, code review для поліпшення роботи вашої команди та якості коду.
   * Вивчення основ роботи з Git є дуже важливим для комфортної роботи в команді програмістів. В подальшому, вивчайте додаткові команди та інструменти відповідно до вашої конкретної роботи та завдань.