**Київський фаховий коледж зв’язку**

**Циклова комісія комп’ютерної та програмної інженерії**

(назва циклової комісії)

ЗАТВЕРДЖУЮ

Заступник директора

з навчальної роботи

\_\_\_\_\_\_\_\_\_ В.С. Шматко

«29» серпня 2025 р.

### 

**Методичний посібник**

**“WORK-CASES IN LINUX”**

з навчальної дисципліни

**“Операційні системи“**

для самостійної роботи студентів денного та заочного відділення

| **Спеціальності** | **121 “Інженерія програмного забезпечення”**  **123 “Комп’ютерна інженерія”**  **125 “Кібербезпека”** |
| --- | --- |
|  | (код і назва спеціальності) |

Укладач Сушанова В.С.

РОЗГЛЯНУТО ТА РЕКОМЕНДОВАНО

на засіданні циклової комісії

Комп’ютерної та програмної інженерії

Протокол № 1

від “28” серпня 2025 р.

Голова циклової комісії КПІ

\_\_\_\_\_\_\_\_\_ Вікторія СУШАНОВА

2025

**ЗМІСТ**

**1 ЗАГАЛЬНІ ВІДОМОСТІ 2**

**2 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ 3**

**3 ТЕМАИКА ЗАВДАНЬ САМОСТІЙНОЇ РОБОТИ 4**

3.1 Практичне завдання “Work-case 1” 4

**4 ПРИКЛАД ВИКОНАННЯ ПРАКТИЧНИХ ЗАВДАНЬ 8**

4.1 Оформлення звіту виконання практичного завдання у вигляді презентації 8

4.2 Оформлення звіту виконання практичного завдання у вигляді текстового документу 13

4.3 Оформлення звіту виконання практичного завдання в репозиторії github 16

**5 ПЕРЕЛІК ІНТЕРАКТИВНИХ РЕСУРСІВ ДЛЯ САМОСТІЙНОГО ВИВЧЕННЯ 18**

5.1 Перелік ресурсів для вивчення git 18

5.2 Перелік інтерактивних ресурсів для вивчення роботи shell, bash, linux 21

# 

# 

# 

# 

# 

# 

# 

# 

# **1 ЗАГАЛЬНІ ВІДОМОСТІ**

Практичні завдання є необхідною частиною підготовки майбутнього технічного спеціаліста, тому є невід’ємною складовою навчального процесу.

Виконання запропонованих завдань дозволить більш поглиблено засвоїти як теоретичні аспекти роботи з операційними системами сімейства GNU/Linux, так і отримати досвід по їх налаштуванню, конфігурації їх окремих компонентів та подальшому їх обслуговуванню в контексті програмування, системного адміністрування та ін.

Запропоновані завдання можуть виконуватись як одноосібно студентом так і у групах (2-3 студенти) колективно (за роботу в групі +1 бал). Якщо завдання виконувалось колективно, то необхідно обов’язково визначити які ролі в групі виконував кожен зі студентів (ці ролі у кожному практичному завданні мають змінюватись):

● Technical support (технічний спеціаліст, виконавець практичного завдання);

● Research and Content support (спеціаліст з пошуку інформації / аналітик, майстер презентації / контент-менеджер);

● Reporter / Reviewer (доповідач).

# **2 РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ**

Захист робіт буде відбуватись перед викладачем у вигляді короткої доповіді про виконану роботу (так званий pitching). За кожне з виконаних завдань нараховуються бали всім учасникам групи (за використання англійської мови при оформленні презентації +1 бал). Для кожного наступного завдання в колективі студенті змінюють свої ролі.

1. Продемонструвати працездатне рішення (будь-який з зазначених нижче варіантів):

● Принести ноутбук з демонстрацією кінцевого результату.

● Через технології віддаленого доступу організувати сеанс роботи з вашого телефону на ваш ПК, щоб продемонструвати виконане завдання.

● ***Зробити відео огляд виконаної роботи***.

● ***Зробити детальні скріншоти, які потім будуть додані у презентацію***.

2. Зробити презентацію виконаної роботи:

● Оформити звіт з усіма виконаними етапами (презентація, особистий блог, google sites, github та інший зручний варіант);

● Презентація має містити викладення таких питань – що треба було вивчити для реалізації завдання (нові терміни, загальна характеристика завдання); які програмні засоби для цього використані; які поетапні налаштування (команди, програми тощо) здійснено. Якщо виникли проблеми, то вказати які шляхи використані для їх усунення.

Захист робіт буде відбуватись перед викладачем у вигляді короткої доповіді про виконану роботу (так званий pitching). За кожне з виконаних завдань нараховуються бали всім учасникам групи (за використання англійської мови при оформленні презентації +1 бал). Для кожного наступного завдання в колективі студенті змінюють свої ролі.

# 

**Критерії оцінювання**

| **Якість виконання** | **Бали за роботу** |
| --- | --- |
| Завдання виконано повністю, представлена презентація проєкту з захистом перед своєю групою | **7** |
| Завдання виконано повністю (практична реалізація присутня) | **5** |
| Завдання виконано на базовому рівні (теоретичний огляд без практичної реалізації) | **3** |

**Примітка:** за виконання робіт в командах та оформлення звітів з використанням системи контролю версій (git) та англійської мови може бути нараховано **додатковий 1 бал.**

# 

# **3 ТЕМАТИКА ЗАВДАНЬ САМОСТІЙНОЇ РОБОТИ**

## **3.1 Практичне завдання “Work-case 1”**

1. Опишіть для чого використовують git, які основні дії та команди в ньому виконують.  
 Git – це система контролю версій. Використовують для:

• збереження історії змін у файлах;

• організації командної роботи над проєктами;

• відстеження, хто і коли вніс зміни;

• повернення до попередніх версій файлів.

**Основні дії та команди:**

• git init – створити новий репозиторій

• git clone URL – клонувати існуючий репозиторій

• git status – перевірити статус файлів

• git add . – додати всі змінені файли у staging area

• git commit -m "опис змін" – зробити комміт

• git log – переглянути історію коммітів

• git branch – переглянути/створити гілки

• git checkout / git switch – перемикання між гілками

• git merge – злиття гілок

• git push – відправка змін на сервер (GitHub тощо)

• git pull – завантаження нових змін із сервера

2. Що таке "комміт", як він дозволяє відслідковувати зміни у файлах?  
 Комміт (commit) – це "знімок" стану файлів у певний момент.

Має унікальний ідентифікатор (хеш) і опис змін.

Дозволяє:

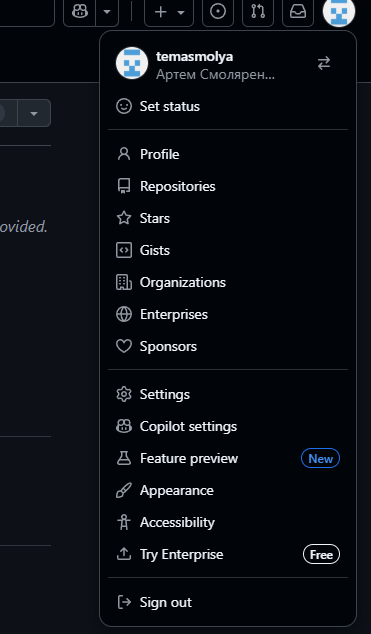
• відстежувати зміни у файлах;

• бачити, хто і що змінював;

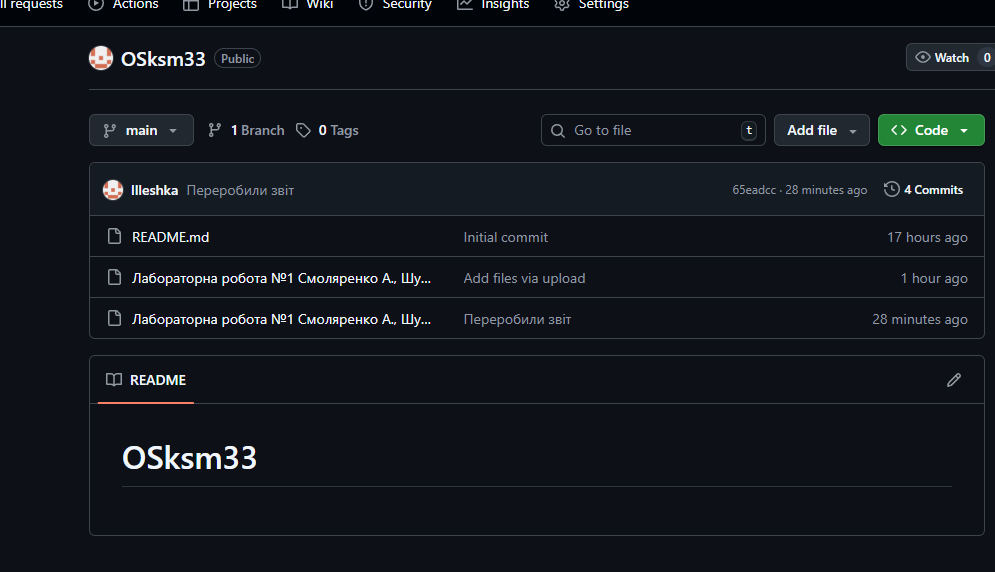
• повертатися до попередніх версій;

• формувати історію проєкту крок за кроком.

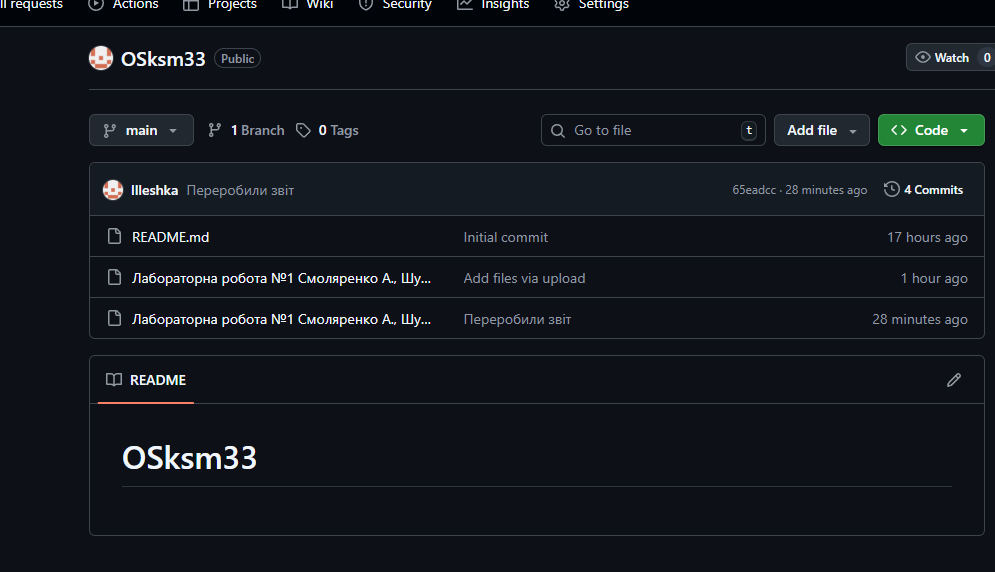
3. Зареєструйте власний git-аккаунт (gitlab, github або інша платформа).



4. Створіть новий публічний репозиторій, який будете використовувати для додавання всіх виконаних робіт з дисципліни «Операційні системи» (якщо працюєте в команді долучіть інших учасників команди до його редакторів).



5. Розмістіть свій перший звіт про виконаний Work-case 1 (презентацію, текстовий файл, html-сторінку) у даному репозиторії.



# 

# 

# 

# 

# 

# **ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Онлайн-курс “Git для розподіленої розробки програмного забезпечення” [Електронний ресурс] - Доступ до ресурсу:

<https://courses.prometheus.org.ua/courses/course-v1:LinuxFoundation+GIT101+2023_T1/about?gclid=Cj0KCQiA3eGfBhCeARIsACpJNU-f-EbxRZj-JZ-GlwxXgg3ygYd4gK6aRZy13TDnpxO_neOQ11-zD1IaAqcsEALw_wcB>

2. Git Підручник [Електронний ресурс] - Доступ до ресурсу:

<https://w3schoolsua.github.io/git/index.html>

3. Pro Git [Електронний ресурс] - Доступ до ресурсу:<https://git-scm.com/book/uk/v2>

4. Learn Git Branching [Електронний ресурс] - Доступ до ресурсу:<https://learngitbranching.js.org/?locale=uk>

5. Git How To: курс навчання Git українською [Електронний ресурс] - Доступ до ресурсу:<https://githowto.com/uk>

6. Shell [Електронний ресурс] - Доступ до ресурсу:<https://www.learnshell.org/>

7. Unix / Linux for Beginners [Електронний ресурс] - Доступ до ресурсу:<https://www.tutorialspoint.com/unix/index.htm>

8. Linux Admin Tutorial [Електронний ресурс] - Доступ до ресурсу:<https://www.tutorialspoint.com/linux_admin/index.htm>

9. Ubuntu Tutorial [Електронний ресурс] - Доступ до ресурсу:<https://www.tutorialspoint.com/ubuntu/index.htm>

10. Linux Survival [Електронний ресурс] - Доступ до ресурсу:<https://linuxsurvival.com/>