# Лабораторна робота №2

**Тема:** Основи роботи з системою керування версіями програмного забезпечення. Реєстрація на веб-сервісі для хостингу проектів GitHub. Первинні налаштування Git. Ініціалізація каталогу. Стан файлів в Git. Перший коміт.

**Мета:** ознайомитись з загальними підходами і принципами роботи СКВ Git, отримати практичні навики застосування основних команд Git. Навчитись створювати репозиторії (сховища).

# Хід роботи

1. Вивчити теоретичний матеріал до лабораторної роботи.
2. Створити папку за робочим комп՚ ютером у розділі **Документи** зі своїм прізвищем (латинською мовою) – до цієї папки надалі зберігатимете усі виконані лабораторні роботи (замість пропуска використовуйте символ нижнього підкреслення — “\_”).
3. Зареєструватися на **GitHub** (https://github.com/ => Sign up).
4. Створити папку **LR\_1**, у ній текстовий файл із назвою «My\_first\_repository» та вмістом «Мій перший репозиторій».
5. Зайти до свого облікового запису **GitHub** (https://github.com/ => Sign in).
6. Перейти до своїх репозиторіїв (меню користувача => Your profile => Repositories або меню користувача => Your repositories).
7. Створити новий репозиторій (кнопка “New”).
8. Дати ім’я вашому репозиторію – **Repository name** – LR\_1 (ім’я повинне бути унікальним у ваших репозиторіях; під час вибору імені будьте уважними, оскільки змінювати його крайнє не бажано), опис – **Description**

– «Мій перший репозиторій», задайте публічний тип доступу (тобто для всіх користувачів можливість перегляду та завантаження) – **Public**; завершіть створення репозиторію – **Create repository**.

1. У вкладці **Code** міститься вся необхідна інформація для створення першого репозиторію. Скопіюйте ці команди до текстового файлу

«My\_first\_repository».

1. Відкрийте консольне вікно (термінал) (Пуск => Усі програми => Службові => **Командний рядок** або в пошуку введіть **cmd**), перейдіть до папки LR\_1 (команда cd назва\_папки). Для коректнішого та швидшого переходу перегляньте властивості папки – ПКМ => Властивості => Загальні => Розташування.
2. Виконайте наступні команди (їх ви також можете побачити на сторінці створення нового репозиторію в групі **…or create a new repository on the command line** або у своєму текстовому файлі «My\_first\_repository»):
   1. **git init** (після ініціалізації можемо бачити папку .git, яка є прихованою системою за замовчуванням – її можна побачити виконавши – меню папки Вигляд => Показати приховані елементи)
   2. **echo** “# LR\_1” >> README.md — створюємо файл README.md — це стандартний файл, який ви можете бачити у будь- якому проекті; у цьому файлі зазвичай міститься основна інформація про проект, як запустити тощо (ця команда повинна бути у вашому файлі, у який ви скопіювали команди зі сторінки створення нового репозиторію).
   3. **git status** – перевіряємо статус файлів у репозиторії — файли зеленого кольору – додавання пройшло успішно – файли готові до коміту, червоного – файли не додано (якщо ця команда показує вам файли з назвою **~$\_**назва\_файла, то необхідно переконатись чи є у вас незакриті файли, якщо є, то закрити їх і повторити команду **git status**)
   4. **git add** назва\_файлу\_з\_розширенням — додаємо файли (для одного файла одна команда) до репозиторію (щоб додати всі файли разом можна використати команду **git add .**)
   5. **git commit -m** "first commit" — команда для збереження змін у репозиторії локально, тобто на комп’ютері (для того, щоб переконатися, що додали файли для коміту, наберіть ще раз команду **git status** – файли зеленого кольору – додавання пройшло успішно – файли готові до коміту, червоного – файли не додано (отже повторіть пункт 11.4))
   6. Оскільки це ваш перший коміт, необхідно авторизуватись – ввести логін та ім’я користувача Git використовуючи команди:

**git config --global user.email "**[you@example.com](mailto:you@example.com)**" git config --global user.name "**Your Name**"**

(якщо ви використовуєте власний комп’ютер, то це можна не робити кожного разу під час створення нового репозиторію, адже дані там зберігаються; якщо ви працюєте у комп’ютерній лабораторії – необхідно щоразу перевіряти налаштування командою **git config -- list**)

* 1. Повторіть команду **git commit -m** "first commit" (якщо 11.5 була неуспішною)
  2. **git remote add origin** посилання\_на\_ваш\_репозиторій — підключається до вашого репозиторію в **GitHub** (скопіюйте з GitHub щоб не помилитись)
  3. **git push -u origin master** – відправляє дані з вашого локального репозиторію (на комп’ютері у папці **.git**) на віддалений (тобто в **GitHub**)

1. Перезавантажте сторінку **GitHub** з вашим репозиторієм. Переконайтесь у тому, що зміни збереглись. Якщо не всі файли додано чи зміни не збереглись, **повторіть кроки з 11**.
2. Відредагуйте файл **README.md** безпосередньо в **GitHub**. Введіть до нього текст «Мій перший репозиторій» та збережіть зміни (так ми можемо редагувати файли безпосередньо в **GitHub** не роблячи змін на комп’ютері).
3. Дайте відповідь на контрольні питання та здайте роботу викладачу.

# Контрольні питання:

1. Що таке Git?
2. В якому статусі можуть перебувати файли при роботі з Git?
3. Для чого використовується команда add?
4. Для чого використовується команда status?
5. Для чого використовується команда commit?
6. Для чого використовується команда init?
7. Що необхідно писати в коментарях комітів?
8. Як переглянути список комітів?
9. Для чого використовується команда checkout?

# Критерії оцінювання

|  |  |  |
| --- | --- | --- |
| **№ п/п** | **Види робіт. Критерії оцінювання знань студентів** | **Максимальна к-сть балів** |
| 1 | Робота виконана у зазначений термін, у повному  обсязі, без помилок і зарахована. | 5 |
| 2 | Робота виконана у зазначений термін, у повному  обсязі, зарахована, але є помилки. | 4 |
| 3 | Робота виконана у неповному обсязі, або з  порушенням терміну виконання, або при наявності значних помилок. | 3 |
| 4 | Виконання пропущеної роботи або повторне  виконання не зарахованої роботи. | 2 |
| 5 | Робота не виконана або не зарахована. | 0 |