



ЗВІТ

про виконання лабораторної роботи №1
з дисципліни: “Автоматизоване проектування комп’ютерних систем”
на тему: «Ініціалізація GIT репозиторію»

Варіант - 17

Виконав:

ст. гр. KI-405

Легкобит Н. В.

Перевірів:

Шпіцер А. С.

Завдання:

1. Створити репозиторій на GitHub з назвою «csad<YY₁YY₂><група><ім'я студента><номер варіанту >», основною гілкою має бути 'develop'.
YY₁ – рік початку навчання;
YY₂ – рік закінчення навчання.
Наприклад: 2019-2020 -> 1920;
Повний приклад назви: csad1920ki47spitzeras03.
2. Додати доступ для автора.
3. Створити гілку feature/develop/<номер завдання>. Наприклад, feature/develop/task1.
4. Створити файл README, де вказати:
 - a. Деталі про репозиторій
 - b. Деталі завдання
 - c. Порядковий номер студента та інформацію про варіант завдання
 - d. Додати інформацію про технології, мову програмування та обладнання, яке буде використовуватися в наступних завданнях.
5. Створити GIT TAG: <НАЗВА ПРОЄКТУ>_<ВЕРСІЯ>_WW<YYWWD>, де
WW – робочий тиждень,
D – номер дня тижня.
6. Створити pull request з назвою 'task1' та надати викладача як рецензента.
7. Після затвердження рецензента – виконати злиття запиту в гілку 'develop'.

Теоретичний матеріал

Git — це розподілена система керування версіями, яка дозволяє відстежувати зміни у файлах і координувати роботу кількох розробників над одним проектом. Вона створена для управління вихідним кодом програмних проєктів, але може використовуватися і для будь-яких інших типів файлів. Основна ідея Git полягає в тому, що всі зміни зберігаються

в комітах, і ці зміни можна повернути або аналізувати. Відмінною рисою Git є можливість локально зберігати повну історію змін кожного файлу, а також легко співпрацювати з іншими розробниками без постійного доступу до мережі.

GitHub — це веб-сервіс для хостингу Git-репозиторіїв, який додає до Git додаткові можливості для зручної співпраці, інтеграції та зберігання коду онлайн. Він дозволяє зберігати код у віддаленому репозиторії, надавати доступ до нього іншим користувачам та використовувати додаткові інструменти для контролю версій, наприклад, через pull requests. GitHub також пропонує можливості для управління проектами, моніторингу помилок і автоматизації тестування через CI/CD.

Гілка (Branch) — це незалежна версія проекту, в якій можна вносити зміни без впливу на інші частини проекту. Основна гілка, яка зазвичай використовується для стабільного коду, називається **develop** або **main**. Від гілок часто створюють підгілки для роботи над новими функціями чи виправленням помилок.

Репозиторій (Repository) — це сховище, де зберігаються всі файли проекту та вся історія їх змін. Він може бути локальним (на вашому комп'ютері) або віддаленим (на GitHub). Репозиторій містить всі коміти, гілки та теги проекту.

Тег (Tag) — це спеціальна мітка, що використовується для позначення певного моменту в історії проекту. Теги зазвичай використовуються для випусків нових версій програмного забезпечення. Вони допомагають легко знайти стабільні версії проекту, наприклад, випуски або релізи.

Хід роботи:

1. Використовуючи GitHub створив репозиторій, де гілкою за замовчуванням обрав 'develop'.
2. Створив нову гілку 'feature/develop/task1'. В файл README.md додав інформацію про проект, завдання, мову та технології виконання

поставленого в методичних вказівках завдання. Вміст файлу README.md наведений в Додатку А.

3. Створив коміт з поясненням змін які відбулися в гілці.
4. Створив тег, назвавши його згідно методичних вказівок, а саме 'csad2425KI405LehkobytNazarii11_1.0_WW24137'.
5. Після цього я додав викладача до репозиторію.
6. Створив pull request, де викладача вказав як 'reviewer-a'.
7. Отримав погодження на заливання змін та замержив гілку 'feature/develop/tsak1' у 'develop'.

Висновок: створив GIT репозиторій, в якому в подальшому буду виконувати наступні завдання курсу. Ознайомився та пропрактикував стандартний алгоритм роботи над подальшими завданнями з цього курсу.

Список використаної літератури

1. Основна документація Git – <https://git-scm.com/doc>
2. Офіційна документація GitHub – <https://docs.github.com>
3. Офіційна документація Arduino – <https://www.arduino.cc/reference/en>
4. Arduino Programming with C/C++ Tutorial (Learn C/C++ with Arduino) – <https://roboticsbackend.com/learn-arduino-c-cpp/>

Додаток А. Вміст файлу README.md

a. details about repo

This repository was created to host the laboratory works of Lehkobyt Nazarii, a student of KI-405 group, from the CSAD course. In the future, it will contain ready-made laboratory works from the initial stages to the work program, which will be the end of the course.

b. task details

The goal of the project is to develop a digital version of the classic and well-known Tic-Tac-Toe (3x3) game. The result is an interactive and intuitive platform where two players can compete by placing their symbols (X or O) on a 3x3 grid.

c. student number and details

Since I am the 11th student in the group, my assignment will look like this:

Student number	Student	Game	Config format
----------------	---------	------	---------------

-----	-----	-----	-----
-------	-------	-------	-------

d. details about technology, program language and HW

In this project, I am going to use a client-server architecture. The server, which will provide data to the client program, will be represented by an Arduino board (I will announce the specific model later). The Arduino board will perform calculations and return the results to the client program at the end. The language for the project development will be Java or C++/C# (no final decision has been made at this time). Note that there may be changes during project development.