МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

факультет програмної інженерії та бізнесу

кафедра інженерії програмного забезпечення

**Практична робота № 1**

з дисципліни « Об’єктно орієнтоване програмування »

*назва дисципліни*

на тему: « Використання Git і GitHub»

Виконав: студент 2 курсу групи № 622п

освітньої програми

121 інженерія програмного забезпечення

(шифр і назва ОП)

Зайченко Ярослав Ігорович

(прізвище й ініціали студента)

Прийняв: доц. Вдовітченко О.В.

(посада, науковий ступінь, прізвище й ініціали)

Кількість балів:

Харків – 2023

# ПОСТАНОВКА ЗАДАЧІ

Git

1. Установити систему контролю версій Git (якщо не встановлена!).
2. Створити локальний репозиторій під контролем Git.
3. Продемонструвати створення не менше трьох комітів у гілці master (main).
4. Вивести історію комітів.
5. Переключитися між комітами.
6. Створити додаткову гілку з не менше ніж двома комітами.
7. Об’єднати гілки і видалити додаткову гілку.

GitHub

1. Створити обліковий запис для GitHub (якщо він відсутній!).
2. Створити віддалений пустий репозиторій.
3. Підключитися до віддаленого репозиторію.
4. Відправити вміст локального репозиторію у віддалений репозиторій.
5. Продемонструвати клонування віддаленого репозиторію.
6. Продемонструвати витягування вмісту віддаленого репозиторію.

Visual Studio

1. На базі однієї з лабораторних робіт (бажано сьомої!):
2. додати створення локального і віддаленого репозиторію (файл Readme.md додаємо!);
3. продемонструвати створення комітів;
4. продемонструвати відправлення вмісту локального репозиторію у віддалений репозиторій;
5. оформити Readme.md файл\* (інформацію беремо із звіту обраної лабораторної роботи!).

ХІД РОБОТИ

Git

Для початку роботи з гітом необхідно завантажити програму з офіційного сайту <https://git-scm.com/downloads>. У роботі було використано Portable ("thumbdrive edition") версію програми.

Для створення локального репозиторію під контролем Git необіхідно відкрити гіт в директорії, де необхідна ініціалізація гіту, або використати команду **cd** зі шляхом необхідної папки та ввести команду **git init.** Результат на рисунку 1.

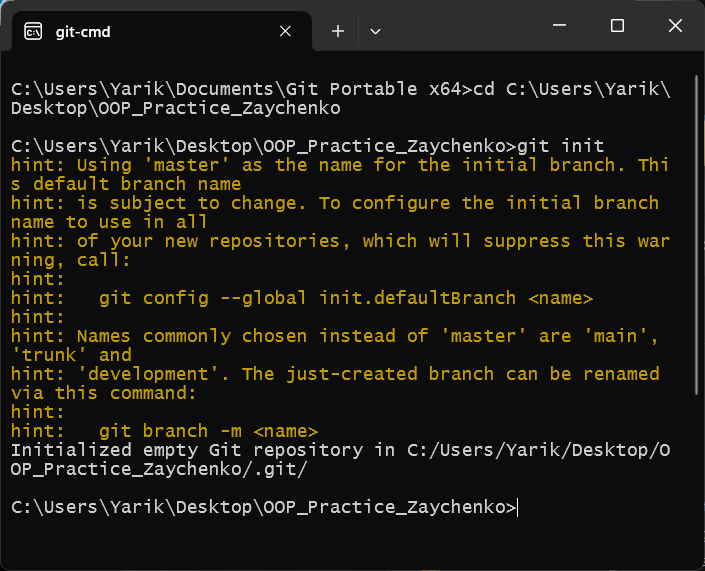


Рисунок 1 – Створення пустого локального репозиторію

Для створення комітів у гілці master (main) необхідно використати команду **git add .** для додавання всіх змін до індексу. Потім команда **git commit -m "коментар до коміту"** для фіксації змін. Приклад на рисунку 2.

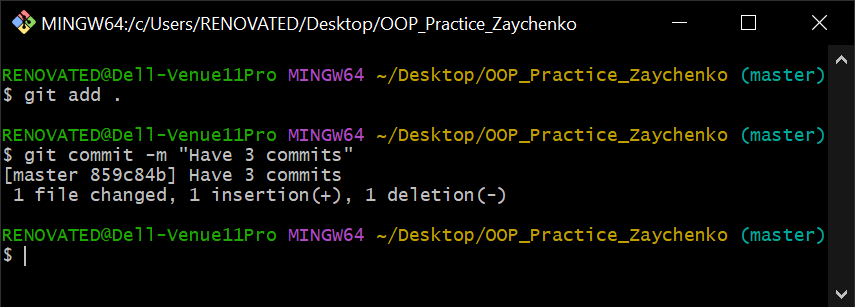


Рисунок 2 – Створення 4-го коміту з наявністю 3-х комітів у репозиторії

Для перегляду історії створених комітів використовують команду **git log**. Результат створених комітів у репозиторії зображено на рисунку 3.

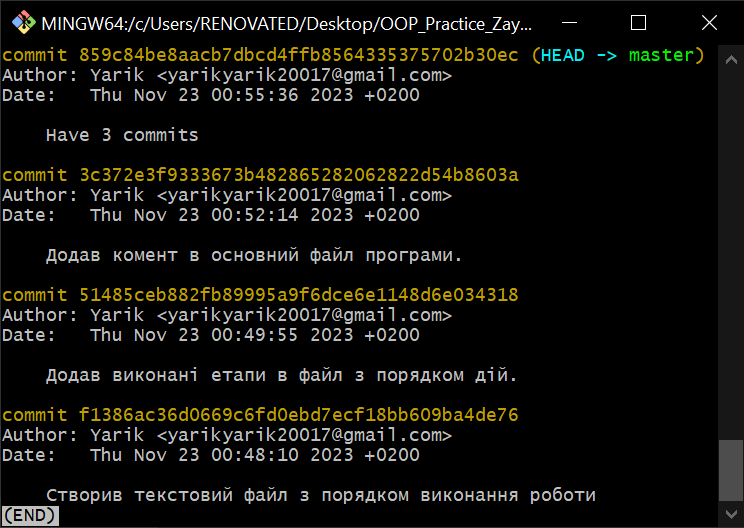


Рисунок 3 – Історія комітів локального репозиторію

Щоб переключитися між комітами, скористаймося командою **git checkout**, а також додати хеш коміту. Результат зображено на рисунку 4.

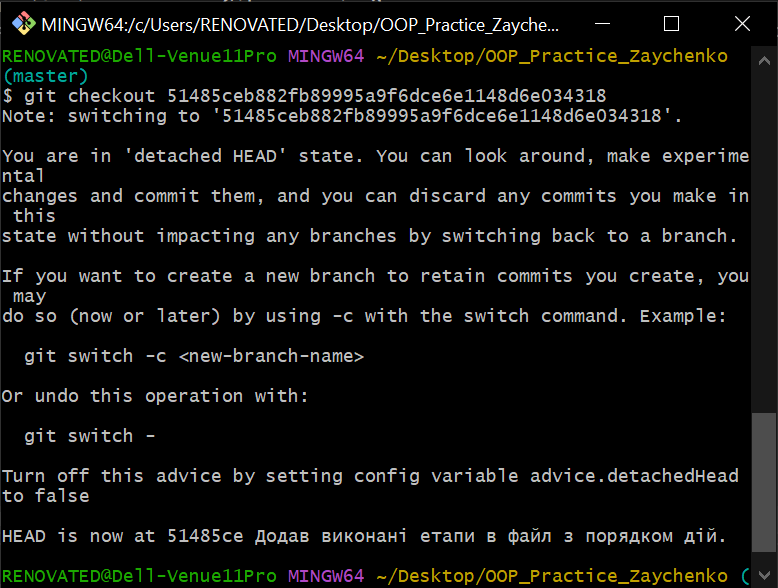


Рисунок 4 – Переключення на коміт з хешом 51485ce

Для створення нової гілки в локальному репозиторії використовують команду **git branch** для створення нової гілки та **git checkout** або **git switch** для переключення на нову гілку. Результат створення нової гілки з комітами на рисунках 5 та 6.

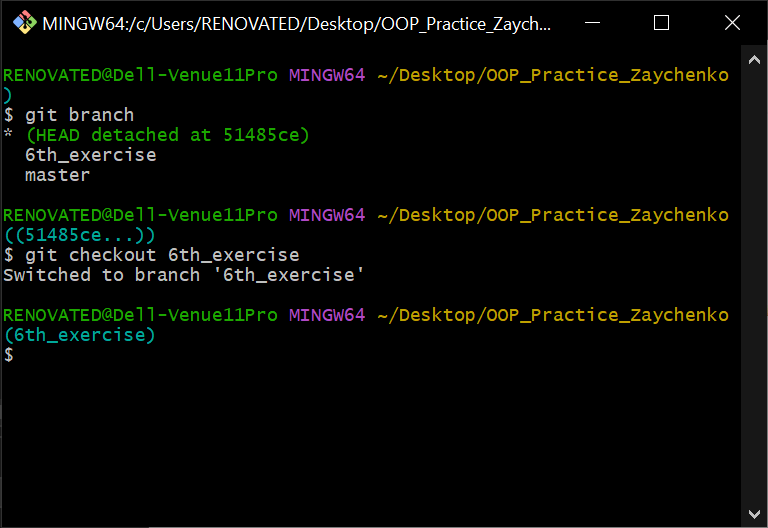


Рисунок 5 – Створення нової гілки 6th\_exercise та переключення на гілку

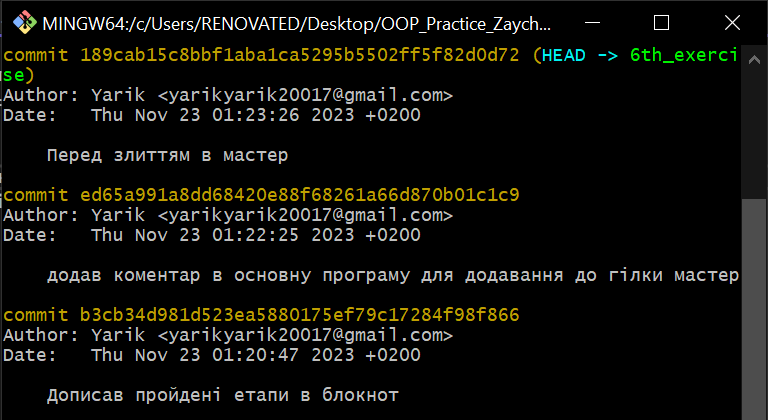


Рисунок 6 – Історія комітів на новій створеній гілці 6th\_exercise

Для об’єднання гілок необхідно переключитись на цільову гілку **git checkout master** та об'єднати гілки командою **git merge 6th\_exercise**.

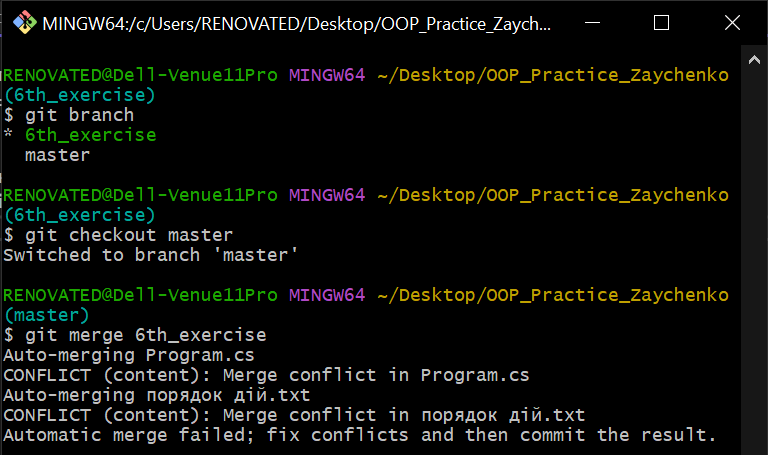


Рисунок 7 – Об’єднання гілок

Конфлікти були полагоджені в IDE MSVS2023 безпосередньо в файлах програми.

Після успішного об’єднання можна видалити вже не потрібну гілку 6th\_exercise командою **git branch -d 6th\_exercise**. Якщо гілка містить невикористані зміни, можна використати параметр -D замість -d: **git branch -D 6th\_exercise**, що примусово видаляє гілку. Результат на рисунку 8.

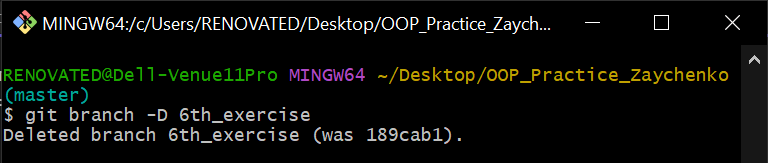


Рисунок 8 – Видалення гілки 6th\_exercise

GitHub

Створив обліковий запис на сервісі для зберігання віддалених репозиторіїв GitHub. <https://github.com/Yarik-Za/>.

Створив віддалений пустий репозиторій. Рисунки 9 та 10. Посилання на віддалений репозиторій: <https://github.com/Yarik-Za/OOP_Practice1_Zaychenko>

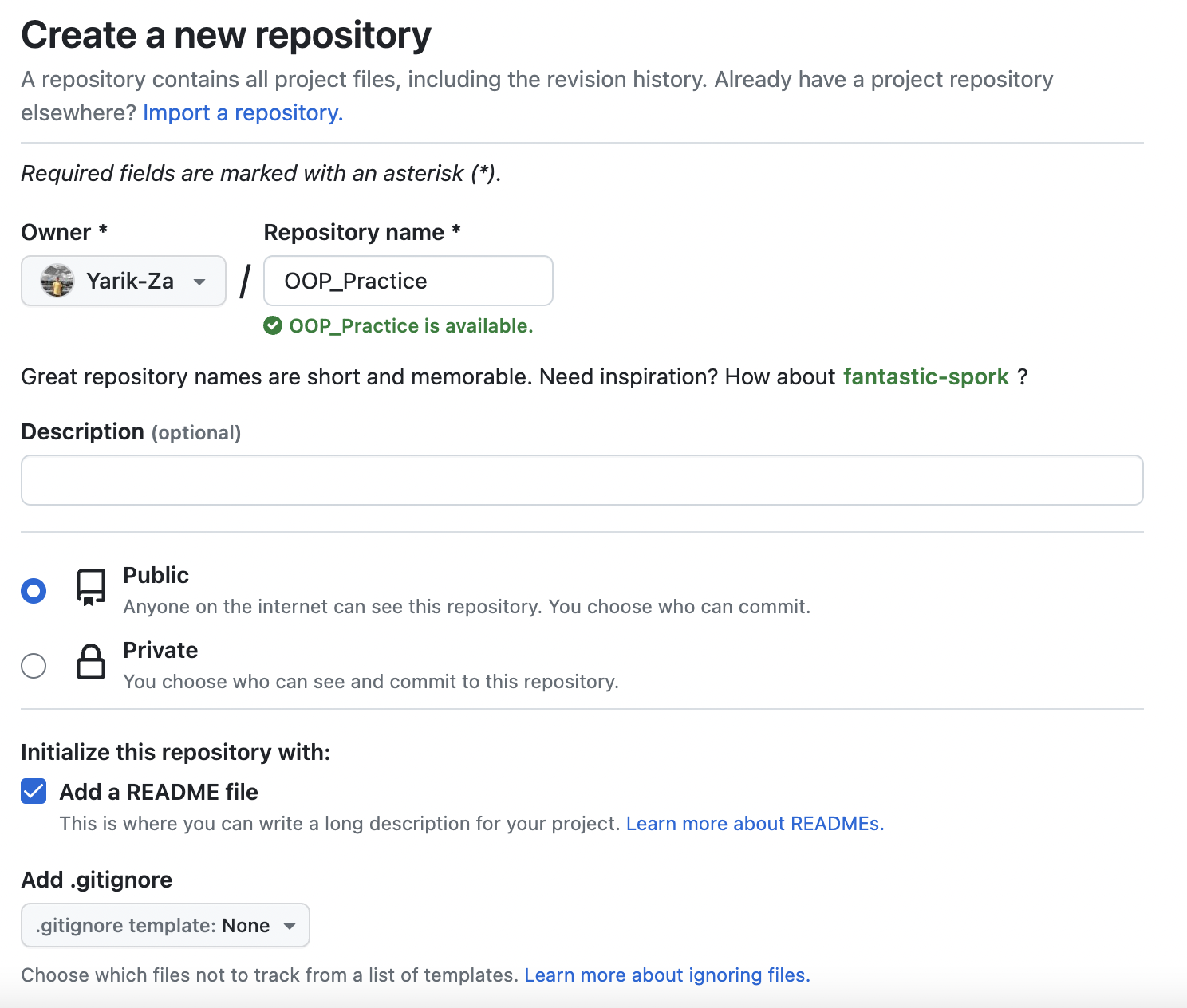


Рисунок 9 – Створення віддаленого репозиторію на сервісі GitHub

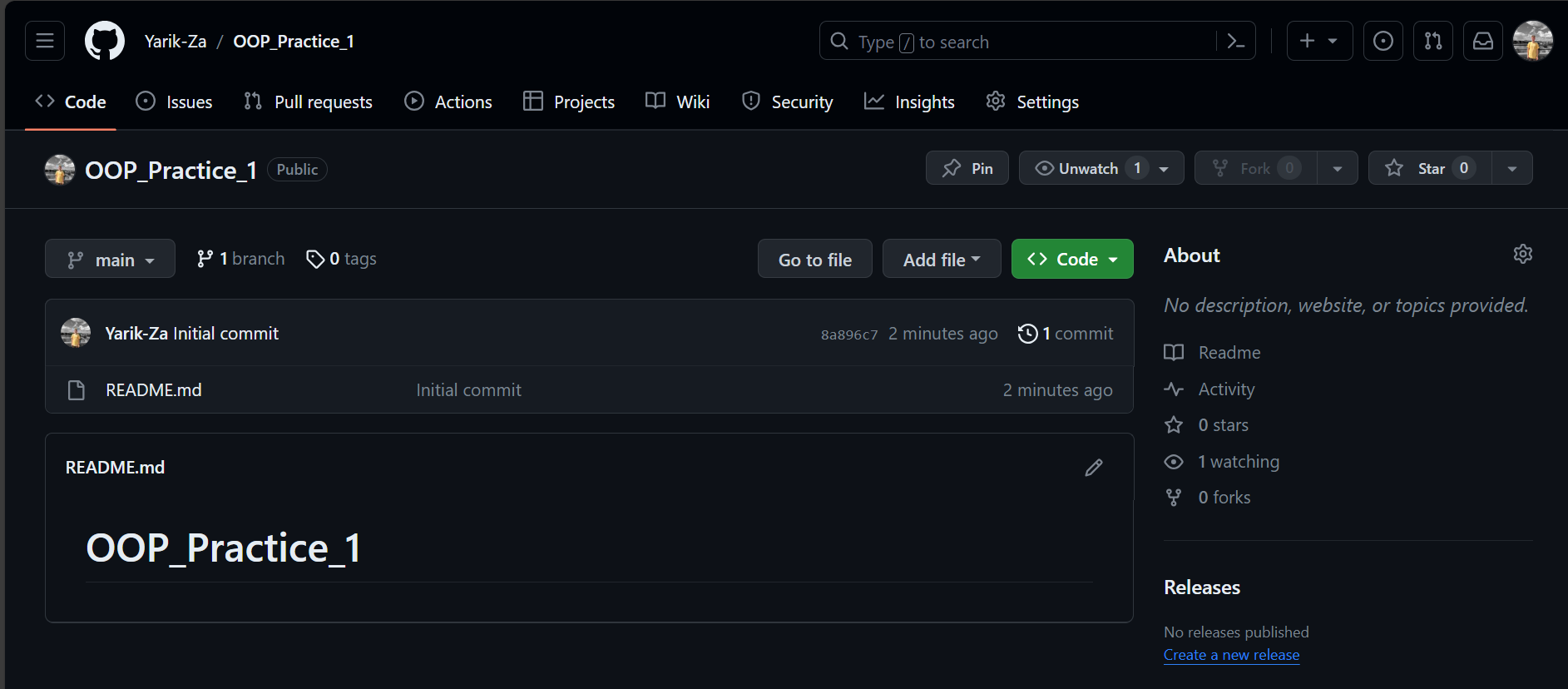


Рисунок 10 – Створений пустий віддалений репозиторій на сервісі GitHub

Для підключення до віддаленого репозиторію використовується команда **git remote add origin <GitRepo URL>**. Результат на рисунку 11.

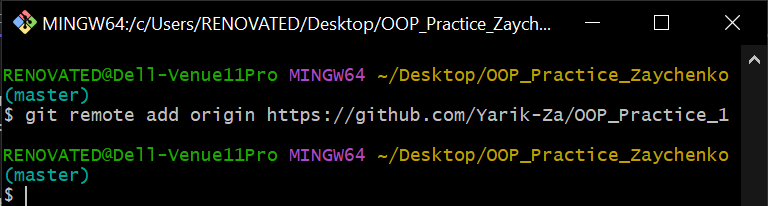


Рисунок 11 – Підключення віддаленого репозиторію до локального

Щоб відправити вміст локального репозиторію у віддалений репозиторій необхідно виконати команду **git push origin <назва гілки>**.Результат на рисунку 12.

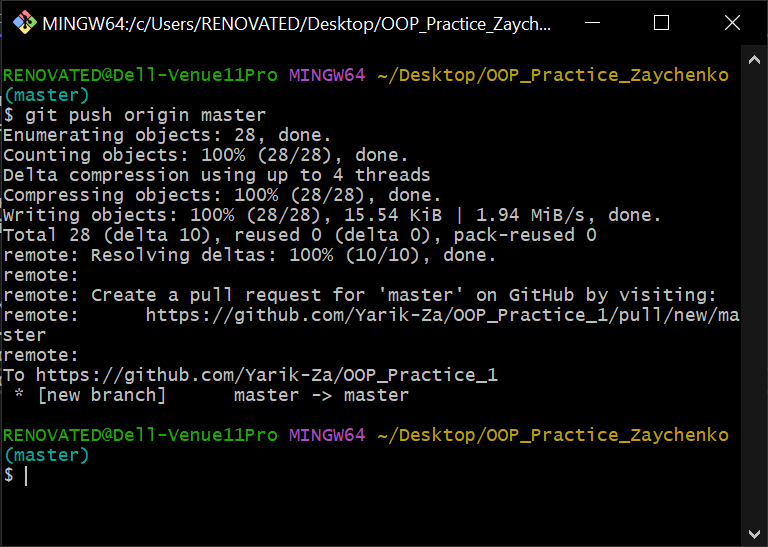


Рисунок 12 – Виконання синхронізації локального до віддаленого репозиторіїв

Для клонування віддаленого репозиторію використаю свій інший віддалений репозиторій, який скопіюю на іншу машину. Для цієї дії використовується команда **git clone <GitRepo URL>**. Результат клонування зображений на рисунку 13

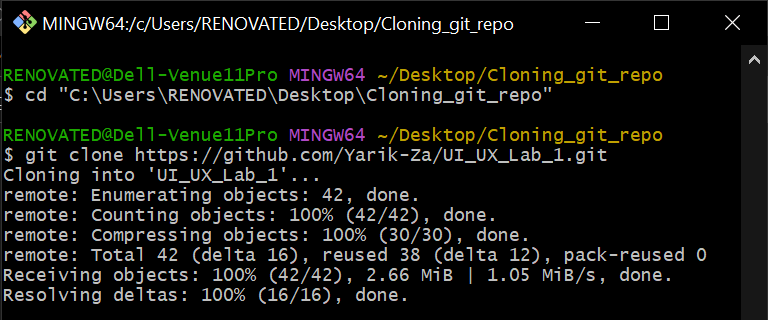


Рисунок 13 – Виконання клонування віддаленого репозиторію на машину

На сайті сервісу GitHub створив файл та коміт, щоб виконати витягування вмісту віддаленого репозиторію. Результат на рисунку 14.

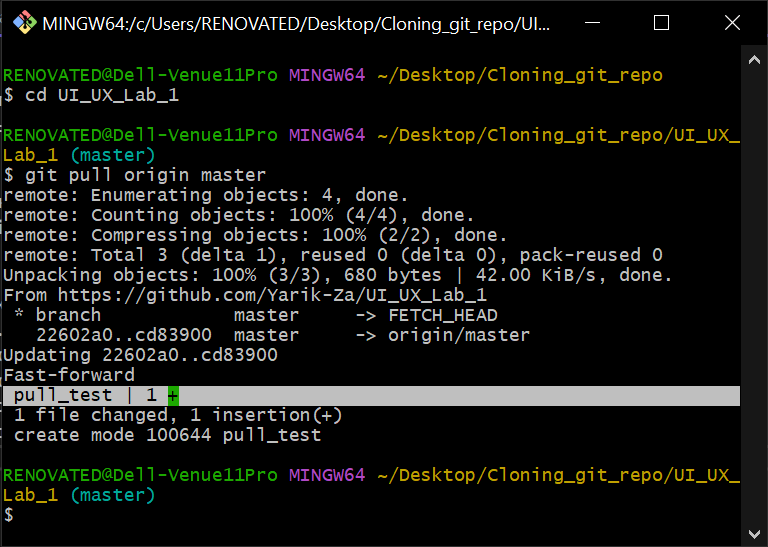


Рисунок 14 – Виконання витягування з віддаленого репозиторію

Робота з git у IDE Visual Studio 2023

Cтворення локального і віддаленого репозиторію через інтерфейс IDE Visual Studio 2023 подано на рисунках 15 та 16

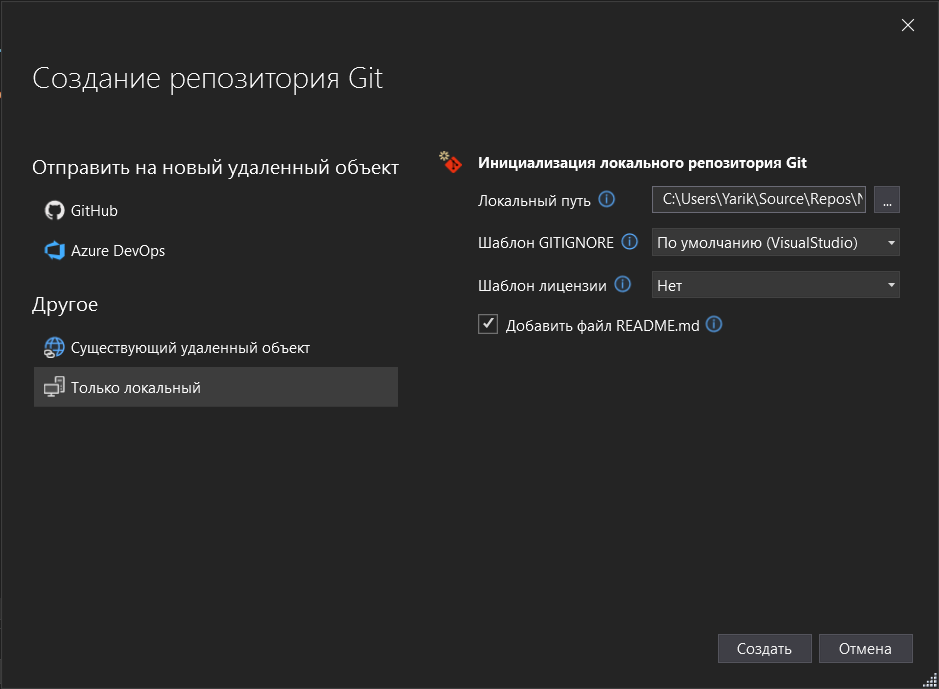


Рисунок 15 – Створення локального репозиторію через IDE Visual Studio 2023

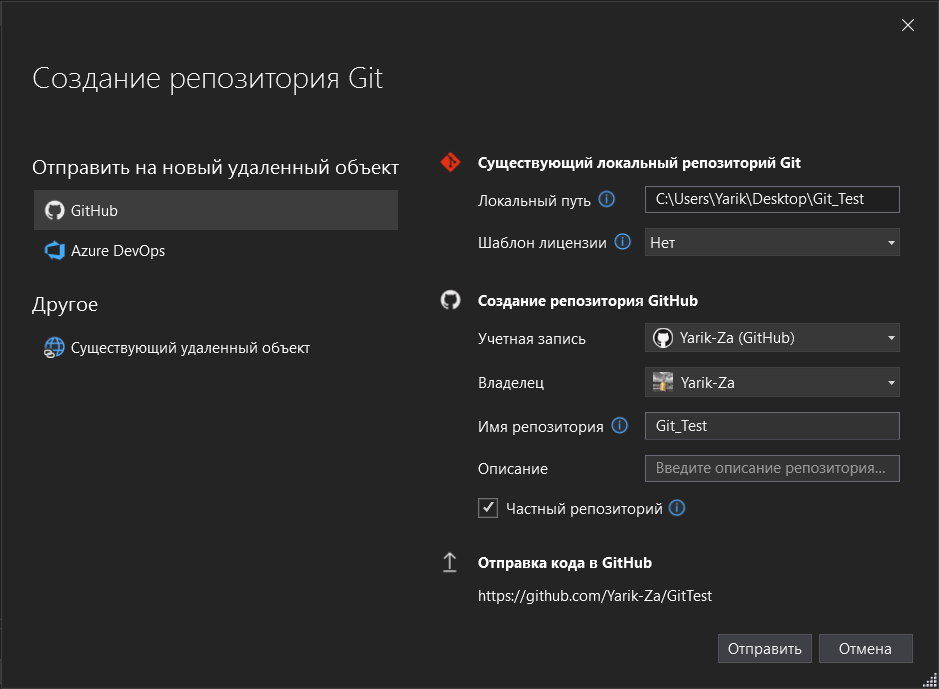


Рисунок 16 – Створення віддаленого репозиторію через IDE Visual Studio 2023

Для створення комітів в інтерфейсі середовища розробки існує спеціальне вікно, зображене на рисунку 17.

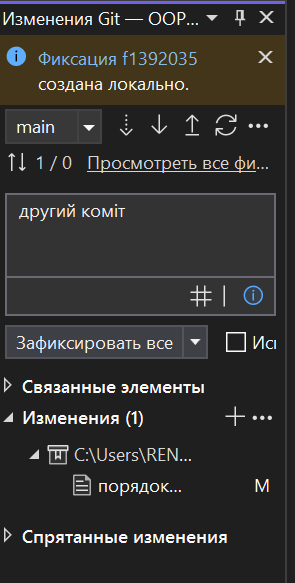


Рисунок 17 – Вікно для роботи з гіт в середовищі розробки

Подивитись історію комітів репозиторію теж можна в спеціальному вікні. Приклад на рисунку 18.

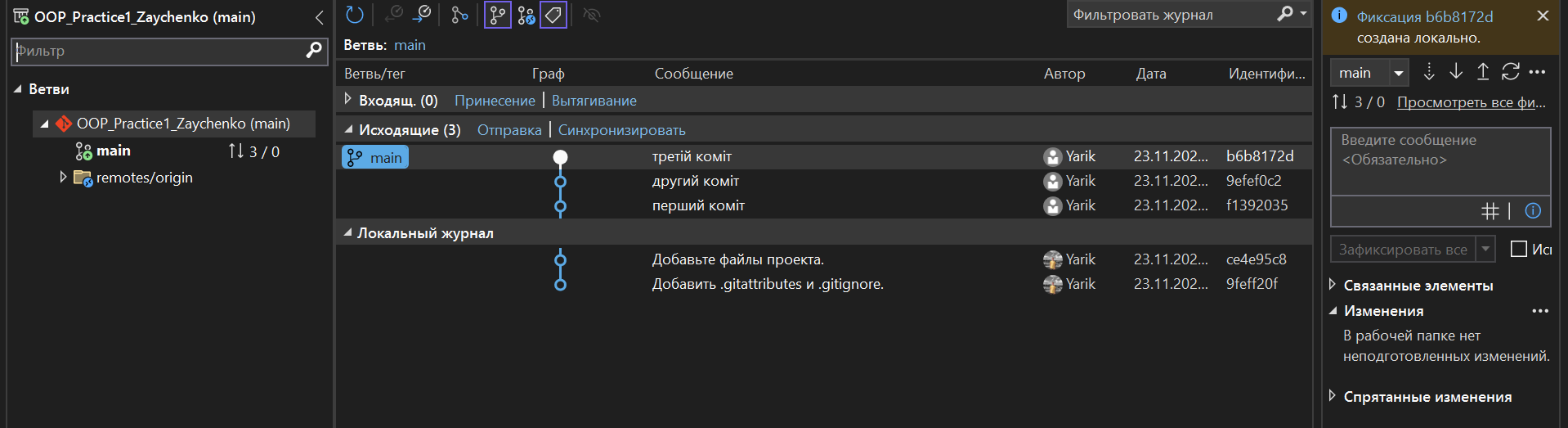


Рисунок 18 – Вікно для відслідковування створенних комітів в репозиторії

Відправлення вмісту локального репозиторію у віддалений репозиторій відбувається за допомогою спеціальних кнопок, виділені червоним обведенням. Зображено на рисунку 19.

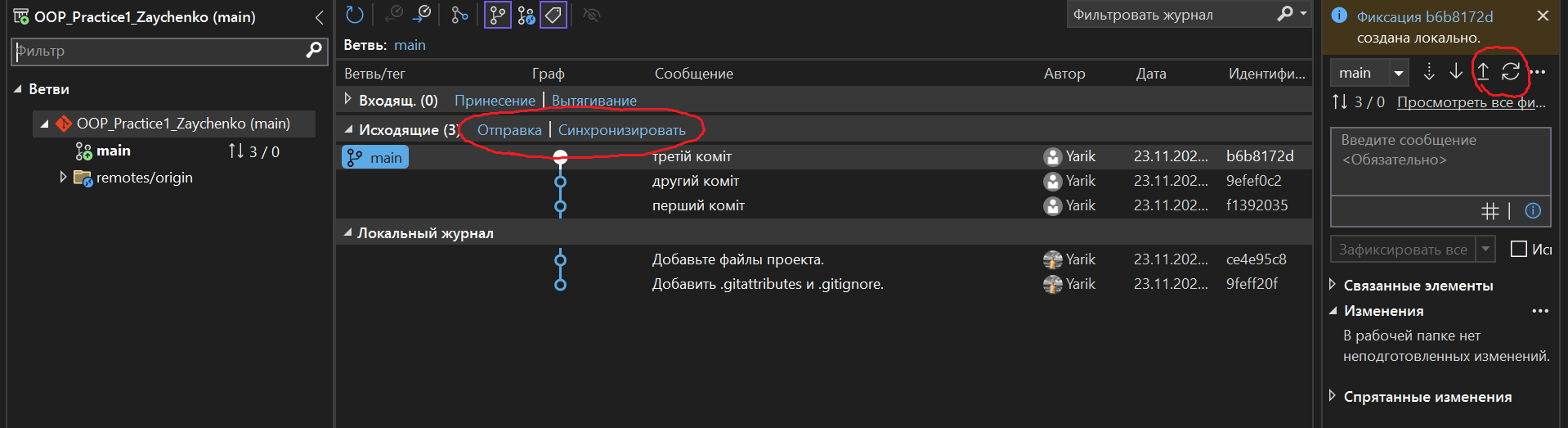


Рисунок 19 – Кнопки для синхронізації локальних змін з віддаленим репозиторієм

Після успішного відправлення змін на віддалений репозиторій коміти з категорії Вихідні мають перейти до категорії Локальний журнал. Результат на рисунку 20.

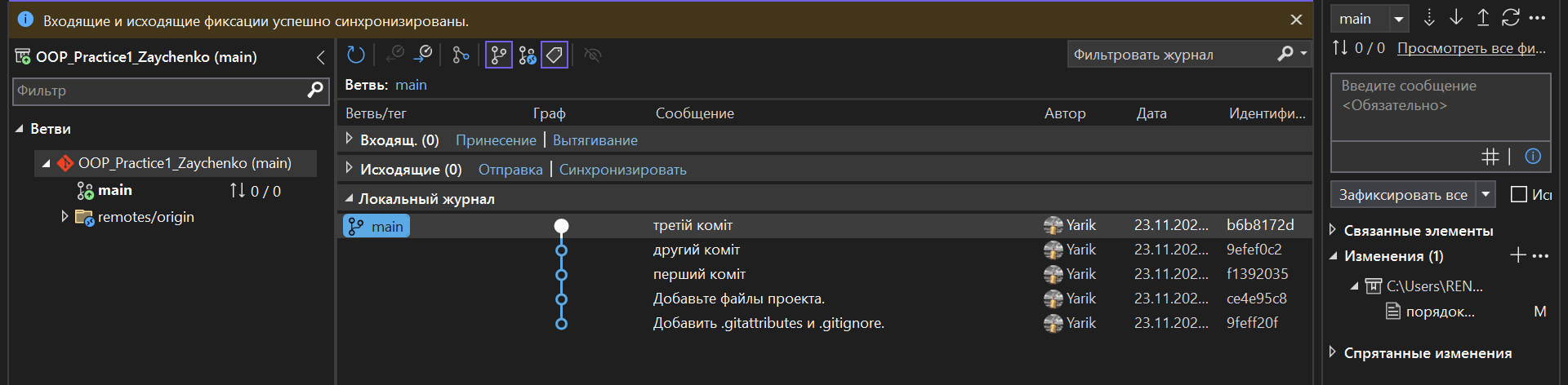


Рисунок 20 – успішне відправлення комітів на віддалений репозиторій

Оформлення файлу Readme.md було виконано за змістом звіту лабораторної роботи 7 з предмету Об’єктно-орієнтоване програмування.

Файл Readme.md з розміткою Markdown наведений у звіті цілком у Додатку А

ДОДАТОК А  
*Машинний лістинг файлу README.md з розміткою Markdown*

**README.md:**