***Лабораторна робота №1***

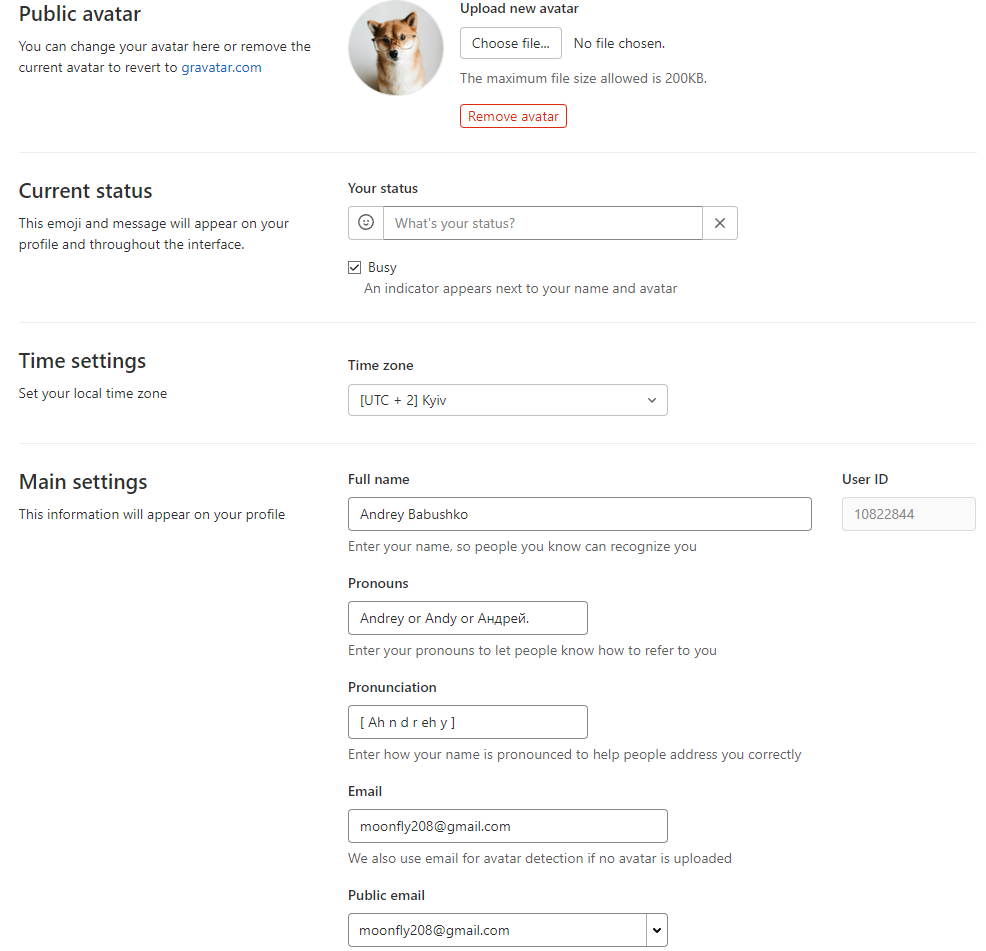
***Тема: Знайомство з системою контролю версій Git. Поняття репозиторію.***

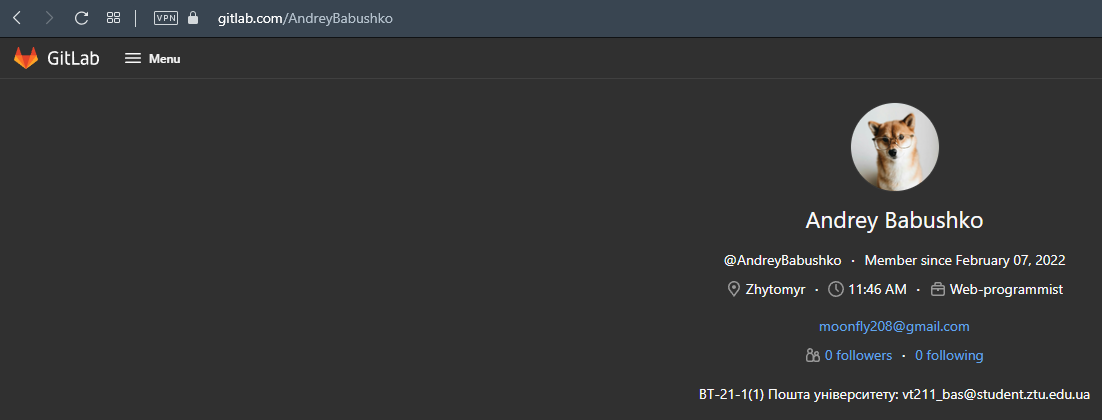
***Мета роботи***: ознайомитися системами керування версіями. Дослідити та отримати практичні навики щодо створення найпростішої програми та власного репозиторію.

***Хід роботи:***

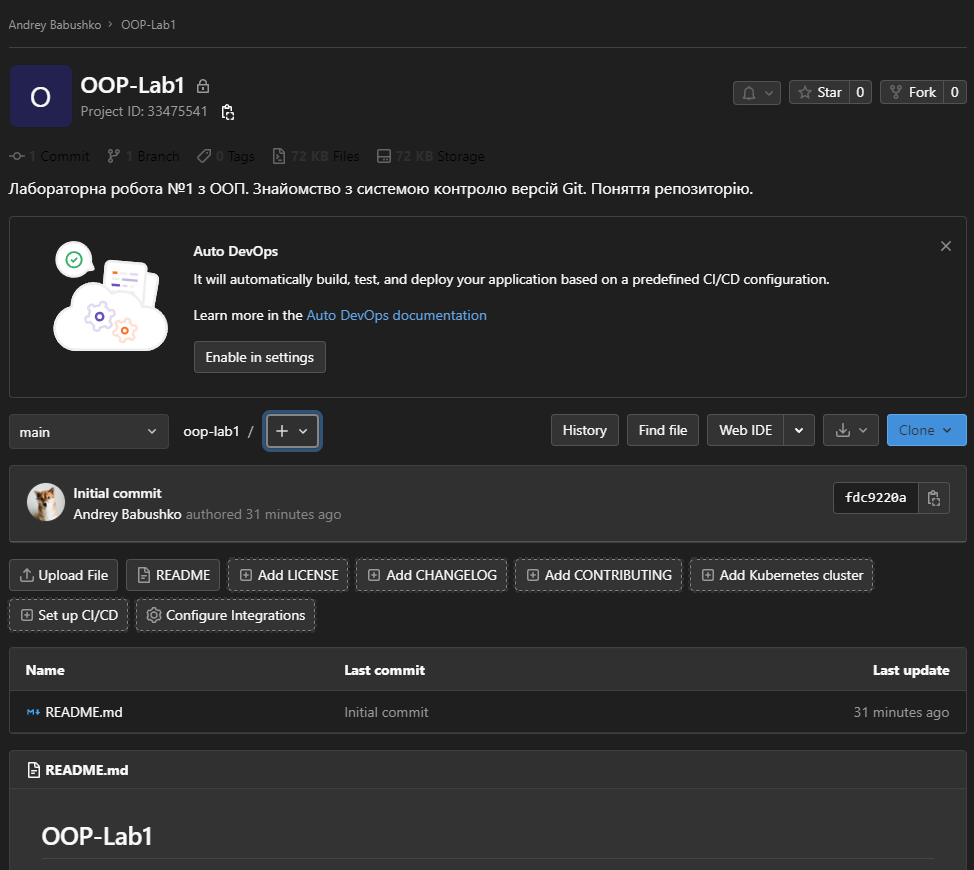
***Завдання на лабораторну роботу:***

1. ***Ознайомитись з теоретичними відомостями,*** ретельно опрацювати матеріал. Вміти давати пояснення термінам та поняттям: система керування версіями; централізовані та розподілені системи контролю версіями; репозиторій; приватні та відкриті репозиторії; GitHub; GitLab; BitBucket.
2. ***Зареєструватися на сайті GitLabта створити репозиторій:*** 
   1. Зареєструватися на сайті <https://gitlab.com/>

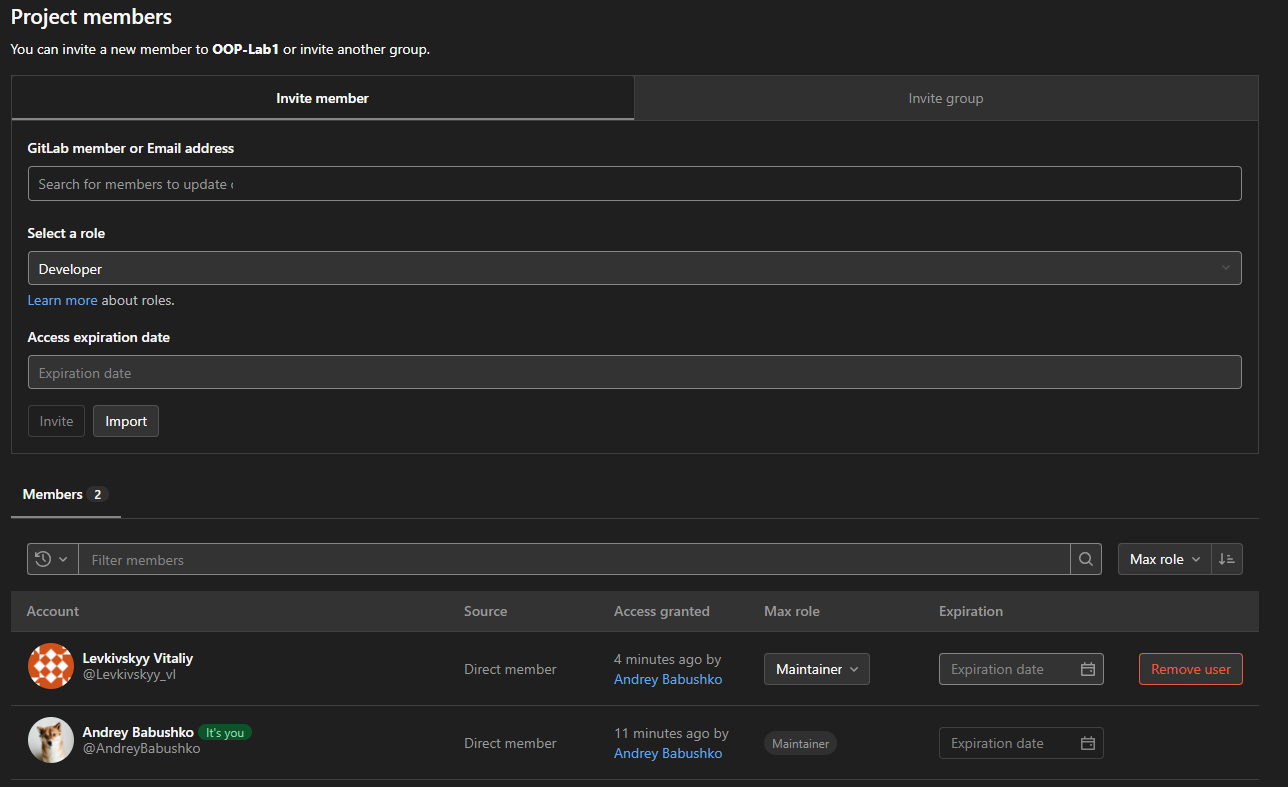




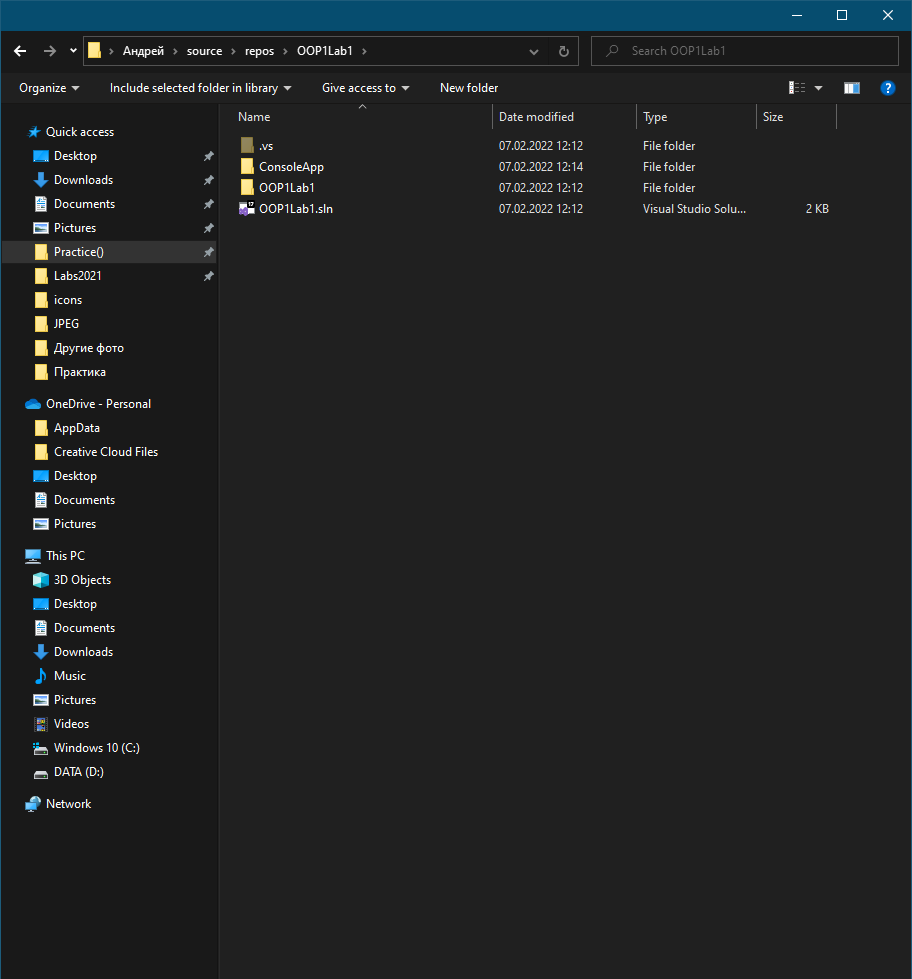
* 1. Увійти на власну пошту та підтвердити реєстрацію у листі, який надійшов з сервера GitLab. – Виконано.
  2. Повернутися на сайт GitLabі увійти під власним логіном та паролем. – Змінив пароль та перезайшов у GitLab.
  3. Створити перший репозиторій з назвою «OOP-Lab1»:



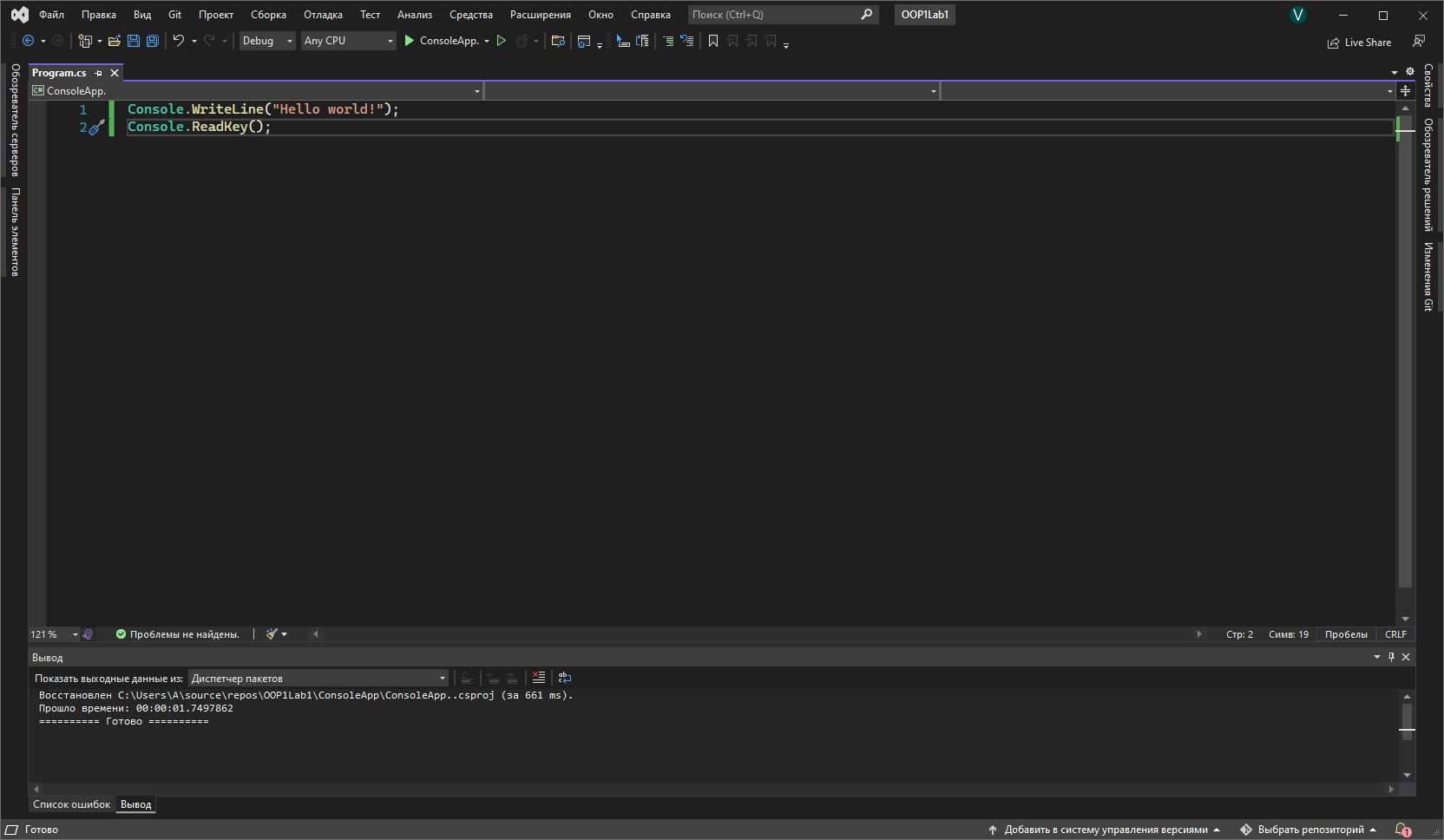
* 1. Надати доступ до репозиторію викладачам:



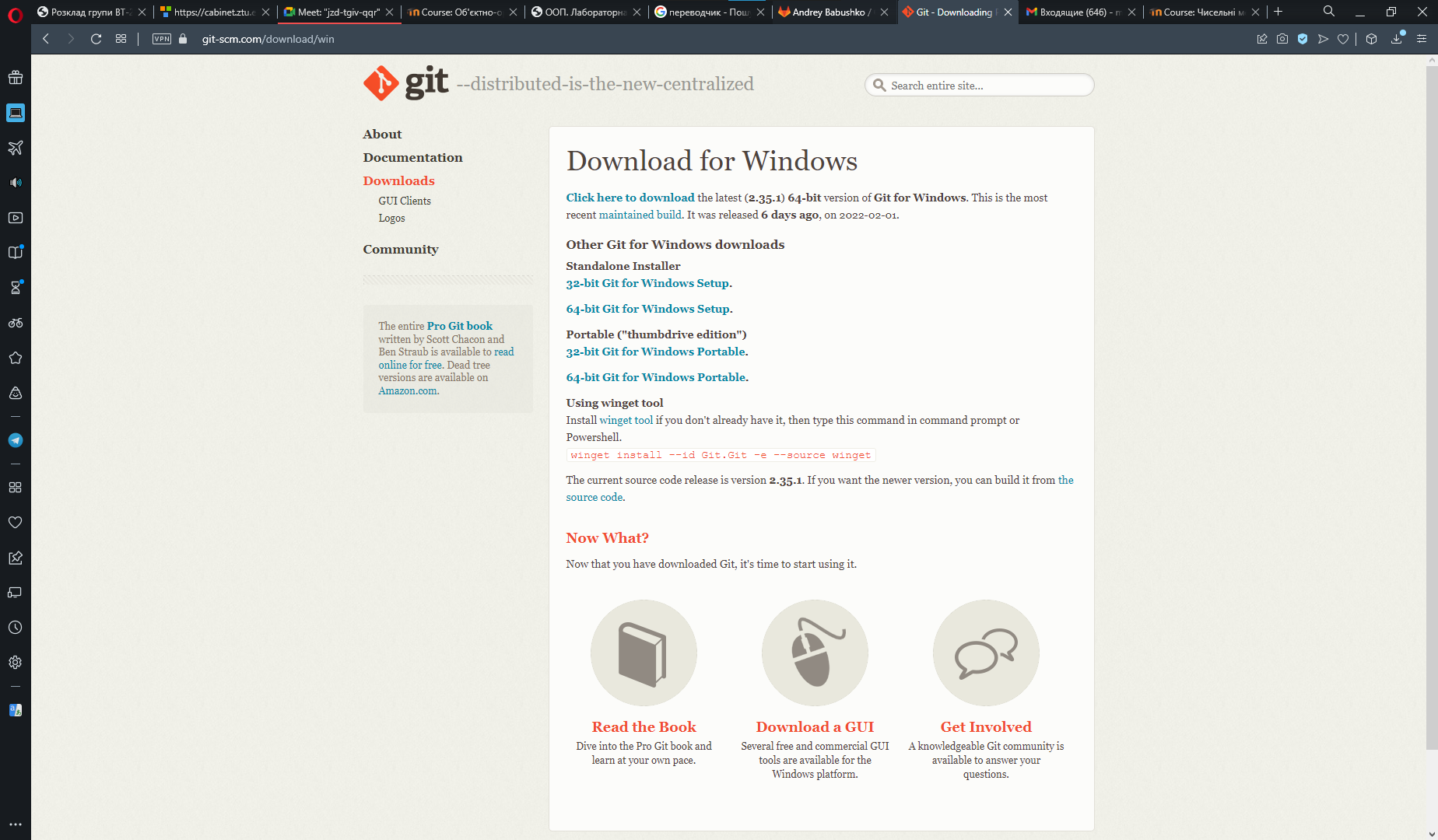
1. ***Створення проекту у Visual Studio.*** 
   1. Створити консольний проект Visual C# з такими параметрами:
      1. Назва рішення: OOP1Lab1;
      2. Назва проекту: ConsoleApp.

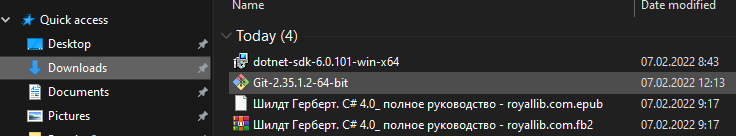


* 1. У програму додати два рядки:
     1. Console.WriteLine("Hello world!");
     2. Console.ReadKey();

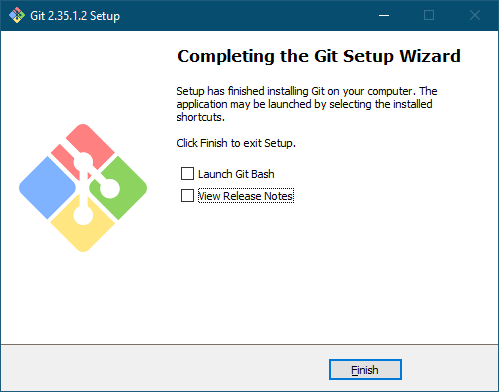


1. ***Встановити програмне забезпечення для системи контролю версій.*** 
   1. Зайти на сайт (https://git-scm.com/), скачати установочний файл:

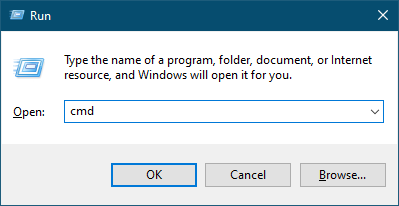




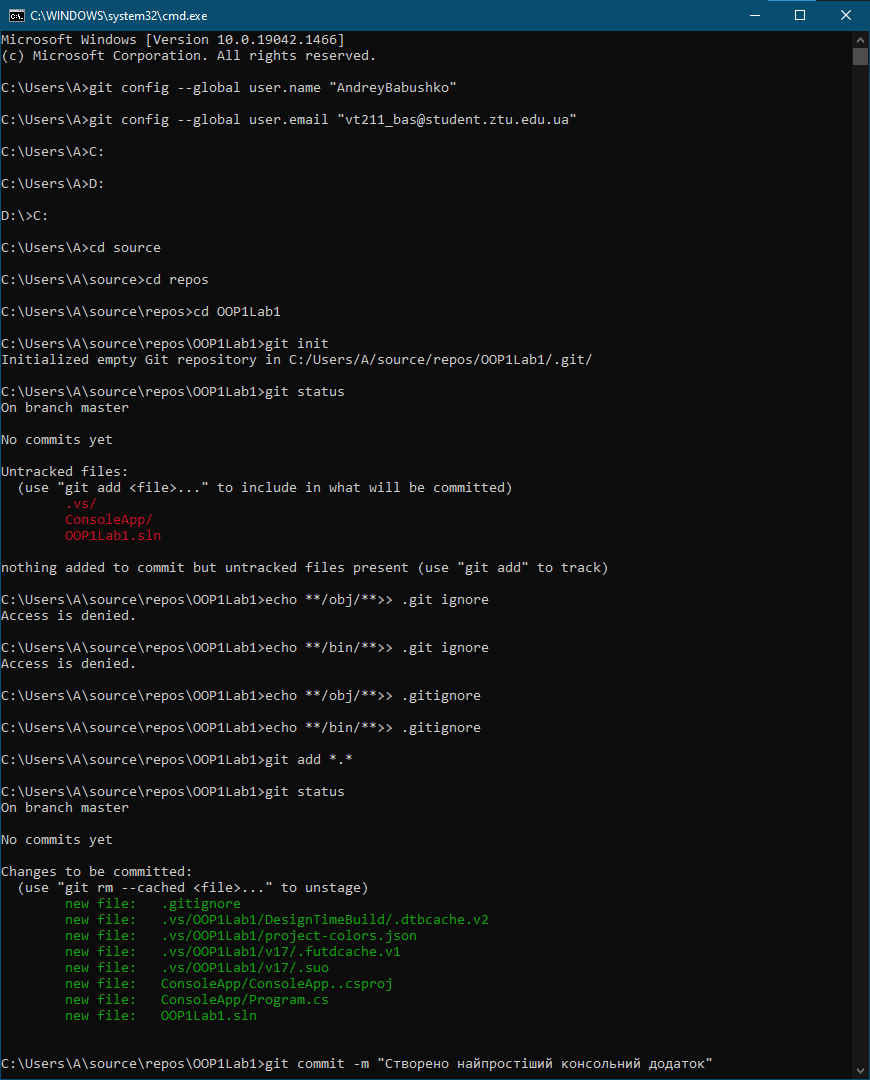
* 1. Запустити установочний файл і у діалогових вікнах вибрати налаштування:

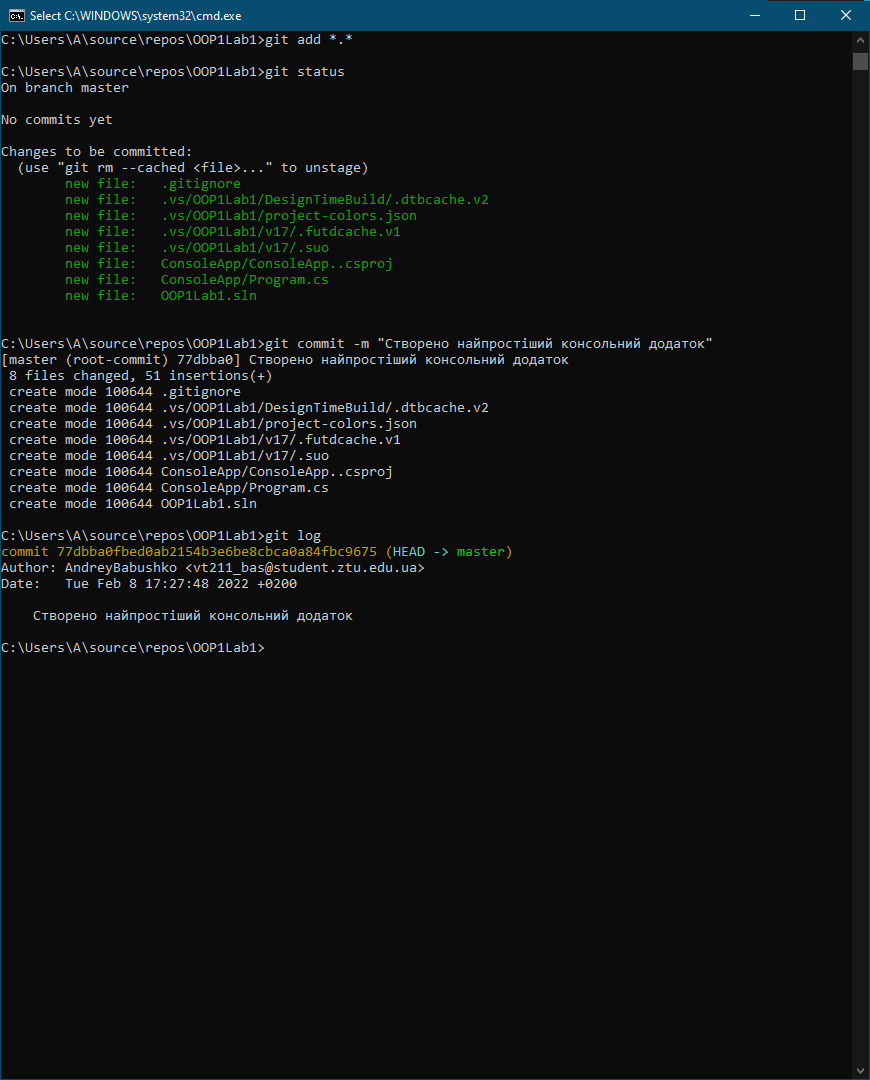


1. ***Виконати початкове налаштування локального репозиторію.*** 
   1. Відкрити командний рядок (комбінація клавіш Win-R, команда “cmd”):

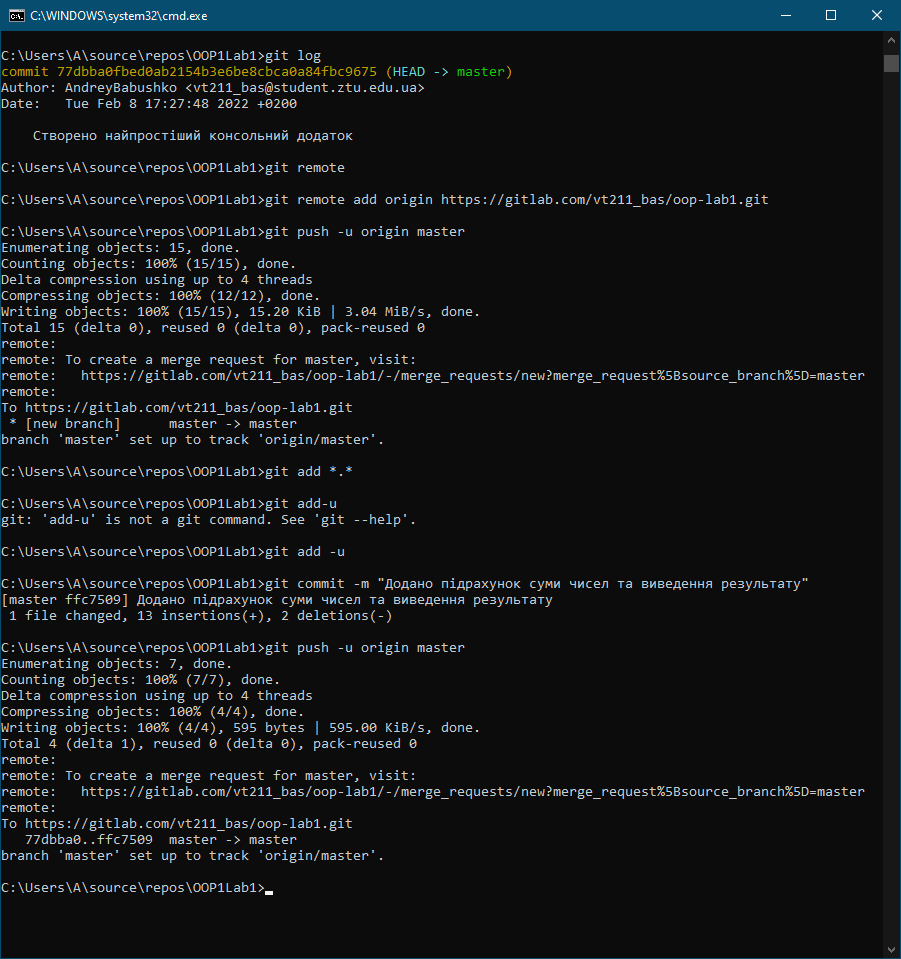


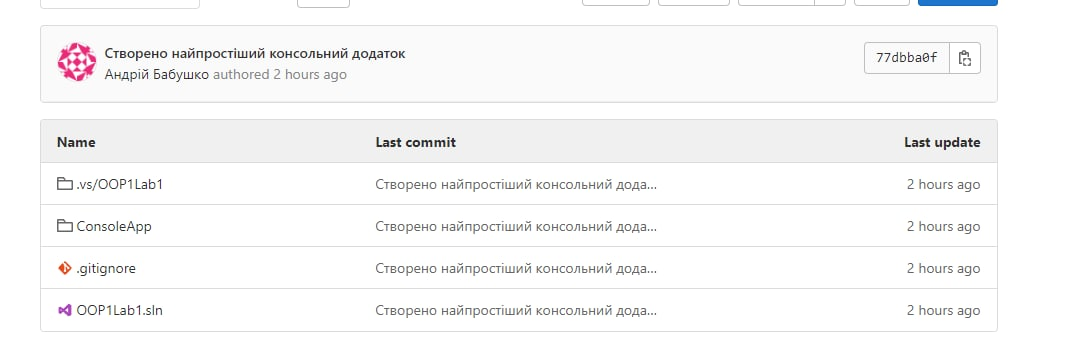
* 1. Виконуємо початкові налаштування:



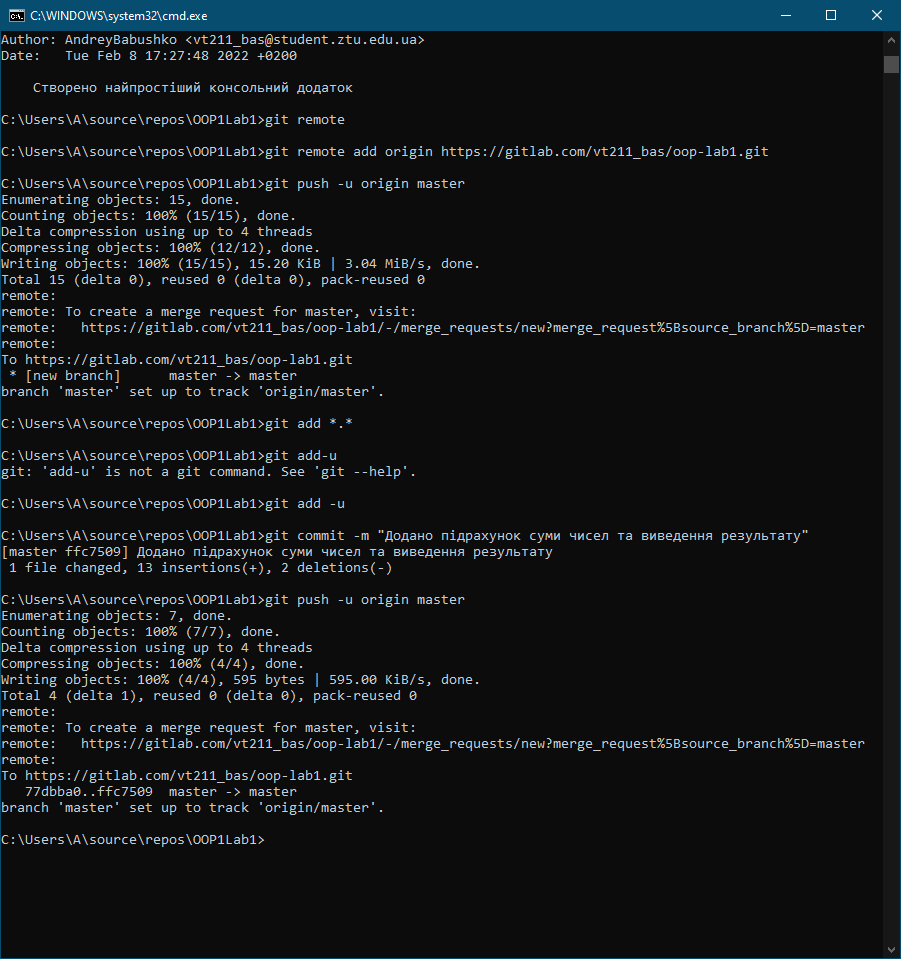


1. ***Підключення віддаленого репозиторіюта відправка комітів***
   1. Переглядаємо список віддалених репозиторіїв, які вже підключені у вашій системі:
   2. Додаємо зв’язок з віддаленим репозиторієм на GitLab.
   3. Завантажити підготовлені локальні коміти на сервер
   4. З’явиться вікно, у якому потрібно буде ввести логін та пароль від сайту GitLab: Виконано.
   5. Якщо всі дії виконано правильно, то з’явиться повідомлення про успішне завантаження файлів:

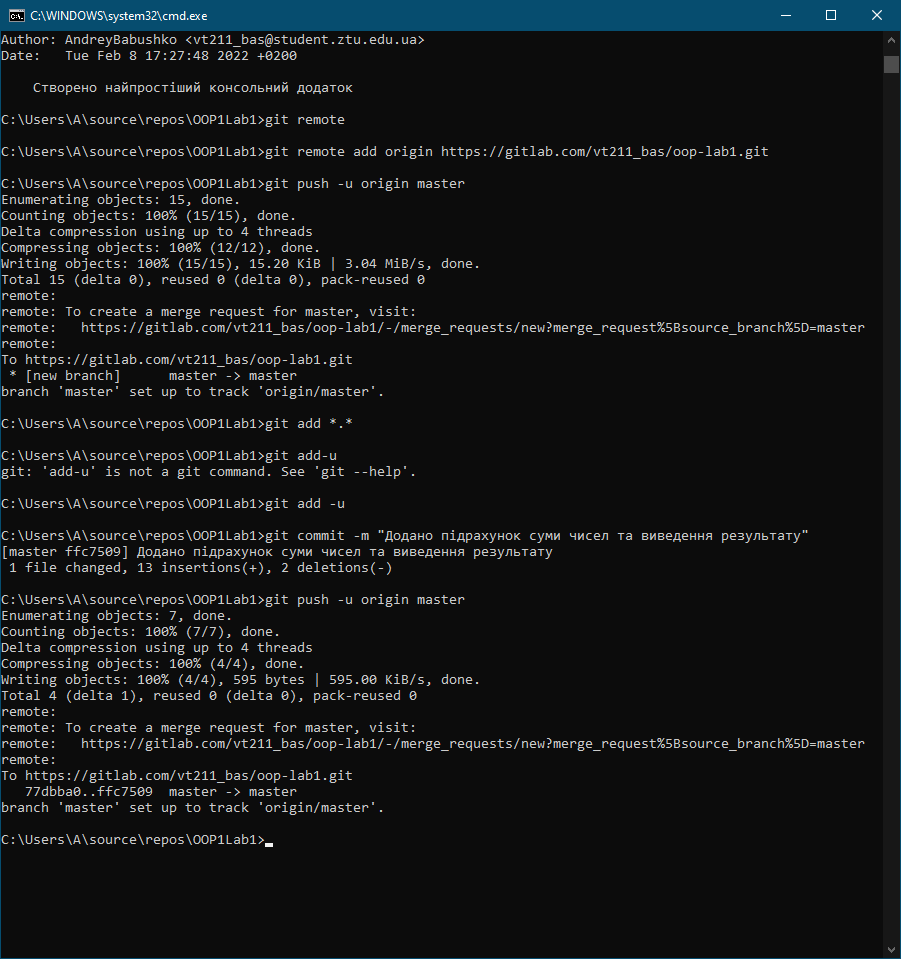




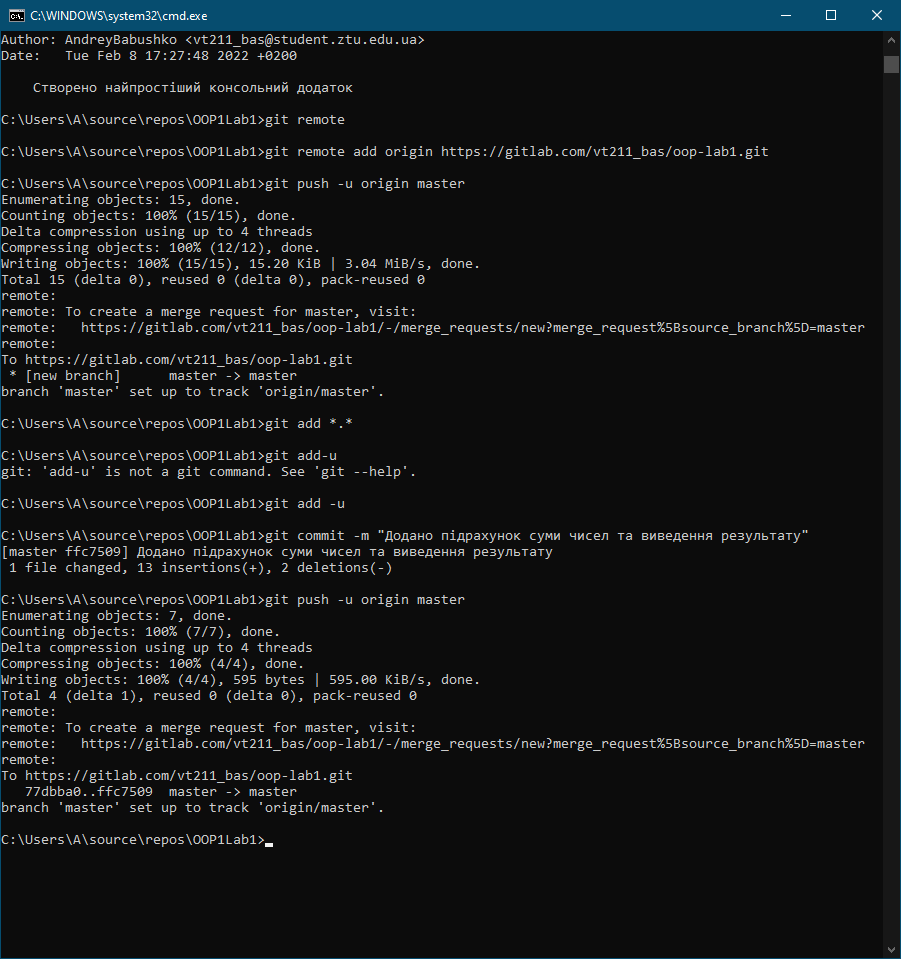
1. ***Додавання змін до рішення та завантаження їх на сервер***
   1. Внесемо зміни у програмний код консольного додатку:



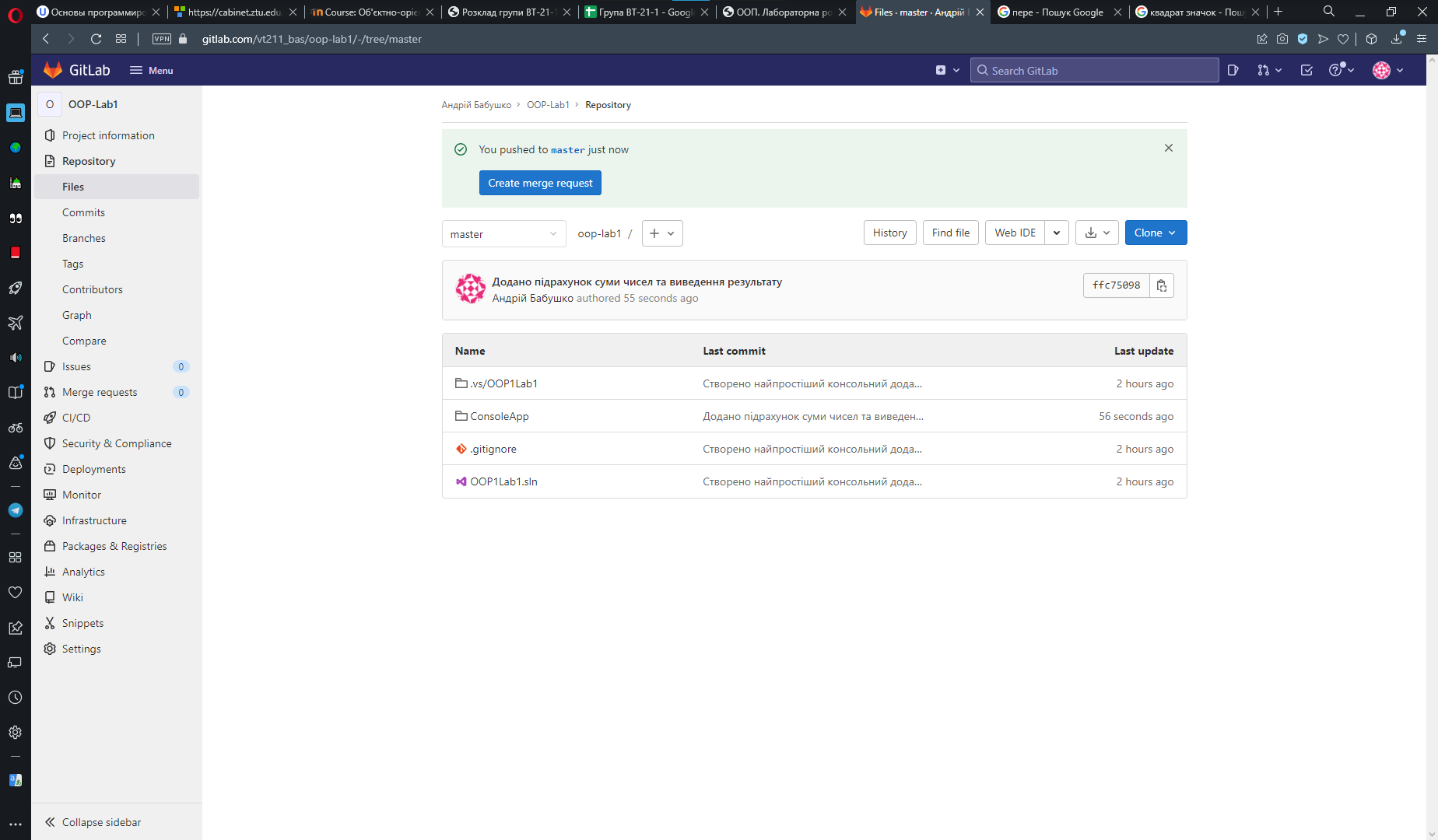
* 1. Для того, щоб їх завантажити на сервер GitLab виконаємо команди:



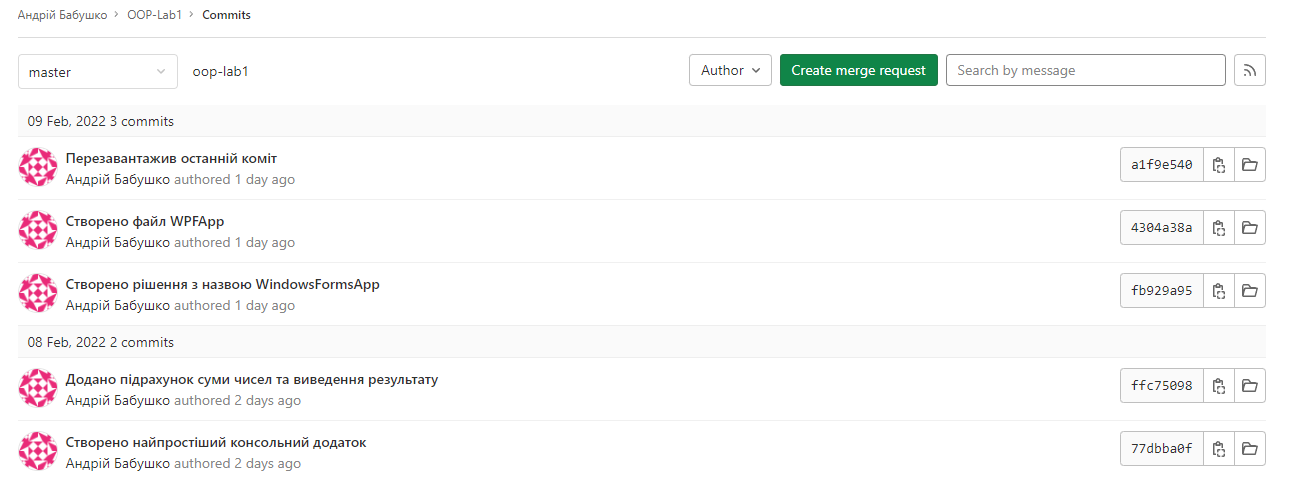
* 1. Тепер відправимо зміни на сервер командою:



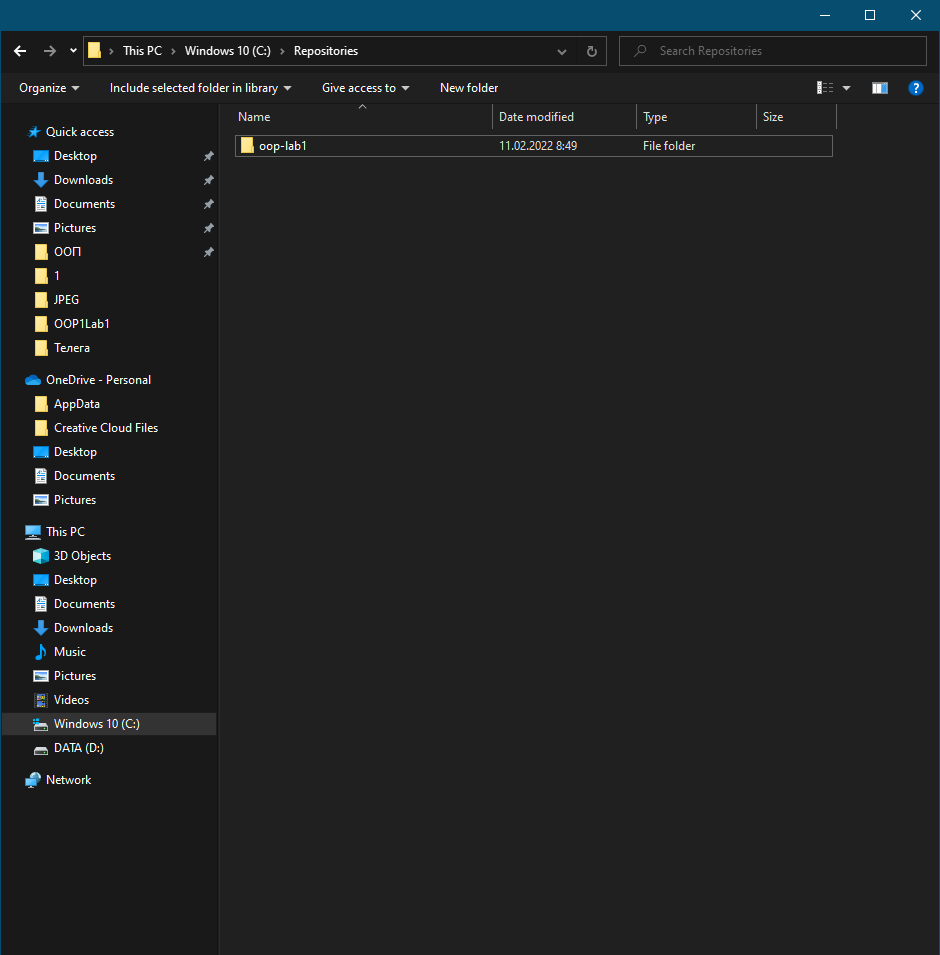
* 1. Перегляньте, чи з’явилися файли на сайті GitLab:



* 1. Створіть ще один проект у рішенні з назвою «WindowsFormsApp»: Виконано.
  2. Завантажте зміни на сервер: Виконано.
  3. Створіть додаток з назвою «WPFApp»типу «Приложение WPF» і завантажте зміни на сервер. Виконано.
  4. Тепер в списку комітів на сайті GitLab має бути чотири коміти:



1. ***Клонування репозиторію***
   1. Перейдемо у кореневий каталог диску C (або іншого): Виконано.
   2. Виконаємо клонування репозиторію з сервера GitLab: Виконано.
   3. Перевіряємо, чи з’явилася папка з рішенням на диску C (або іншому).

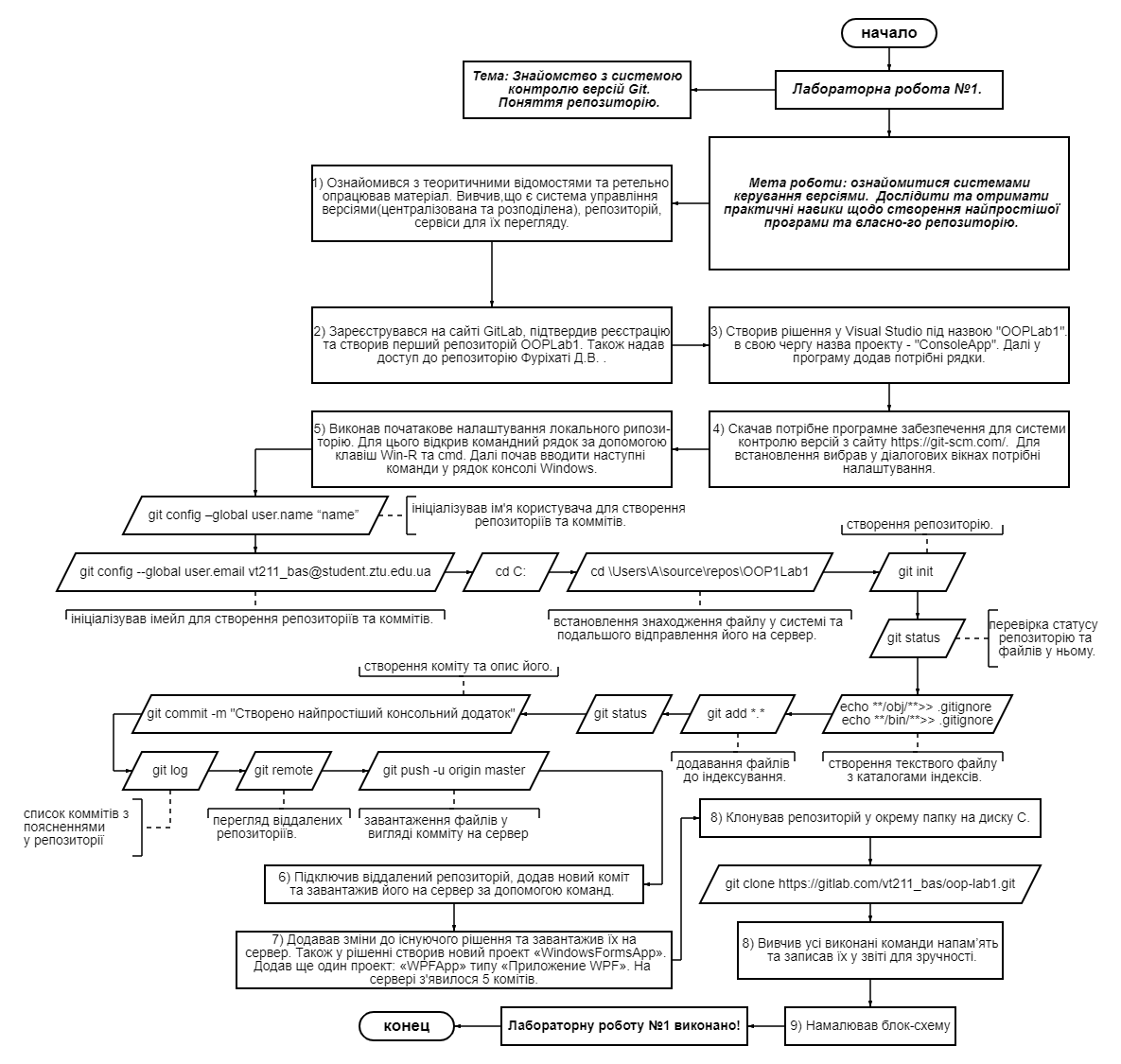


1. **Вивчити усі виконані команди напам’ять (на наступному занятті буде письмова контрольна робота на знання команд та розуміння усіх виконаних кроків).**

git commands які були використані під час заняття:

* 1. git config –global user.name “name” - ініціалізація імені користувача для створення репозиторіїв та коммітів.
  2. git config –global user.email “email” – ініціалізація імейлу для створення репозиторіїв та коммітів.
  3. cd \Шлях до файлу програми – встановлення знаходження файлу у системі та подальшого відправлення його на сервер.
  4. git init – створення репозиторію.
  5. git status – перевірка статусу репозиторію та файлів у ньому.
  6. git log – список коммітів з поясненнями у репозиторії
  7. echo \*\*/obj/\*\*>> .gitignore – створення текствого файлу з каталогами індексів.
  8. echo \*\*/bin/\*\*>> .gitignore – створення текствого файлу з каталогами індексів.
  9. git add \*.\* - додавання файлів до індексування.
  10. git add . – додавання усіх файлів проекту.
  11. git add -u – додавання лише деяких у початковому файлі
  12. git commit -m “Опис комміту” – створення коміту та опис його.
  13. git push -u origin master – завантаження файлів у вигляді комміту на сервер.
  14. git remote – перегляд віддалених репозиторіїв.
  15. git remote remove origin – виключення віддаленого репозиторію.
  16. git remote add origin “ссилка репозиторію” – вказання адресу для завантаження комітів у репозиторій.
  17. git clone “ссилка репозиторію” – копія файлів репозиторію на комп’ютер.

1. **Вивчити матеріал щодо графічного оформлення алгоритмів у вигляді блок-схем:** [**https://ru.wikipedia.org/wiki/Блок-схема**](https://ru.wikipedia.org/wiki/Блок-схема) **. Намалювати блок-схему виконаної роботи:**

****

1. **Оформити звіт з лабораторної роботи** – Виконано.

**Висновок:** під час виконання лабораторної роботи було отримано навички створення репозиторію, комітів та завантаження їх на сервер за допомогою команд git у консолі.