МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА АВТОМАТИЗАЦІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

НАВЧАЛЬНА ДИСЦИПЛІНА

«АЛГОРИТМИ І СТРУКТУРИ ДАНИХ»

ЗВІТ

З ЛАБОРАТОРНОЇ РОБОТИ №1

Виконала:

студентка групи КН-24-1

Процко П.Д.

Перевірив:

доцент кафедри АІС

Сидоренко В. М.

Кременчук 2025

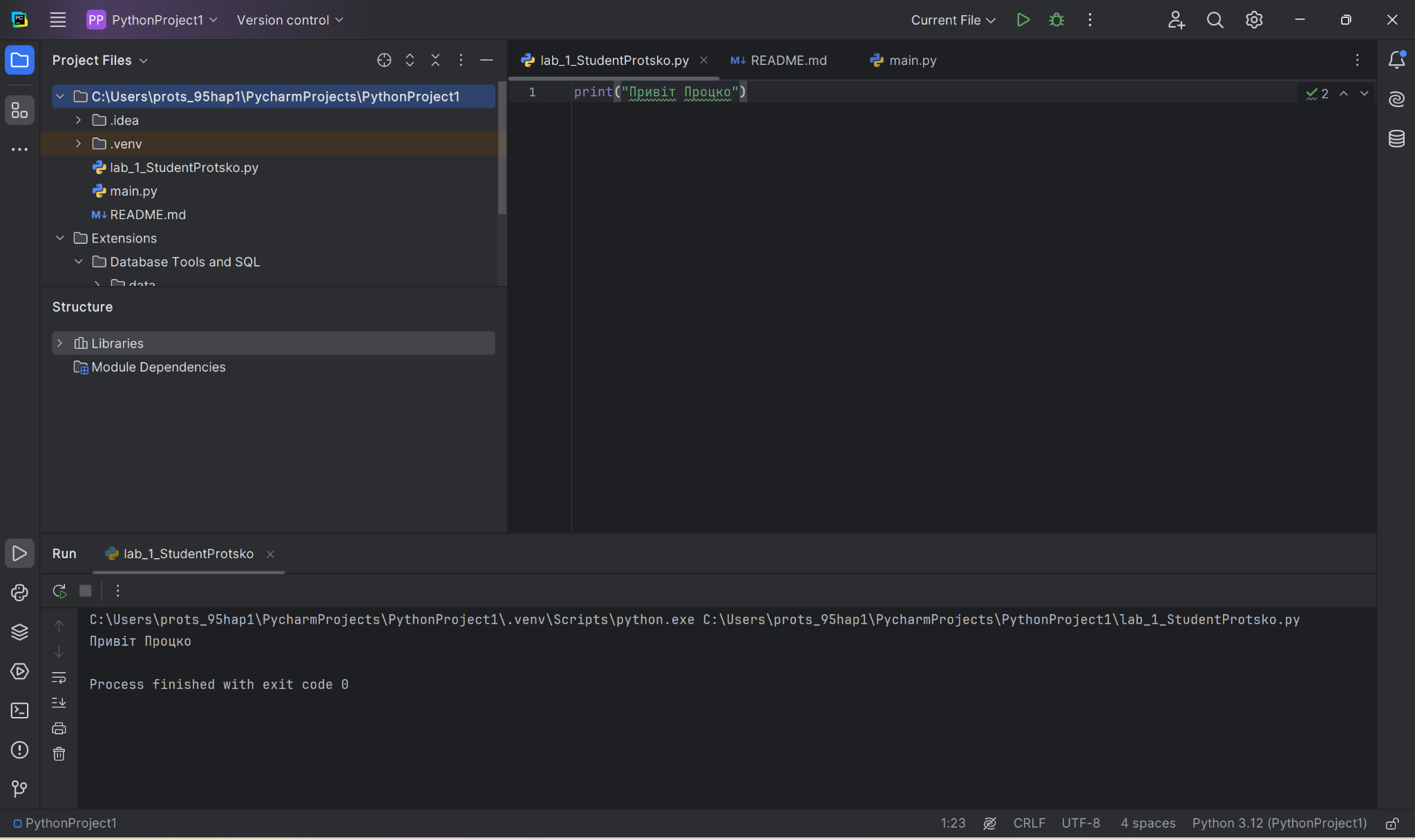
## **Тема**: Налаштування середовища розробки та системи контролю версій Git для роботи з Python-проєктами

## **Мета**: Встановити та налаштувати необхідне програмне забезпечення (Git, Python, PyCharm), створити акаунт на GitHub, навчитися працювати з віддаленими репозиторіями та освоїти базові команди Git для ефективної розробки та управління версіями проєктів.

### **1. Створення акаунту на GitHub та налаштування доступу**

* Зареєструвалася на GitHub за посиланням <https://github.com>
* Створила новий репозиторій з назвою "ProtskoPolina" з файлом README та налаштувала доступ до репозиторію

**2. Встановлення Python**

* Зайшла на офіційний сайт Python (python.org)
* Завантажила актуальну версію Python
* Встановивла інтерпретатор з параметрами за замовчуванням, включно з додаванням Python до змінної PATH
* Перевірила коректність встановлення, запустивши команду python в командному рядку

### **3. Встановлення та налаштування IDE PyCharm**

* Завантажила безкоштовну Community версію PyCharm з офіційного сайту
* Встановила IDE з налаштуванням змінної PATH
* Перезавантажила комп'ютер після завершення інсталяції
* Запустила PyCharm і переконався у коректному запуску середовища

## **Висновки:**

В результаті виконання лабораторної роботи я:

1. Навчилася створювати та налаштовувати акаунт на GitHub, створювати віддалені репозиторії для зберігання коду.
2. Встановила та налаштувала Python та середовище розробки PyCharm для ефективної роботи над проєктами.
3. Опанувала базові команди Git для контролю версій: ініціалізацію репозиторію, додавання файлів до індексу, комітування змін та синхронізацію з віддаленим репозиторієм.

Відповіді на контрольні питання:

1. **Що таке IDE PyCharm?**  
   PyCharm — це інтегроване середовище розробки (IDE) для програмування на мові Python. Воно надає зручні інструменти для написання, відлагодження та тестування коду. PyCharm має багато корисних функцій, таких як автодоповнення коду, рефакторинг, інтеграція з системами контролю версій, підтримка веб-фреймворків і багато іншого.
2. **Що таке система контролю версій (СКВ) і для чого вона призначена? У чому полягає принципова відмінність СКВ SVN і GIT?**  
   Система контролю версій (СКВ) — це інструмент, який дозволяє зберігати історію змін у файлах і координувати роботу кількох людей над проектом. Вона допомагає відслідковувати зміни в коді, повертатися до попередніх версій, а також забезпечує співпрацю між розробниками.

**Принципова відмінність між SVN і GIT:**

* + **SVN** (Subversion) — це централізована система контролю версій, що означає, що є один центральний репозиторій, а всі користувачі працюють з ним безпосередньо.
  + **GIT** — це дистрибутивна система контролю версій, що означає, що кожен користувач має власну копію репозиторію на своєму комп’ютері, і зміни можуть синхронізуватися з центральним репозиторієм лише при необхідності.

1. **Що відбувається унаслідок виконання команди git add .?**  
   Команда git add . додає всі зміни в поточному каталозі (і його підкаталогах) до індексу (стейджинг-ділянки) Git, тобто готує ці зміни для наступного коміту. Вона не змінює репозиторій безпосередньо, але вказує Git, що ці зміни мають бути збережені в історії.
2. **Що відбувається унаслідок виконання команди git commit?**  
   Команда git commit зберігає зміни, які були додані до індексу (стейджинг-ділянки) в локальний репозиторій Git. Коміт містить збережений код разом з коментарем, який описує зміни. Цей коміт є окремою одиницею в історії репозиторію.
3. **Якою Git-командою виконується розміщення локальної копії Git-репозиторію проєкта на віддаленому, що знаходиться на GitHub?**  
   Для розміщення локальної копії репозиторію на віддаленому сервері, що знаходиться на GitHub, використовується команда:

git push origin main

Це відправляє локальні зміни на віддалений репозиторій, зокрема на гілку main (або іншу, якщо вона вказана).