

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

Виконав:
студент групи КН-24-1
Соломка Б. О.

Перевірив:
доцент кафедри АІС
Сидоренко В. М.

Кременчук 2025

Тема: Налаштування середовища розробки та системи контролю версій Git для роботи з Python-проектами

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.

Хід роботи:

1. Створення акаунту на GitHub та налаштування доступу

Зареєструвався на GitHub за посиланням <https://github.com/>

Створив новий репозиторій з назвою "BorysSolomka" з файлом README та налаштував доступ до репозиторію через HTTPS протокол

2. Встановлення Python

Зайшов на офіційний сайт Python (python.org)

Завантажив актуальну версію Python.

Встановив інтерпретатор з параметрами за замовчуванням, включно з додаванням Python до змінної PATH

Перевірів коректність встановлення, запустивши команду `python` в командному рядку

3. Встановлення та налаштування IDE PyCharm

Завантажив безкоштовну Community версію PyCharm з офіційного сайту

Встановив IDE з налаштуванням змінної PATH

Перезавантажив комп'ютер після завершення інсталяції

Запустив PyCharm і переконався у коректному запуску середовища

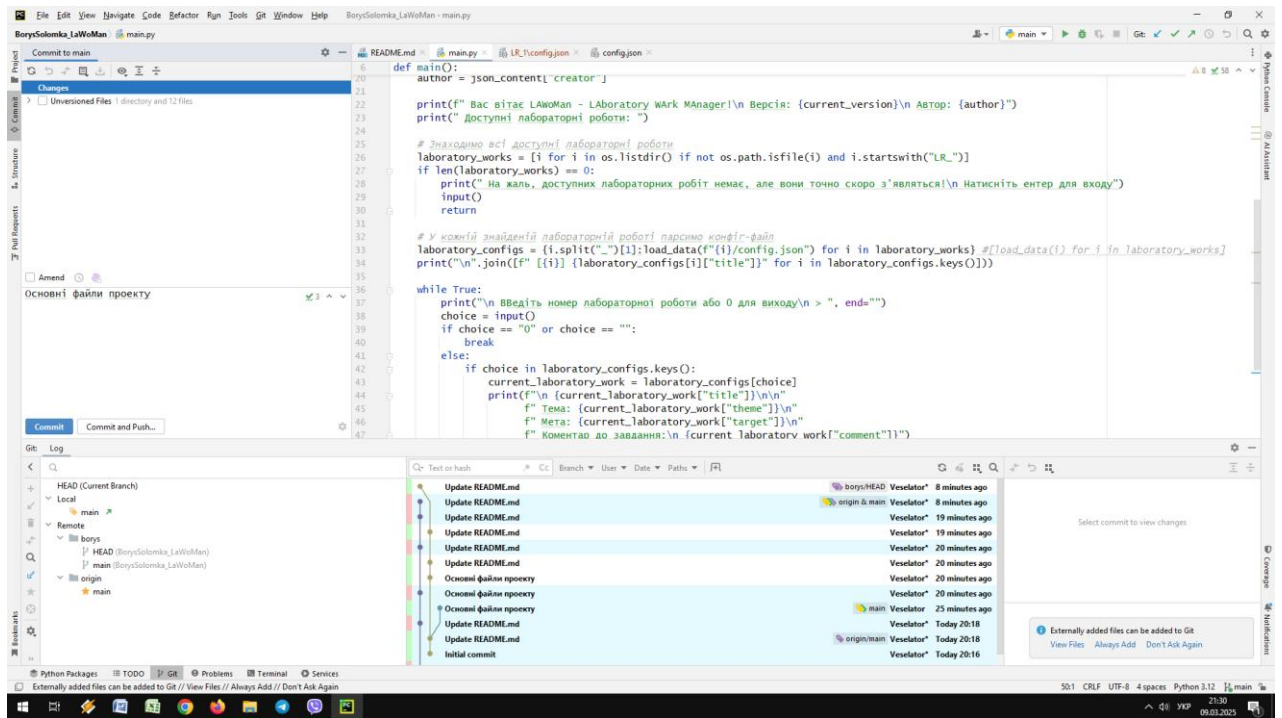


Рисунок 1 – Інтерфейс середовища PyCharm

4. Створення Python-проєкту та зв'язування з Git-репозиторієм

Створив новий проєкт у PyCharm з відповідними налаштуваннями:

Вибрав віртуальне середовище

Визначив розташування проєкту

Вибрав версію інтерпретатора Python

Ініціалізував Git-репозиторій у проєкті через меню VCS → Create Git Repository

Зв'язав локальний репозиторій з віддаленим на GitHub через меню Git → Manage Remotes, вказавши URL мого репозиторію

```
D:\Борис№3\University\BorysSolomka_LaWoMan>git clone https://github.com/ASD-122-2025/BorysSolomka.git
Cloning into 'BorysSolomka'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (17/17), done.
remote: Total 21 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (21/21), 7.29 KiB | 3.65 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

Рисунок 2 – Приклад роботи із Git через консольний інтерфейс

5. Створення та комітування Python-файлу

Створив файл `main.py` та написав програму, яка реалізує менеджмент лабораторних робіт. Хоча на даний момент в такому рішенні немає необхідності, в подальшому саме через `main.py` будуть запускатись скрипти лабораторних робіт.

Додав файл до індексу Git через меню Git → Add

Закомітив зміни з коментарем "Додано першу лабораторну роботу"

6. Відправлення змін на GitHub

Відправив зміни на віддалений репозиторій через меню Git → Push

Перевірів наявність змін на GitHub

Висновки:

В результаті виконання лабораторної роботи я:

Навчився створювати та налаштовувати акаунт на GitHub, створювати віддалені репозиторії для зберігання коду.

Встановив та налаштував Python та середовище розробки PyCharm для ефективної роботи над проєктами.

Опанував базові команди Git для контролю версій: ініціалізацію репозиторію, додавання файлів до індексу, комітування змін та синхронізацію з віддаленим репозиторієм.

Створив локальний Python-проєкт і успішно зв'язав його з віддаленим репозиторієм на GitHub, що дозволяє зберігати всі версії проєкту та ефективно працювати як самостійно, так і в команді.

Зрозумів основи інтеграції середовища розробки PyCharm з системою контролю версій Git, що значно спрощує процес розробки.

Отримані навички та знання стануть основою для подальшого навчання та розробки складніших проєктів з використанням Python, а також для ефективної командної роботи з використанням інструментів контролю версій.