

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут комп'ютерних наук та інформаційних технологій

Кафедра систем штучного інтелекту

Ірина Швороб

Методичні вказівки до лабораторних робіт з навчальної  
дисципліни

**Прикладне програмування**

Львів – 2018

## Вступ

Для виконання лабораторних робіт рекомендовано використовувати Python 3.6 (можна завантажити тут <https://www.python.org/downloads/release/python-366/> )

Рекомендовані ресурси:

1. Allen Downey. Think Python, 2nd Edition. How to Think Like a Computer Scientist / O'Reilly, 2015. - 289 p.
2. Charles R. Severance. Python for Everybody: Exploring Data in Python 3 / CreateSpace Independent Publishing Platform, 2016. - 244 pages
3. Wes McKinney & PyData Development Team. pandas: powerful Python data analysis toolkit
4. <https://docs.python.org/3.6/>
5. <https://www.codecademy.com/learn/learn-python>
6. <https://learn.edx.org/mit-python/>
7. <https://www.learnpython.org/>
8. <https://bottlepy.org/docs/dev/>
9. <http://flask.pocoo.org/>
10. <http://www.django-rest-framework.org/>

Кожна лабораторна робота оцінюється 5-ма балами. Для лабораторних робіт, які поділені на два рівні, **перший рівень є обов'язковим до виконання і без виконання всіх завдань цього рівня лабораторна робота не буде прийнятою, оцінюється 2,5 бали.**

### **Лабораторна робота № 1. Робота з інтегрованим середовищем розробки IDE. Типи даних. Умовні операції та цикли.**

**Завдання 1.** Вибрати з переліку запропонованих інтегрованих середовищ два, встановити та налаштувати їх, створити, відлагодити та запустити тестові проекти. Здійснити порівняння обраних середовищ.

Рекомендовано для огляду наступні IDE: Atom, Emacs, Eric Python IDE, Idle, Komodo, NetBeans\*, PyCharm, PyDev(Eclipse), Spyder, UliPad, Vim, PTVS, Pyscripter, Ninja-IDE.

**Завдання 2.** Написати програму, що виводить 'hello world'.

**Завдання 3.** Написати функцію, що приймає два параметри: два додатні числа, а повертає boolean значення чи перше число ділиться на друге без остачі. Передбачити вивід помилки (Exception), коли введено хоча б одне від'ємне число.

**Завдання 4.** Використовуючи функцію із попереднього завдання створити нову функцію, яка приймає два параметри – два числа, та повертає масив простих чисел у цьому відрізку або видає помилку 'NoSimpleDigits', коли таких немає.

### **Завдання 5. Дано масив**

```
['a', ['c', 1, 3], ['f', 7, [4, '4']], [{'lalala': 111}]]
```

Написати програму, яка перетворить заданий масив з кількома рівнями вкладеності (масив з неоднотипними елементами) в масив з одним рівнем вкладеності. Тобто програма має вивести масив, з такими елементами:

```
['a', 'c', 1, 3, 'f', 7, 4, '4', {'lalala': 111}]
```

При написанні програми передбачити можливість введення іншого масиву з іншими даними в якості контрольного прикладу для перевірки коректності роботи розробленого алгоритму.

## **Лабораторна робота № 2. Робота з файлами.**

### **Рівень 1.**

**Завдання 1.** Написати програму згідно з варіантом. Варіант визначається за порядковим номером у списку підгрупи.

Варіант 1. Написати програму мовою Python для виведення *n* перших рядків з текстового файлу.

Варіант 2. Написати програму мовою Python для додавання тексту в кінець текстового файлу та вивести весь текст.

Варіант 3. Написати програму мовою Python для додавання тексту після *n*-го рядка текстового файлу та вивести весь текст.

Варіант 4. Написати програму мовою Python для виведення *n* останніх рядків з текстового файлу.

Варіант 5. Написати програму мовою Python для зчитування текстового файлу рядок за рядком та записати ці рядки у список (list).

Варіант 6. Написати програму мовою Python для зчитування текстового файлу рядок за рядком та записати ці рядки у масив (array).

Варіант 7. Написати програму мовою Python для знаходження найдовшого слова, записаного в текстовий файл.

Варіант 8. Написати програму мовою Python для знаходження найкоротшого слова, записаного в текстовий файл.

Варіант 9. Написати програму мовою Python для підрахунку кількості рядків в текстовому файлі.

Варіант 10. Написати програму мовою Python для підрахунку кількості слів в текстовому файлі.

Варіант 11. Написати програму мовою Python для підрахунку кількості зустрічання слів у текстовому файлі.

Варіант 12. Написати програму мовою Python для визначення розміру текстового файлу.

Варіант 13. Написати програму мовою Python для запису даних зі списку (list) у файл.

Варіант 14. Написати програму мовою Python, яка комбінує кожен рядок з першого файлу з відповідним рядком у другому файлі та записати все в третій файл.

Варіант 15. Написати програму мовою Python, яка виводить випадковий рядок з текстового файлу.

Варіант 16. Написати програму мовою Python, яка видаляє з текстового файлу всі символи переходу на новий рядок.

**Завдання 2.** Створити файл a.txt із віршем Тараса Шевченка «Садок вишневий коло хати» (Забезпечити правильність закриття файлу, роботи з дескриптором файлів). Створити файл b1.txt, в який записати кожен парний рядок із файлу a, всі слова записати у верхньому регістрі. Створити файл b2.txt, в який записати всі непарні рядки, всі слова повинні бути записаними у нижньому регістрі.

## **Рівень 2.**

**Завдання 3.** Створити файл c.xml (згідно з стандартами), записати в нього всі унікальні слова та кількість їх вживання у тексті, використовуючи файл a.txt із завдання 2.

**Завдання 4.** Використовуючи регулярні вирази записати слова, що однаково закінчуються (останні 3 букви), у файл c.xml (закінчення - кількість повторень - список кортежів (tuple) із слова, номер рядка, номер слова в рядку).

## **Лабораторна робота № 3. Написання класів, що відповідають основним принципам ООП. Реалізація простого веб-сервера. Робота з SQL базами даних.**

**Завдання 1.** Дано клас UserBasic.

```
class UserBasic(object):
    def __init__(self, first_name, mid_name, last_name):
        self.first_name = first_name
        self.mid_name = mid_name
        self.last_name = last_name

    def full_name(self):
        return f'{self.last_name} {self.first_name} {self.mid_name}'
```

Використовуючи клас UserBasic, створити власний клас із додатковими методами, яких потребує ваш варіант (порядковий номер у списку підгрупи). Створити API (REST) для створення, редагування, видалення та читання даних із відповідним збереження у БД. CRUD для користувачів та інших об'єктів у БД. Реалізувати механізм транзакцій.

(Використати один із запропонованих фреймворків: flask bottle, django api, як базу даних вибрати щось з SQLite, MySQL, PostgreSQL)

Варіант 1. Створити сервіс переказу коштів між користувачами, кожен користувач має власний гаманець та можливість переказувати чи отримувати кошти від іншого користувача.

Варіант 2. Створити сервіс коротких (404 символи) заміток (із тегами) для кожного користувача із можливістю перегляду, редагування і видалення, а також надавати доступ редагувати замітку іншими користувачами (до 5 користувачів). Також надати можливість бачити статистику користувача, скільки повідомлень, коли редаговані і ким.

Варіант 3. Створити сервіс оголошень + CRUD із двома рівнями повідомлень. Оголошення повинні бути локальними та публічними. Локальні оголошення тільки для користувачів, що знаходяться в тому ж місці. Публічні для всіх, навіть для не користувачів сервісу.

Варіант 4. Створити сервіс кредитування користувачів на основі даних, що користувач вводить при реєстрації, кошти для кредитування видаються із бюджету 517 000 грн ставка 30%. Також реалізувати можливість погашення кредиту.

Варіант 5. Написати сервіс статей (2000 символів). Статті є публічними для всіх, зареєстровані користувачі можуть редагувати статтю та очікувати на схвалення її модераторами (користувачі із більшими правами). Передбачити варіант редагування, коли стаття на розгляді модератором, а інший користувач її теж редагує. Модератори мають бачити статті, які очікують їх схвалення.

Варіант 6. Написати сервіс простий інтернет магазин. Користувачі можуть купувати один із 8 товарів, які є в обмеженій кількості на складі, не допустити можливості продажу одного і того ж товару кільком користувачам.

Варіант 7. Створити сервіс для резервування аудиторій на певну дату час та проміжок часу від 1 години до 5 днів. Користувачі мають можливість резервувати аудиторії, а також редагувати, скасовувати та видаляти їх. Застерегти користувачів від накладок (два користувачі не можуть зарезервувати аудиторію на певний період час)

Варіант 8. Написати сервіс для купівлі та бронювання квитків на концерти, події і т.д. Користувачі мають можливість купувати квиток, бронювати квиток,

скасовувати бронь. Унеможливити купівлю чи бронювання одного і того ж квитка кількома користувачами.

Варіант 9. Написати сервіс для створення плейлистів. Користувач може мати як приватні (видимі тільки для нього), так і публічні (видимі для всіх) плейлисти. Публічні плейлисти можуть редагувати всі користувачі.

Варіант 10. Створити сервіс для календаря подій. Користувач має можливість створювати подію, редагувати її, видаляти, долучати інших користувачів до події, переглядати перелік всіх створених подій, та подій до яких він долучений.

Варіант 11. Створити сервіс для проведення онлайн занять. Повинні бути користувачі двох рівнів – викладачі та учні. Викладачі можуть створювати, видаляти, редагувати курс, переглядати перелік створених курсів, долучати студентів до курсу. До курсу може бути долучено не більше 5 студентів. Студенти можуть переглядати всі дані лише тих курсів, до яких вони долучені. Також студенти можуть надсилати на участь у якомусь курсі, а викладач має можливість прийняти або відхилити запит.

Варіант 12. Створити сервіс для збереження та редагування рейтингу студентів. Для зберігання даних про студента використовувати json. Також реалізувати можливість отримання списку кращих за рейтингом студентів.

Варіант 13. Написати сервіс для роботи з сімейним бюджетом на спільному рахунку. В сім'ї повинно бути не менше 3 людей. Кожен користувач має можливість переглядати бюджет, додавати в нього кошти, знімати кошти на свій персональний рахунок. Також передбачити можливість збереження та виведення переліку витрат та доходів як сім'ї загалом, так і кожного її учасника.

Варіант 14. Створити сервіс для прокату авто. Користувачі сервісу можуть бути двох рівнів – адміністратори та пасажери. Адміністратори можуть додавати та видаляти авто із системи, редагувати інформацію про авто. Пасажири можуть переглядати каталог та бронювати авто на певний час.

Варіант 15. Написати сервіс для роботи аптеки. Провізор може додавати препарати в базу, видаляти, редагувати інформацію про них. Користувач може переглядати інформацію про препарати, здійснювати купівлю, якщо немає препарату в наявності, то його можна додати в попит.

Варіант 16. Написати сервіс для роботи кінотеатру. Адміністратор може скласти розклад показу фільмів, з урахуванням тривалості фільму та зайнятості залу, редагувати розклад, видаляти та додавати фільми.

## **Лабораторна робота № 4. Покриття коду тестами (для лаб. №3)**

### **Рівень 1.**

**Завдання 1.** Покрити код з лабораторної роботи №3 на 80% тестами. Тести створювати за принципами модульного тестування. Для тестування всі студенти, що мають непарний порядковий номер у списку групи, повинні використати бібліотеку `unittest`, всі, що мають парний порядковий номер - `py.test`.

**Завдання 2.** Ознайомитись з основними принципами інтеграційного тестування за такими підходами: Bottom Up Integration, Top Down Integration, "Big Bang" Integration.

### **Рівень 2.**

**Завдання 3.** Покрити код з лабораторної роботи №3 на 50% тестами, для тестування всі студенти, що мають парний порядковий номер у списку групи, повинні використати бібліотеку `unittest`, всі, що мають непарний порядковий номер - `py.test`.

На основі завдань 1 та 3 навести порівняння бібліотек для тестування коду.