ПРИКАРПАТСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ВАСИЛЯ СТЕФАНИКА

ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ІНФОРМАТИКИ

КАФЕДРА ДИФЕРЕНЦІАЛЬНИХ РІВНЯНЬ І ПРИКЛАДНОЇ МАТЕМАТИКИ

**З В І Т**

з виробничої практики

студентки 4 курсу освітнього рівня бакалавр групи ПМ-41

Гуцуляк Аліни Володимирівни

База практики ТзОВ Елекс

Термін практики з «30» січня 2023 р.

до «10» березня 2023 р.

Керівники практики:

Від бази директор Коновалов П. О.

Від кафедри доцент Мазуренко В. В.

ЗМІСТ

[ВСТУП 3](#_Toc127533586)

[РОЗДІЛ 1. Актуальність мови Python в програмуванні 4](#_Toc127533587)

[***1.1.*** ***Переваги мови Python*** 4](#_Toc127533588)

[***1.2.*** ***Особливості роботи в середовищі Python*** 5](#_Toc127533589)

[РОЗДІЛ 2. Розробка індивідуального завдання 7](#_Toc127533590)

[***2.1. Постановка індивідуального завдання*** 7](#_Toc127533591)

[***2.2. Робота з файлами*** 7](#_Toc127533592)

[***2.3. Робота з вмістом файлу*** 8](#_Toc127533593)

[***2.4. Сортування отриманих даних з файлу*** 9](#_Toc127533594)

[ВИСНОВКИ 11](#_Toc127533595)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 12](#_Toc127533596)

[ДОДАТОК 1 13](#_Toc127533597)

# ВСТУП

Виробнича практика студентів є одним із важливих видів навчальної роботи і здійснюється з метою набуття виробничих навиків приймати самостійно рішення в реальних виробничих умовах. Вона покликана підготувати майбутніх спеціалістів до реальної практичної роботи, забезпечити належний рівень їхньої професійної підготовки.

Практика є необхідною складовою підготовки фахівців. Метою практичної підготовки є оволодіння студентами сучасними методами, формами організації та знаряддями праці в галузі їх майбутньої професійної діяльності, формування на базі одержаних знань професійних умінь і досвіду прийняття самостійних рішень під час конкретної роботи в реальних ринкових і виробничих умовах, виховання потреби систематично оновлювати свої знання та творчо їх застосовувати в практичній діяльності.

У мене з’явилась можливість проходити виробничу практику на базі іноваційної компанії Eleks. Тривалість практики, згідно наказу, у період з 30 січня по 10 березня.

ELEKS (Елекс) — IT-компанія з українським корінням, яка надає послуги з розробки програмних продуктів, консультування та забезпечення контролю якості. Входить до 100 найкращих аутсорсингових підприємств світу 2022 року. Компанія відома своїми продуктами для сучасного кінематографу та додатками для автоматизації діяльності медичних установ (наприклад, інформаційна система Doctor Eleks). Діяльність компанії спрямована на міжнародний ринок: 95 % клієнтів ELEKS — закордонні замовники і лише 5 % — українські клієнти. ELEKS — на 7-му місце в рейтингу 50 найкращих роботодавців України і на 3-му серед ІТ-роботодавців у 2022 році за версією Forbes.

**Мета практики**: вдосконалити набуті знання у галузі програмування, використати їх на практиці виконуючи індивідуальне завдання; навчитись співпрацювати з колективом на обраній базі практики й виконувати поставлені завдання керівником практики.

**Завдання практики**: вивчення організації і етапів розробки програмного продукту, набуття практичних навичок програмування, проектування та реалізації індивідуального завдання.

# РОЗДІЛ 1. Актуальність мови Python в програмуванні

## ***Переваги мови Python***

Сьогодні, коли обчислювальні потужності персональних комп'ютерів і серверів стали досить високими, з'явився величезний попит на інтерпретовані мови програмування. Адже крім запуску самої програми необхідний запуск інтерпретатора, що вимагає додаткових ресурсів. Саме такою мовою і є Python.

Популярність Python обумовлена ​​тим, що дана мова здатна вирішувати широке коло завдань і його застосування можливе на всіх популярних на сьогоднішній день платформах, за винятком мобільного сегменту, на якому щільно вкоренилися специфічні мови. В останні роки Python придбав свою популярність за рахунок ефективності в таких сферах розробки як Machine Learning і Data Science завдяки своїй розширюваності і гнучкості, що так необхідно в даному сегменті програмування. У той же час Python широко застосовується в WEB, в якому дана мова відчуває себе чудово з початку 2000-х років і до цього дня розвиває і вдосконалює цей напрямок.

***Переваги:***

* *Гнучкість* — основна перевага мови, так як завдяки своїй гнучкості мова отримала популярність серед багатьох розробників.
* Можливість розширення — один із слоганів мови звучить як — Just Import! — що повністю пояснює, наскільки мова розширюється і була розширена за останні роки. Існують бібліотеки і фреймворки під будь-який тип завдань і потреб. Також величезним плюсом є те, що ми можемо використовувати C код з Python.
* Простота синтаксису. Синтаксис — з синтаксису було прибрано все зайве, код чистий і зрозумілий без зайвих дужок і виразів.
* Інтерпретованість. Інтерпретатор Python існує для всіх популярних платформ і за замовчуванням входить в більшість дистрибутивів Linux, а значить є на більшості серверів «з коробки».
* PEP — єдиний стандарт для написання коду, що робить код підтримуваним і читабельним навіть при переході від одного програміста до іншого. Це підтримує популярність Python.
* Open Source — код інтерпретатора Python є відкритим, що дозволяє будь-кому, хто зацікавлений у розвитку мови взяти участь в його розробці і поліпшити його. Якщо дивитися деталі релізу однією з версій мови, то можна помітити, що величезні частини нового функціоналу реалізовані сторонніми розробниками.

Всі ці переваги мови зробили його популярним і затребуваним на даний момент, дозволивши Python розвиватися величезними темпами. Існує вже третя версія мови, яка є основною сьогодні. Друга версія мови перестала підтримуватися в грудні 2019-го року.

## ***Особливості роботи в середовищі Python***

Мова Python має багато спільного з такими мовами як Perl, C та Java. Однак, є і деякі певні відмінності.

Існує два основних способи запустити програму, написану на мові Python.

Інтерактивний інтерпретатор, який поставляється разом з Python, дає можливість експериментувати з невеликими програмами. Вводячи команди рядок за рядком і миттєво отримуючи результат кожної з них.

Проте, як правило програми містять дуже велику кількість рядків коду, тож їх зберігають у вигляді текстових файлів з розширенням .py, а потім запускають.

Працювати в інтерактивному режимі можна в консолі. Для цього слід виконати команду python. Запуститься інтерпретатор, де спочатку виведеться інформація про інтерпретатор. Далі, послідує запрошення до вводу (>>>).

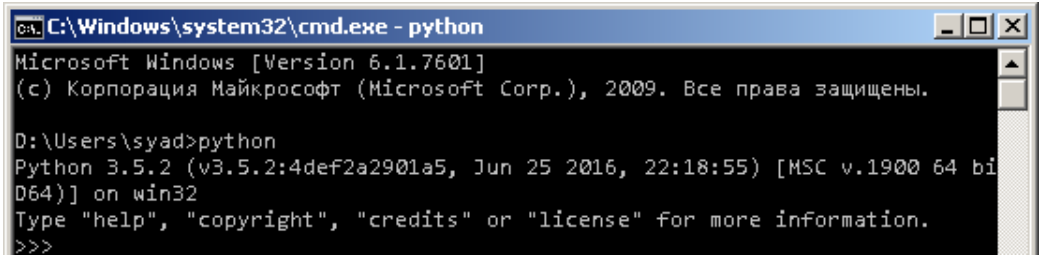


Рисунок 1 Консоль. Запуск інтерпритатора

Такий режим можна використовувати для вводу інформації або для розрахунків. Вводиться команда та натискається клавіша Enter (завершення вводу команди). Після чого відразу з’являється відповідь. Автоматичне виведення значення – це особливість інтерактивного інтерпретатора, що економить час.

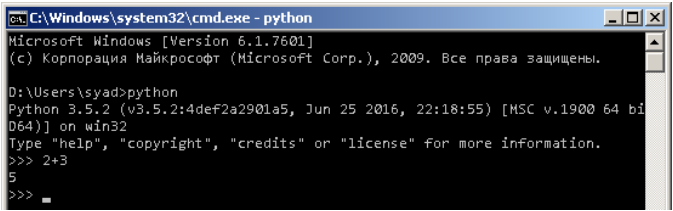


Рисунок 2 Консоль. Виконання простих математичних функцій

Буває, що в процесі уведення була допущена помилка або потрібно повторити раніше використовувану команду. Щоб не писати рядок спочатку, в консолі можна прокручувати список команд, використовуючи для цього стрілки на клавіатурі.

Незважаючи на зручності інтерактивного режиму роботи при написанні програм на Python, звичайно потрібно зберігати вихідний програмний код для подальшого використання. В такому випадку зберігаються файли, які передаються потім інтерпретатору на виконання. В інтерпретованих мовах програмування, часто вихідний код називають скриптом.

Підготувати скрипти можна у вбудованому середовищі IDLE або будь-якому редакторі редакторі. Крім того, існують спеціальні програми для розробки.

Запускати підготовлені файли можна в IDLE та в консолі за допомогою команди python ***адреса\ім’я\_файлу.py***.

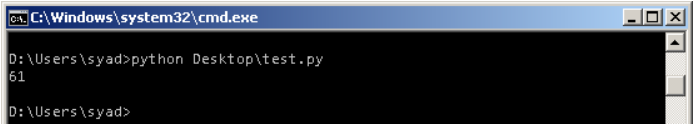


Рисунок 3 Результат виконання команди, збереженої у файлі, в консолі

# РОЗДІЛ 2. Розробка індивідуального завдання

## ***2.1. Постановка індивідуального завдання***

Відповідно до обраного індивідуального завдання потрібно написати консольну програму, яка зчитує файл і виводить частоту використання різних слів починаючи від найбільш вживаних слів закінчуючи найменш вживаними словами.

## ***2.2. Робота з файлами***

Великі обсяги даних має сенс зберігати не в списках або словниках, а в файлах. Тому в мовах програмування передбачена можливість роботи з файлами. В Python файли розглядаються як об'єкти файлових класів, тобто, наприклад,текстовий файл - це тип даних поряд з типами списку, словника, цілого числа і ін.

Зазвичай файли ділять на текстові і байтові (бінарні). Перші розглядаються як містять символьні дані, рядки. Другі - як потік байтів. Побайтово зчитуються, наприклад, файли зображень.

Робота з бінарними файлами трохи складніше. Нерідко їх обробляють за допомогою спеціальних модулів Python (pickle, struct). Розглянемо базові прийоми читання текстових файлів.

Режими:

* ***r***  -   тільки читання (значення за замовчуванням). Після відкриття файлу покажчик встановлюється на початок файлу. Якщо файл не існує, то збуджується виключення IOError.
* ***r+***  -   читання і запис. Після відкриття файлу покажчик встановлюється на початок файлу. Якщо файл не існує, то збуджується виключення IOError.
* ***w***   -  запис. Якщо файл не існує, то він буде створений. Якщо файл існує, то він буде перезаписаний. Після відкриття файлу покажчик встановлюється на початок файлу.
* ***w+***  -   читання і запис. Якщо файл не існує, то він буде створений. Якщо файл існує, то він буде перезаписаний. Після відкриття файлу покажчик встановлюється на початок файлу.
* ***a***   -    запис. Якщо файл не існує, то він буде створений. Запис здійснюється у кінець файлу. Вміст файлу не видаляється.
* ***a+***  -   читання і запис. Якщо файл не існує, то він буде створений. Запис здійснюється у кінець файлу. Вміст файлу не видаляється.

Відкриття файлу виконується за допомогою вбудованої в Python функції***open()***. Зазвичай їй передають один або два аргументи.

Перший - ім'я файлу або ім'я з адресою, якщо файл знаходиться не в тому каталозі, де знаходиться скрипт. Другий аргумент - режим, в якому відкривається файл.

Зазвичай використовуються режими читання ( **'r'**) і запису ( '**w**').

Якщо файл відкритий в режимі читання, то запис в нього неможлива. Можна тільки зчитувати дані з нього.

Якщо файл відкритий в режимі запису, то в нього можна тільки записувати дані, зчитувати не можна.

За допомогою файлового методу***read ()*** можна прочитати файл цілком або тільки певну кількість байт. На рисунку 4 наведено який запис і метод був використаний при написанні програми.

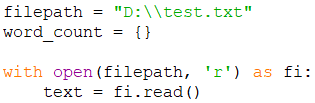


Рисунок 4 Відкриття файлу для читання

## ***2.3. Робота з вмістом файлу***

Для того щоб була можливість порахувати кількість повторення слів у тексті файлу, необхідно очистити його від знаків пунктуації, зайвих пробілів і зробити так щоб всі слова були записані малими літерами. Для даних дій в Python є команда strip\_punctuation(). Синтаксис її використання в програмі набедено на рисунку 5.

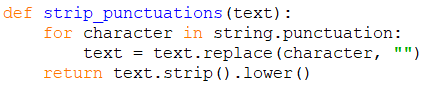


Рисунок 5

Для роботи з множинами в Python створено клас ***set().*** Множини в Python – це структури даних, котрі можуть містити невпорядковані елементи. Також елементи не є індексованими. Як і список, множина дозволяє внесення та вилучення елементів. Проте, є ряд відмінностей, які відрізняють мноину від інших структур даних:

* Множина не містить елементів, які повторюються.
* Елементи множини є незмінними (їх неможна міняти), але сама по собі множина є змінною, її можна міняти.
* Так як елементи не індексуються, множина не підтримує жодних операцій зрізу та індексування.

Тому з допомогою даного класу вдається сформувати список унікальних слів у тексі відкритого документа. Синтаксис наведено на рисунку 6.



Рисунок 6

## ***2.4. Сортування отриманих даних з файлу***

Для сортування отриманого списку використовується функція sorted(). Вона повертає новий відсортований список. По замовчуванню вона відсортовує його в порядку зростання.

Сортування рядків відбувається за таким синтаксисом (рис. 7):



Рисунок 7

В даному записі iterable – це рядок або список значень; key – є необов’язковим параметром, якщо вказувати ключ, то сортування буде виконане по функції цього ключа; reverse – по замовчуванню сортування відбувається по зростанню. Якщо вказати reverse = True то сортування буде здійснено в порядку спадання.

Параметр key може бути: вбудованою функцією, функцією яку оприділив користувач, лямбда-функцією.

При написанні програми, для сортування було здійснено з використанням параметру key як лямбда-функція. Синтаксис коду наведено на рисунку 8.



Рисунок 8

Виведення отриманих результатів здійснювало через команду ***print(),*** яка може виводити не тільки значення змінних, але і значення будь-яких виразів. Запис даної команди наведений на рисунку 9.



Рисунок 9

Відповідно до даного запису результат виконання програми буде виводить на екран користувачу наступним чином (рис. 10):

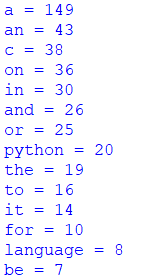


Рисунок 10 Результат виконання програми

Лістинг написаної програми можна переглянути в ДОДАТКУ 1.

# ВИСНОВКИ

Виробнича практика є джерелом накопичення досвіду роботи по вибраній спеціальності та закріплення теоретичних знань. Вона відіграє важливу роль у підготовці спеціалістів високого рівня, які володіли б сучасними технологіями, вміло використовували методологію та методи розробки різного програмного забезпечення за допомогою математичних знань.

Також практика дозволяє реалізувати себе як особистість, вчить самостійності, ініціативності, сприяє набуттю організаційних, дослідницьких навичок, які забезпечують роботу студентів за фахом.

Під час проходження виробничої практики вдалося поглибити свої знання в програмуванні, зокрема при використанні мови Python, виконуючи поставлене індивідуальне завдання від бази практики.

Також завдяки практиці ознайомилася із організаційною структурою ТзОВ ЕЛЕКС, навчилася швидко та якісно виконувати поставлені перед мною завдання, прагнути до вдосконалення своєї роботи та здобула великий багаж нових знань та навичок, який знадобиться для подальшої самореалізації у житті.

Внаслідок зробленої роботи можна зробити висновок, що програмування завжди актуальне та потрібне, а мова програмування Python володіє досить широким функціоналом, потрібним для зручної та якісної роботи.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Head First Python. Легкий для сприйняття довідник, П. Берри, 2021 р., 624 с.
2. Python для всех, Ч. Р. Северанс, 2021 р., 262 с.
3. Научное программирование на Python: ДМК Прес, К. Хилл, 2021 р., 646 с.
4. Простой Python. Современный стиль программирования. 2-е издание: Пытер Прес, Б. Любанович, 2021 р., 592 с.

# ДОДАТОК 1

import string

def strip\_punctuations(text):

for character in string.punctuation:

text = text.replace(character, "")

return text.strip().lower()

filepath = "D:\\test.txt"

word\_count = {}

with open(filepath, 'r') as fi:

text = fi.read()#зчитуємо весь зміст файлу

text = strip\_punctuations(text) #очищуємо від розділових знаків, зайвих пробілів і щоб літери були всі малі

unique\_words = set(text.split()) #формуємо список унікальних слів у тексті

for word in unique\_words:

word\_count[word] = text.count(word) #записуємо у словник слово та його кількість у тексті

sorted\_word\_count = sorted(word\_count.items(), key=lambda x:x[1], reverse=True) #сортуємо

for (word, count) in sorted\_word\_count:

print(word, '=' ,count)