

Міністерство освіти та науки України
Національний університет водного господарства та природокористування

Навчально-науковий інститут кібернетики,
інформаційних технологій та інженерії
Кафедра комп'ютерних наук та прикладної математики

04-01-82М

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторної роботи на тему
«Читання, відображення та збереження зображень
за допомогою функцій бібліотеки OpenCV»
у Jupyter Notebook мовою програмування Python
з освітньої компоненти «**Цифрова обробка зображень**»
для здобувачів вищої освіти другого (магістерського) рівня
за освітньо-професійною програмою «Прикладна математика»
спеціальності 113 «Прикладна математика» та
за освітньо-професійною програмою «Прикладна інформатика»
спеціальності 122 «Комп'ютерні науки»
денної та заочної форм навчання

Рекомендовано науково-методичною
радою з якості ННІ КІТІ
Протокол № 7 від 17.06.2024 р.

Рівне – 2024

Методичні вказівки до виконання лабораторної роботи на тему «Читання, відображення та збереження зображень за допомогою функцій бібліотеки OpenCV» у Jupyter Notebook мовою програмування Python з освітньої компоненти «Цифрова обробка зображень» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Прикладна математика» спеціальності 113 «Прикладна математика» та за освітньо-професійною програмою «Прикладна інформатика» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання. [Електронне видання] / Климюк Ю. Є. – Рівне : НУВГП, 2024. – 101 с.

Укладач: Климюк Юрій Євгенійович, кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук та прикладної математики ННІ КІТІ НУВГП.

Відповідальний за випуск: Турбал Юрій Васильович, доктор технічних наук, професор, завідувач кафедри комп'ютерних наук та прикладної математики ННІ КІТІ НУВГП.

Керівник групи забезпечення для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Прикладна математика» спеціальності 113 «Прикладна математика»:

Климюк Юрій Євгенійович,, кандидат технічних наук, доцент кафедри комп'ютерних наук та прикладної математики ННІ КІТІ НУВГП.

Керівник групи забезпечення для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Прикладна інформатика» спеціальності 122 «Комп'ютерні науки»:

Мічута Ольга Романівна, кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук та прикладної математики ННІ КІТІ НУВГП.

Рекомендовано кафедрою комп'ютерних наук та прикладної математики ННІ КІТІ НУВГП (протокол №11 від 10.06.2024 р.)

Попередня версія методичних вказівок 04-01-69М.

© Ю. Є. Климюк, 2024

© НУВГП, 2024

Лабораторна робота

Тема: Читання, відображення та збереження зображень за допомогою функцій бібліотеки OpenCV.

Мета: Ознайомитися із читанням, відображенням та збереженням зображень за допомогою функцій бібліотеки OpenCV, оволодіння сучасними методами проектування та розробки комп'ютерних програм для читання, відображення та збереження зображень у Jupyter Notebook мовою програмування Python за допомогою функцій бібліотеки OpenCV.

Література:

1. Download the latest version for Windows. Python 3.12.4. URL: <https://www.python.org/downloads/>

2. Методичні вказівки до виконання лабораторної роботи на тему «Читання, відображення та збереження зображень за допомогою функцій бібліотеки OpenCV» у Jupyter Notebook мовою програмування Python з освітньої компоненти «Цифрова обробка зображень» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Прикладна математика» спеціальності 113 «Прикладна математика» і за освітньо-професійною програмою «Прикладна інформатика» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання [Електронне видання] / Ю.Є. Климюк. Рівне : НУВГП, 2022. 88 с.

3. Chung Bryan WC. Pro Processing for Images and Computer Vision with OpenCV. Apress, 2017. 301 p.

4. Rosebrock Adrian. Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision. PyImageSearch, 2016. 166 p.

5. Szeliski Richard. Computer Vision: Algorithms and Applications. 2nd Edition. Springer, 2021. 1230 p.

6. Villán Alberto Fernández. Mastering OpenCV 4 with Python : A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. Packt Publishing : Birmingham – Mumbai, 2019. 650 p.

Хід роботи

I. Виконайте наступні завдання:

Завдання 1. Виконайте інсталяцію останньої версії мови програмування Python3 (на момент формування документу актуальна версія 12.4).

Завдання 2. Виконайте інсталяцію віртуального середовища virtualenv.

Завдання 3. Виконайте інсталяцію останньої версії бібліотеки OpenCV та інсталюйте середовище розробки Jupyter Notebook.

Завдання 4. Розробіть у Jupyter Notebook мовою програмування Python комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як ϵ (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створюватиме вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 5. Розробіть у Jupyter Notebook мовою програмування Python комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме з поточної папки деяке зображення як ϵ (з альфа-каналом) з файлу, що задається як параметр командного рядка, створюватиме вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 6. Розробіть у Jupyter Notebook мовою програмування Python комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу з ім'ям "image.webp" з поточної папки, створюватиме вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 7. Розробіть у Jupyter Notebook мовою програмування Python комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.webp" з поточної папки, зберігатиме його файлі з ім'ям "image_grayscale.webp" у поточній папці, створюватиме вікно з ім'ям "Grayscale image", дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 8. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.webp" з поточної папки, зберігатиме його файлі з ім'ям "image_grayscale.webp" у поточній

папці, створюватиме вікно з ім'ям “Grayscale image”, дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 9 для самостійної роботи по варіантах.

1. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

2. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

3. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого та розміром 1/2 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

4. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/2 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці

з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

5. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/4 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

6. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/4 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

7. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/8 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

8. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/8 з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці

з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

9. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканального у відтінках сірого з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

10. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

11. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого та розміром 1/2 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

12. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/2 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у

поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

13. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/4 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

14. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/4 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

15. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/8 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

16. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/8 з файлу з ім'ям “image.png” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.png” у поточній папці

з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

17. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканального у відтінках сірого з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

18. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

19. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого та розміром 1/2 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

20. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/2 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній

папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

21. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/4 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

22. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/4 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

23. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/8 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

24. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/8 з файлу з ім'ям “image.webp” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній

папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

25. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як ϵ (з альфа-каналом) з файлу з ім'ям “image.jpg” з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Завдання 10 для самостійної роботи по варіантах.

1. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканального у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

2. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.jpg” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

3. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як

однокальне у відтінках сірого та розміром $1/2$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

4. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром $1/2$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

5. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром $1/4$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

6. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром $1/4$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні

зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

7. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/8 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

8. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/8 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

9. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканального у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

10. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу, що задається як параметр командного рядка, з

поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

11. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/2 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

12. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/2 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з файлу з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

13. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/4 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

14. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/4 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

15. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/8 з файлу з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

16. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/8 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.png" у поточній папці з рівнем стиснення від 0 до 9, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

17. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканального у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.webp" у поточній папці з якістю зображення від 0 до 100,

створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

18. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

19. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром 1/2 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

20. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром 1/2 з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям “image_resulting.webp” у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям “Resulting image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

21. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як

однокальне у відтінках сірого та розміром $1/4$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.webp" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіши.

22. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром $1/4$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.webp" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіши.

23. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як однокальне у відтінках сірого та розміром $1/8$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.webp" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відобразатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіши.

24. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове та розміром $1/8$ з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.webp" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразатиме у цьому вікні

зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

25. Розробіть у Jupyter Notebook мовою програмування Python з використанням об'єктно-орієнтованого підходу комп'ютерну програму, що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, зберігатиме його у файлі з ім'ям "image_resulting.jpg" у поточній папці з якістю зображення від 0 до 100, створюватиме вікно з ім'ям "Resulting image", автоматично дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Хід виконання

Пояснення щодо виконання завдання 1

Крок 1. Скачайте файл інсталлятора Python 3 (остання версія на момент підготовки 12.4, можна скачати новішого). Для цього перейдіть за посиланням: <https://www.python.org/downloads/> і натисніть на кнопку «Download Python 3.12.4» (рис. 1). При цьому має скачатися файл python-3.12.4-amd64.exe.

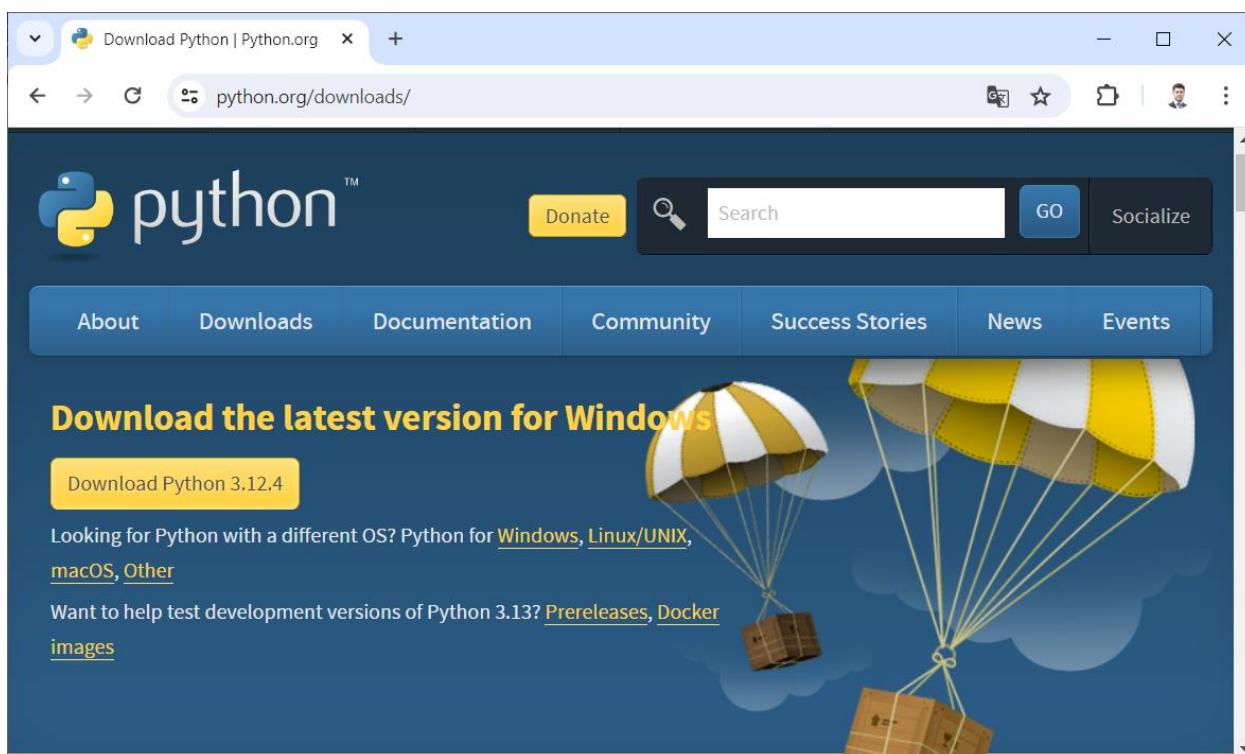


Рис. 1. Сторінка для скачування інсталлятора Python 3 версії 12.4

Крок 2. Запустіть на виконання скачаний файл інсталятора python-3.12.4-amd64.exe, щоб розпочати процес встановлення мови програмування Python 3 (рис. 2–4). У стартовому вікні обов’язково встановить галочку add path (рис. 2).

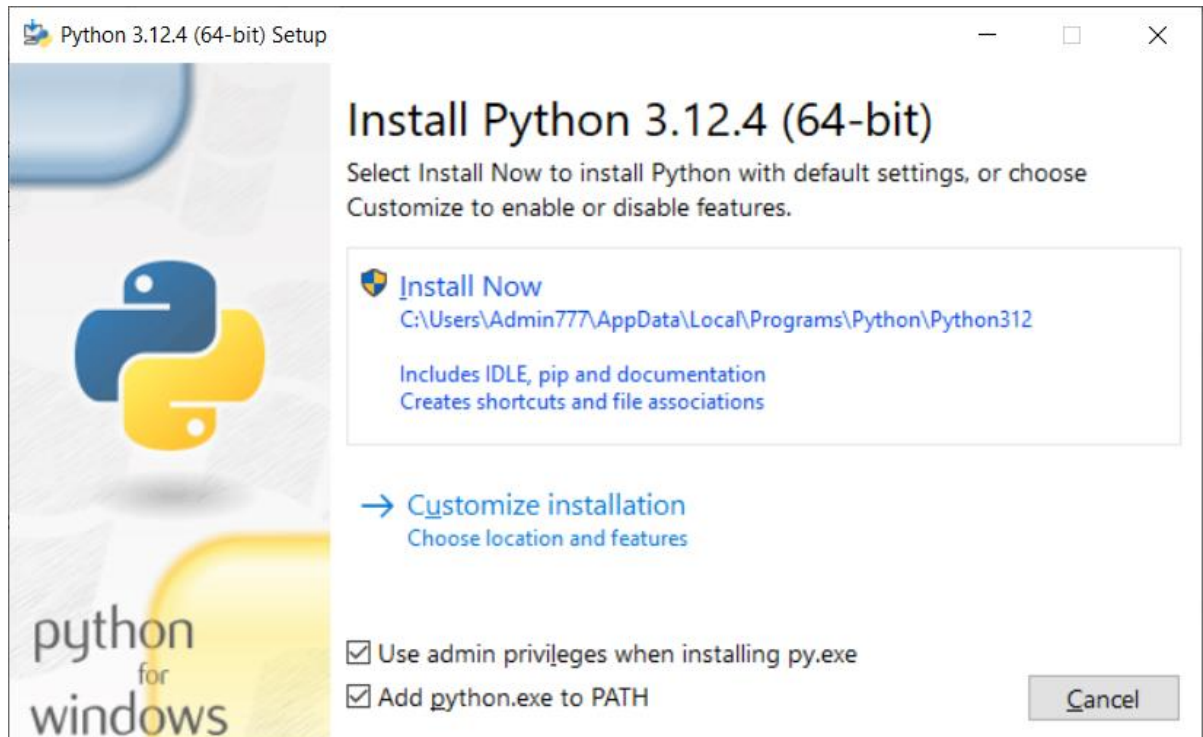


Рис. 2. Стартове вікно інсталятора Python 3 версії 12.4

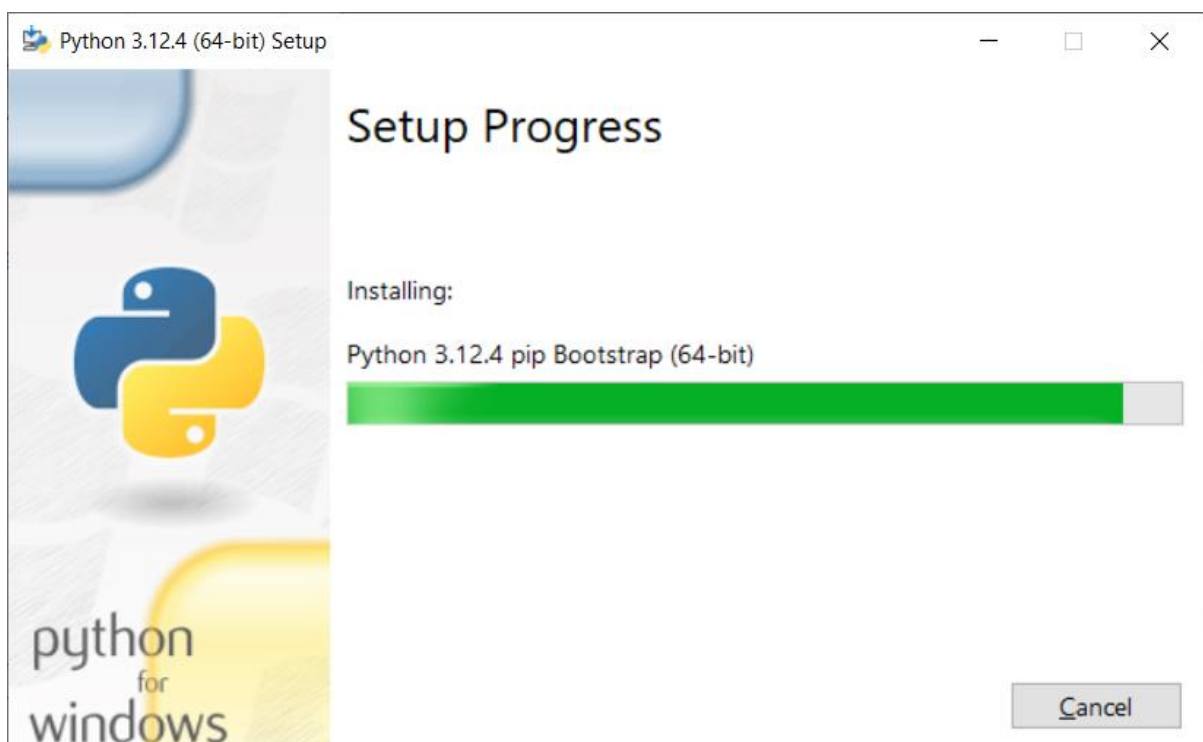


Рис. 3. Інсталяція бібліотек та різних утиліт Python 3 версії 12.4

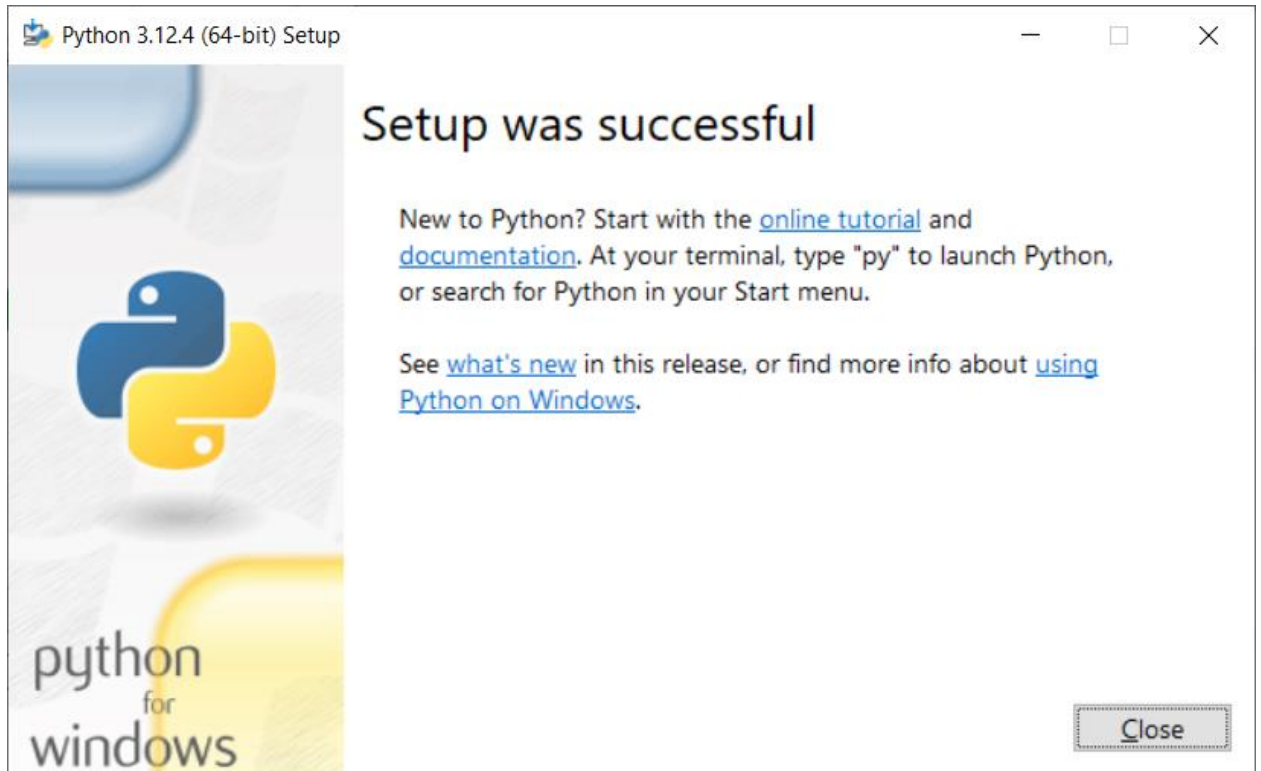


Рис. 4. Завершення інсталяції Python 3 версії 12.4

Крок 3. Для перевірити коректної інсталяції Python 3 виконайте наступні команди (рис. 5):

```
Start -> Run... -> cmd  
python  
print("Це Python!")  
exit()
```

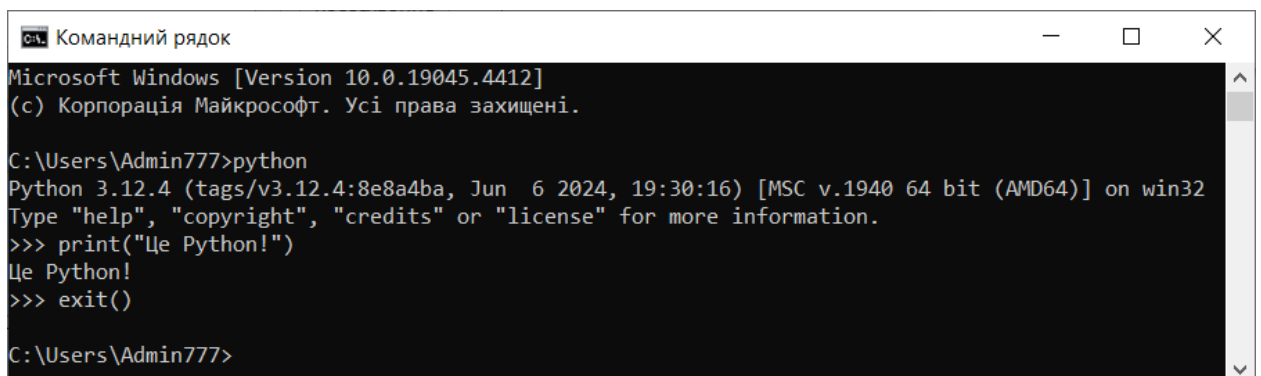
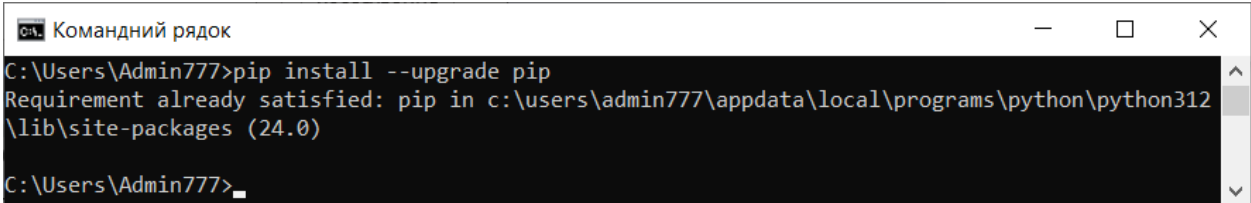


Рис. 5. Перевірка запуску і виконання команд Python 3 версії 12.4

Пояснення щодо виконання завдання 2

Крок 1. Віртуальне середовище `virtualenv` інстальватимемо за допомогою системи керування пакетами `pip`, яка використовується для встановлення та керування програмними пакетами/бібліотеками, написаними на Python. Він інстальюється разом з Python, але досить часто його потрібно оновлювати командою (рис. 6):

```
pip install --upgrade pip
```

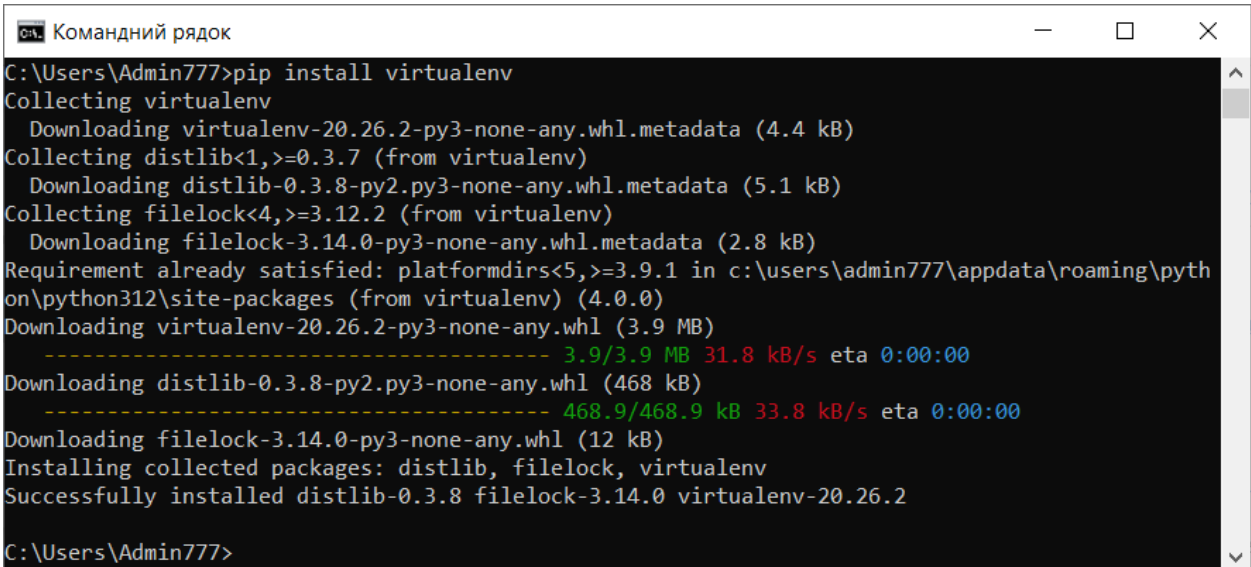


```
cmd. Командний рядок
C:\Users\Admin777>pip install --upgrade pip
Requirement already satisfied: pip in c:\users\admin777\appdata\local\programs\python\python312\lib\site-packages (24.0)
C:\Users\Admin777>
```

Рис. 6. Оновлення системи керування пакетами `pip`

Крок 2. Інстальюємо віртуальне середовище `virtualenv` (рис. 7), яке потрібне для того, щоб для кожного окремого проекту була своя “кімната” зі своїми версіями встановлених бібліотек, які не залежатимуть від інших проектів та плутатимуться між собою:

```
pip install virtualenv
```

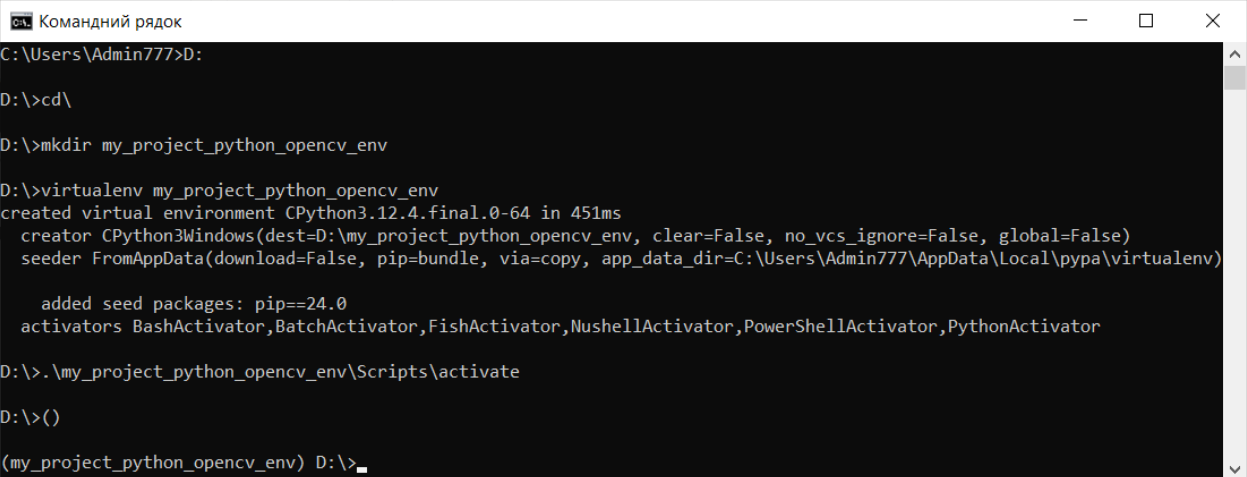


```
cmd. Командний рядок
C:\Users\Admin777>pip install virtualenv
Collecting virtualenv
  Downloading virtualenv-20.26.2-py3-none-any.whl.metadata (4.4 kB)
Collecting distlib<1,>=0.3.7 (from virtualenv)
  Downloading distlib-0.3.8-py2.py3-none-any.whl.metadata (5.1 kB)
Collecting filelock<4,>=3.12.2 (from virtualenv)
  Downloading filelock-3.14.0-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: platformdirs<5,>=3.9.1 in c:\users\admin777\appdata\roaming\python\python312\site-packages (from virtualenv) (4.0.0)
Downloading virtualenv-20.26.2-py3-none-any.whl (3.9 MB)
----- 3.9/3.9 MB 31.8 kB/s eta 0:00:00
Downloading distlib-0.3.8-py2.py3-none-any.whl (468 kB)
----- 468.9/468.9 kB 33.8 kB/s eta 0:00:00
Downloading filelock-3.14.0-py3-none-any.whl (12 kB)
Installing collected packages: distlib, filelock, virtualenv
Successfully installed distlib-0.3.8 filelock-3.14.0 virtualenv-20.26.2
C:\Users\Admin777>
```

Рис. 7. Інсталяція віртуального середовища `virtualenv`

Крок 3. Виконуємо послідовно наступні команди для створення папки проекту, налаштування відповідного віртуального середовища та його активації (рис. 8):

```
D:
cd\
mkdir my_project_python_opencv_env
virtualenv my_project_python_opencv_env
.\my_project_python_opencv_env\Scripts\activate
```



```
Командний рядок
C:\Users\Admin777>D:
D:\>cd\
D:\>mkdir my_project_python_opencv_env
D:\>virtualenv my_project_python_opencv_env
created virtual environment CPython3.12.4.final.0-64 in 451ms
creator CPython3Windows(dest=D:\my_project_python_opencv_env, clear=False, no_vcs_ignore=False, global=False)
seeder FromAppData(download=False, pip=bundle, via=copy, app_data_dir=C:\Users\Admin777\AppData\Local\pypa\virtualenv)

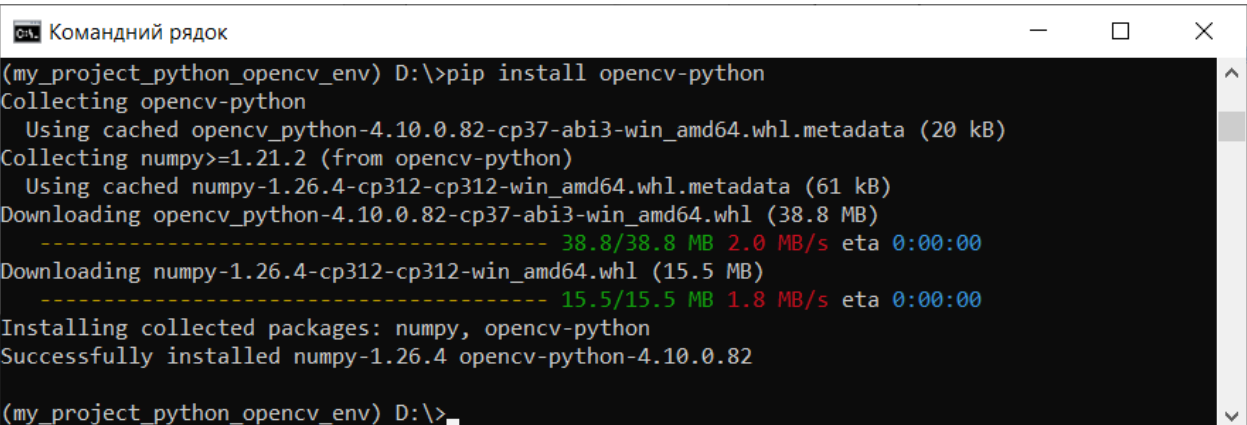
added seed packages: pip==24.0
activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
D:\>.\my_project_python_opencv_env\Scripts\activate
D:\>()
(my_project_python_opencv_env) D:\>_
```

Рис. 8. Створення папки проекту і віртуального середовища та його активація

Пояснення щодо виконання завдання 3

Крок 1. Інсталюємо пакети `opencv-python` та `numpy`, які знадобляться для тестування функцій бібліотеки `OpenCV`, командою (рис. 9):

```
pip install opencv-python
```



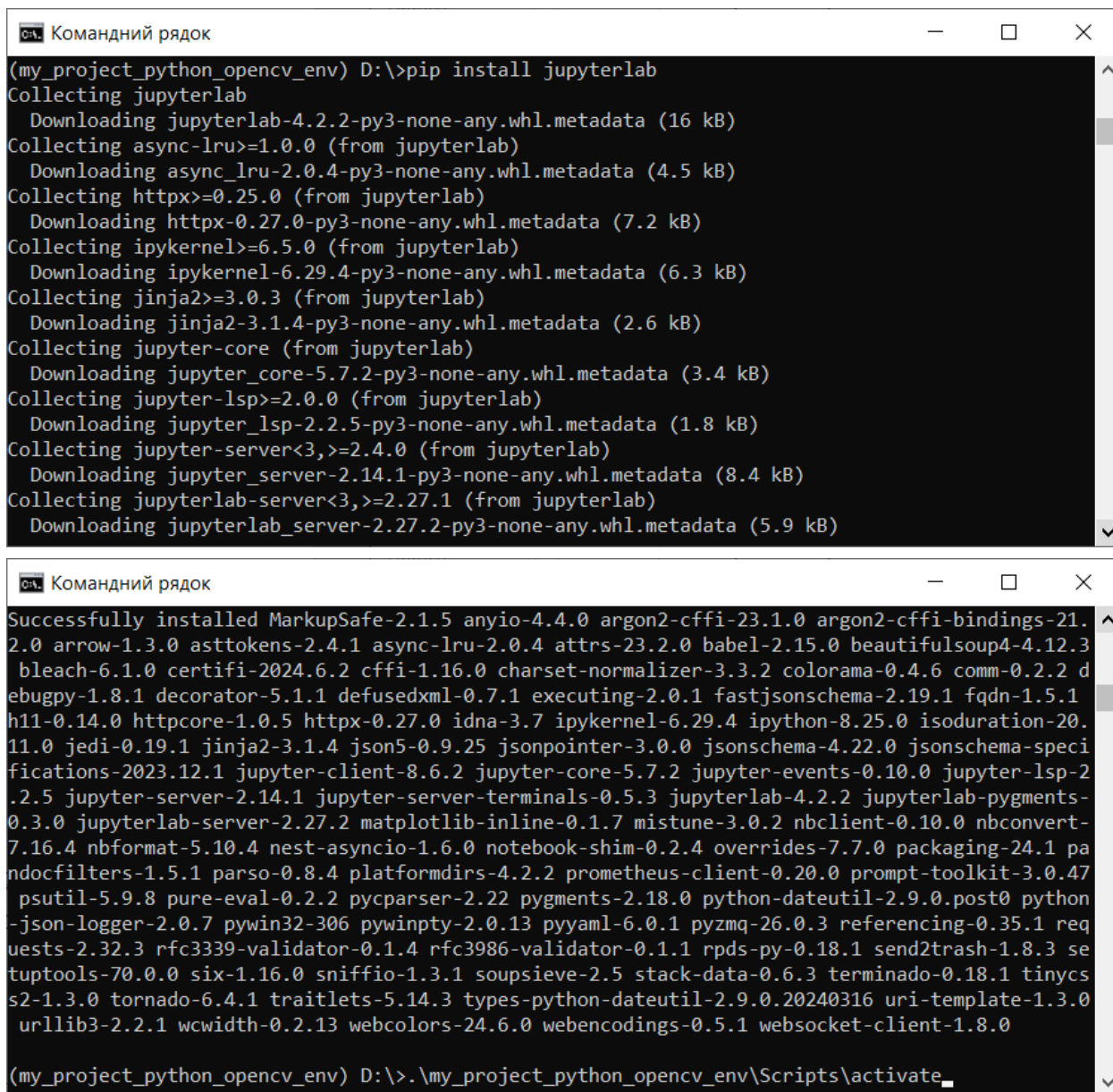
```
Командний рядок
(my_project_python_opencv_env) D:\>pip install opencv-python
Collecting opencv-python
  Using cached opencv_python-4.10.0.82-cp37-abi3-win_amd64.whl.metadata (20 kB)
Collecting numpy>=1.21.2 (from opencv-python)
  Using cached numpy-1.26.4-cp312-cp312-win_amd64.whl.metadata (61 kB)
Downloading opencv_python-4.10.0.82-cp37-abi3-win_amd64.whl (38.8 MB)
----- 38.8/38.8 MB 2.0 MB/s eta 0:00:00
Downloading numpy-1.26.4-cp312-cp312-win_amd64.whl (15.5 MB)
----- 15.5/15.5 MB 1.8 MB/s eta 0:00:00
Installing collected packages: numpy, opencv-python
Successfully installed numpy-1.26.4 opencv-python-4.10.0.82
(my_project_python_opencv_env) D:\>_
```

Рис. 9. Інсталяція пакетів `opencv-python` та `numpy`

Крок 2. Для використання середовища розробки `Jupyter Notebook` виконаємо інсталяцію пакету `jupyterlab` командою (рис. 10):

```
pip install jupyterlab
```

У середовищі розробки Jupyter Notebook можна розробляти, документувати та виконувати програми на мові Python. Середовище складається з двох компонентів: веб-додаток, що запускається в браузері, та ноутбуки – файли, в яких можна працювати з вихідним кодом програми, запускати його, вводити та виводити дані тощо.



```
(my_project_python_opencv_env) D:\>pip install jupyterlab
Collecting jupyterlab
  Downloading jupyterlab-4.2.2-py3-none-any.whl.metadata (16 kB)
Collecting async-lru>=1.0.0 (from jupyterlab)
  Downloading async_lru-2.0.4-py3-none-any.whl.metadata (4.5 kB)
Collecting httpx>=0.25.0 (from jupyterlab)
  Downloading httpx-0.27.0-py3-none-any.whl.metadata (7.2 kB)
Collecting ipykernel>=6.5.0 (from jupyterlab)
  Downloading ipykernel-6.29.4-py3-none-any.whl.metadata (6.3 kB)
Collecting jinja2>=3.0.3 (from jupyterlab)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting jupyter-core (from jupyterlab)
  Downloading jupyter_core-5.7.2-py3-none-any.whl.metadata (3.4 kB)
Collecting jupyter-lsp>=2.0.0 (from jupyterlab)
  Downloading jupyter_lsp-2.2.5-py3-none-any.whl.metadata (1.8 kB)
Collecting jupyter-server<3,>=2.4.0 (from jupyterlab)
  Downloading jupyter_server-2.14.1-py3-none-any.whl.metadata (8.4 kB)
Collecting jupyterlab-server<3,>=2.27.1 (from jupyterlab)
  Downloading jupyterlab_server-2.27.2-py3-none-any.whl.metadata (5.9 kB)

(my_project_python_opencv_env) D:\>.\my_project_python_opencv_env\Scripts\activate_
Successfully installed MarkupSafe-2.1.5 anyio-4.4.0 argon2-cffi-23.1.0 argon2-cffi-bindings-21.
2.0 arrow-1.3.0 asttokens-2.4.1 async-lru-2.0.4 attrs-23.2.0 babel-2.15.0 beautifulsoup4-4.12.3
bleach-6.1.0 certifi-2024.6.2 cffi-1.16.0 charset-normalizer-3.3.2 colorama-0.4.6 comm-0.2.2 d
ebuggy-1.8.1 decorator-5.1.1 defusedxml-0.7.1 executing-2.0.1 fastjsonschema-2.19.1 fqdn-1.5.1
h11-0.14.0 httpcore-1.0.5 httpx-0.27.0 idna-3.7 ipykernel-6.29.4 ipython-8.25.0 isoduration-20.
11.0 jedi-0.19.1 jinja2-3.1.4 json5-0.9.25 jsonpointer-3.0.0 jsonschema-4.22.0 jsonschema-speci
fications-2023.12.1 jupyter-client-8.6.2 jupyter-core-5.7.2 jupyter-events-0.10.0 jupyter-lsp-2
.2.5 jupyter-server-2.14.1 jupyter-server-terminals-0.5.3 jupyterlab-4.2.2 jupyterlab-pygments-
0.3.0 jupyterlab-server-2.27.2 matplotlib-inline-0.1.7 mistune-3.0.2 nbclient-0.10.0 nbconvert-
7.16.4 nbformat-5.10.4 nest-asyncio-1.6.0 notebook-shim-0.2.4 overrides-7.7.0 packaging-24.1 pa
ndocfilters-1.5.1 parso-0.8.4 platformdirs-4.2.2 prometheus-client-0.20.0 prompt-toolkit-3.0.47
psutil-5.9.8 pure-eval-0.2.2 pycparser-2.22 pygments-2.18.0 python-dateutil-2.9.0.post0 python
-json-logger-2.0.7 pywin32-306 pywinpty-2.0.13 pyyaml-6.0.1 pyzmq-26.0.3 referencing-0.35.1 req
uests-2.32.3 rfc3339-validator-0.1.4 rfc3986-validator-0.1.1 rpds-py-0.18.1 send2trash-1.8.3 se
tuptools-70.0.0 six-1.16.0 sniffio-1.3.1 soupsieve-2.5 stack-data-0.6.3 terminado-0.18.1 tinycs
s2-1.3.0 tornado-6.4.1 traitlets-5.14.3 types-python-dateutil-2.9.0.20240316 uri-template-1.3.0
urllib3-2.2.1 wcwidth-0.2.13 webcolors-24.6.0 webencodings-0.5.1 websocket-client-1.8.0

(my_project_python_opencv_env) D:\>.\my_project_python_opencv_env\Scripts\activate_
```

Рис. 10. Інсталяція jupyterlab

Веб-додаток дозволяє:

- 1) редагувати код мовою програмування Python у браузері з підсвічуванням синтаксису, автовідступами та автодоповненням;
- 2) запускати код у браузері;

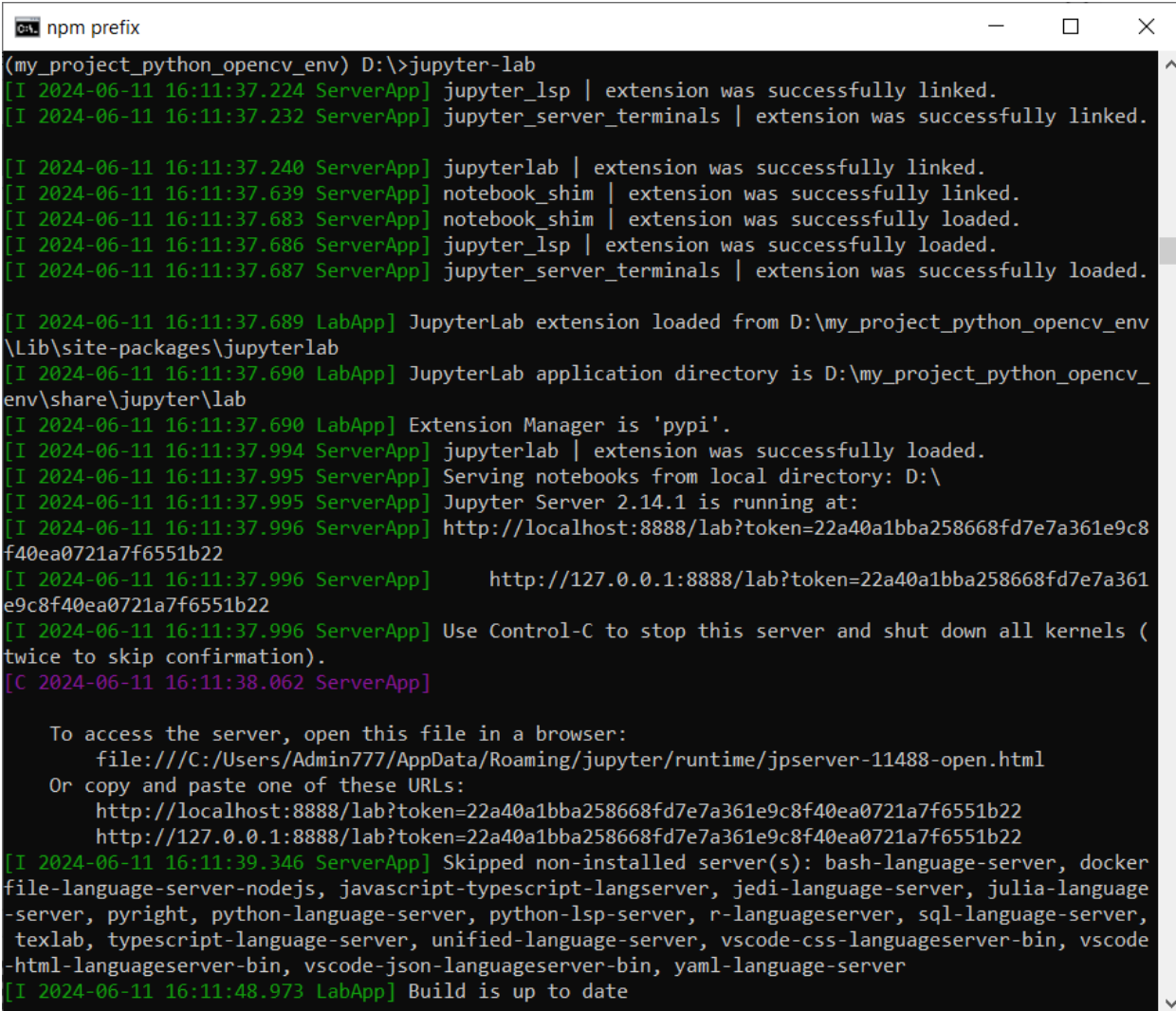
3) відображати результати обчислень із графічним представленням (схеми, графіки).

Ноутбуки – це файли, в яких зберігаються вихідний код, вхідні та вихідні дані, отримані в рамках сесії. Фактично, він є записом роботи користувача, але при цьому дозволяє наново виконати код, який є у ньому. Ноутбуки можна експортувати у документи форматів PDF, HTML.

Jupyter Notebook є графічною оболонкою для IPython, яка розширює ідею консольного підходу до інтерактивних обчислень. IPython дозволяє підключатися клієнтам до одного обчислювального ядра і, завдяки своїй архітектурі, може працювати в паралельному кластері.

Крок 3. Запустимо середовище розробки Jupyter Notebook командою (рис. 11, 12):

```
jupyter-lab
```



```
ca. npm prefix
(my_project_python_opencv_env) D:\>jupyter-lab
[I 2024-06-11 16:11:37.224 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-06-11 16:11:37.232 ServerApp] jupyter_server_terminals | extension was successfully linked.

[I 2024-06-11 16:11:37.240 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-06-11 16:11:37.639 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-06-11 16:11:37.683 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-06-11 16:11:37.686 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-06-11 16:11:37.687 ServerApp] jupyter_server_terminals | extension was successfully loaded.

[I 2024-06-11 16:11:37.689 LabApp] JupyterLab extension loaded from D:\my_project_python_opencv_env
\Lib\site-packages\jupyterlab
[I 2024-06-11 16:11:37.690 LabApp] JupyterLab application directory is D:\my_project_python_opencv_
env\share\jupyter\lab
[I 2024-06-11 16:11:37.690 LabApp] Extension Manager is 'pypi'.
[I 2024-06-11 16:11:37.994 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-06-11 16:11:37.995 ServerApp] Serving notebooks from local directory: D:\
[I 2024-06-11 16:11:37.995 ServerApp] Jupyter Server 2.14.1 is running at:
[I 2024-06-11 16:11:37.996 ServerApp] http://localhost:8888/lab?token=22a40a1bba258668fd7e7a361e9c8
f40ea0721a7f6551b22
[I 2024-06-11 16:11:37.996 ServerApp] http://127.0.0.1:8888/lab?token=22a40a1bba258668fd7e7a361
e9c8f40ea0721a7f6551b22
[I 2024-06-11 16:11:37.996 ServerApp] Use Control-C to stop this server and shut down all kernels (
twice to skip confirmation).
[C 2024-06-11 16:11:38.062 ServerApp]

To access the server, open this file in a browser:
file:///C:/Users/Admin777/AppData/Roaming/jupyter/runtime/jpserver-11488-open.html
Or copy and paste one of these URLs:
http://localhost:8888/lab?token=22a40a1bba258668fd7e7a361e9c8f40ea0721a7f6551b22
http://127.0.0.1:8888/lab?token=22a40a1bba258668fd7e7a361e9c8f40ea0721a7f6551b22
[I 2024-06-11 16:11:39.346 ServerApp] Skipped non-installed server(s): bash-language-server, docker
file-language-server-nodejs, javascript-typescript-langserver, jedi-language-server, julia-language
-server, pyright, python-language-server, python-lsp-server, r-languageserver, sql-language-server,
texlab, typescript-language-server, unified-language-server, vscode-css-languageserver-bin, vscode
-html-languageserver-bin, vscode-json-languageserver-bin, yaml-language-server
[I 2024-06-11 16:11:48.973 LabApp] Build is up to date
```

Рис. 11. Запуск середовища розробки Jupyter Notebook

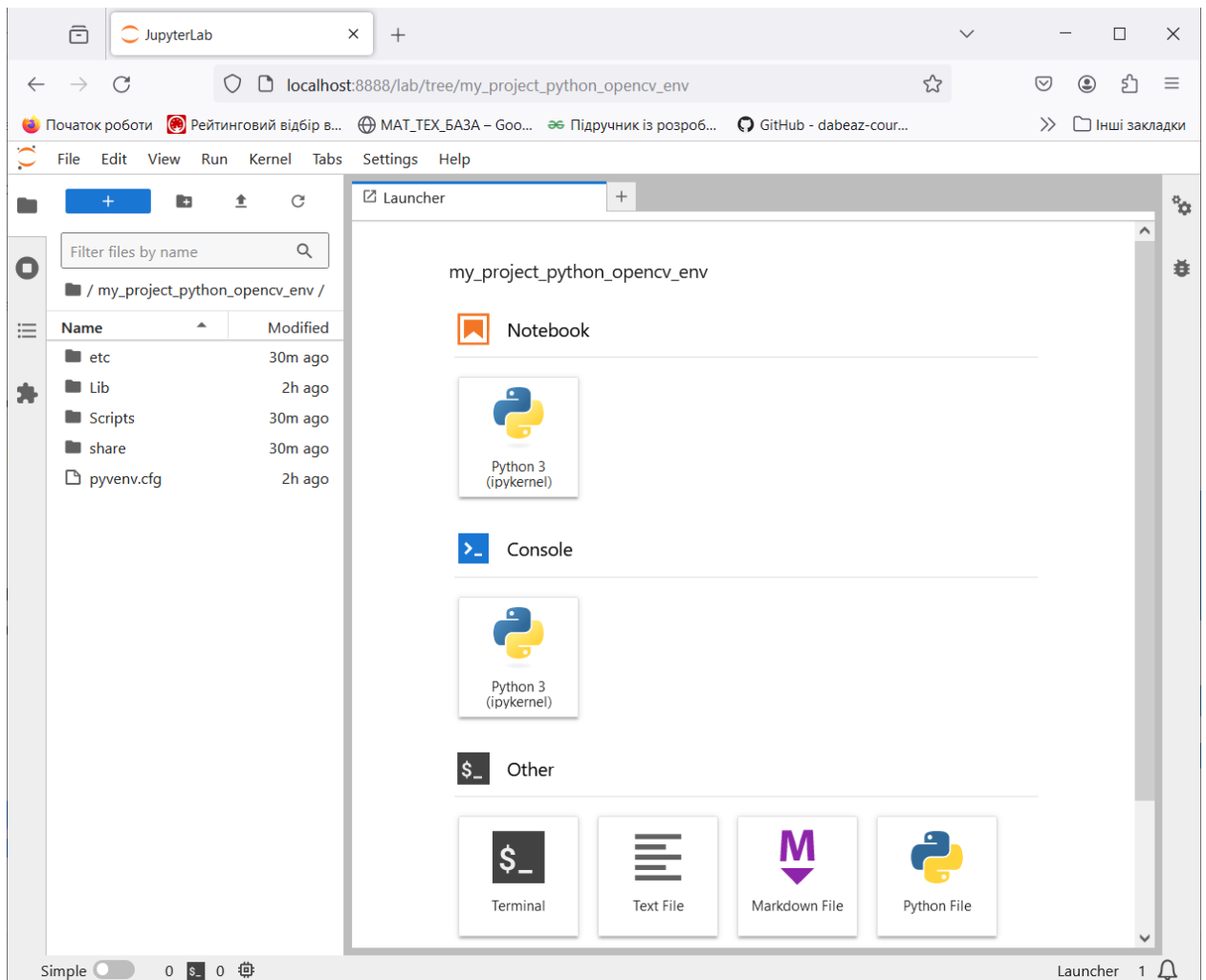


Рис. 12. Середовище розробки Jupyter Notebook

Крок 4. Перевіряємо, чи все у нас працює. У вікні Jupyter Notebook створюємо новий Python 3 (ipykernel) файл (рис. 13).

Код на мові Python або текст у нотації Markdown потрібно вводити до комірок. Якщо це код Python, на панелі інструментів потрібно виставити властивість “Code”. Якщо це текст Markdown – виставити “Markdown”.

Встановлюємо на панелі інструментів властивість “Code”, до комірки блокноту заносимо наступні два рядки:

```
import cv2 as cv
print(cv.__version__)
```

і запускаємо на виконання, натиснувши на кнопку «Run» (рис. 14).

Якщо у вікні Jupyter Notebook виводиться версія бібліотеки OpenCV (рис. 15), то все гаразд, можна приступати до виконання наступних завдань.

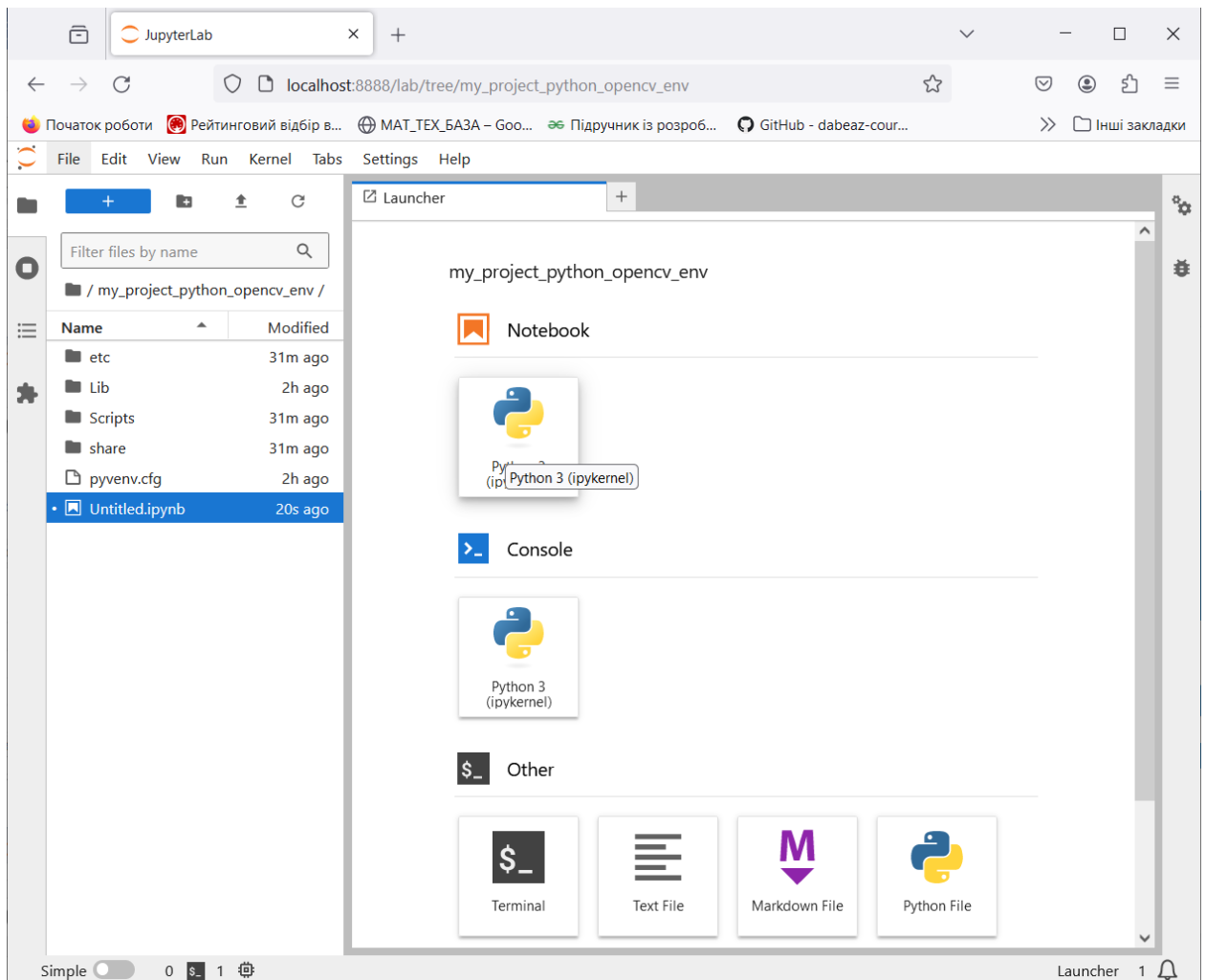


Рис. 13. Створення нового Python 3 (ipykernel) файлу у Jupyter Notebook

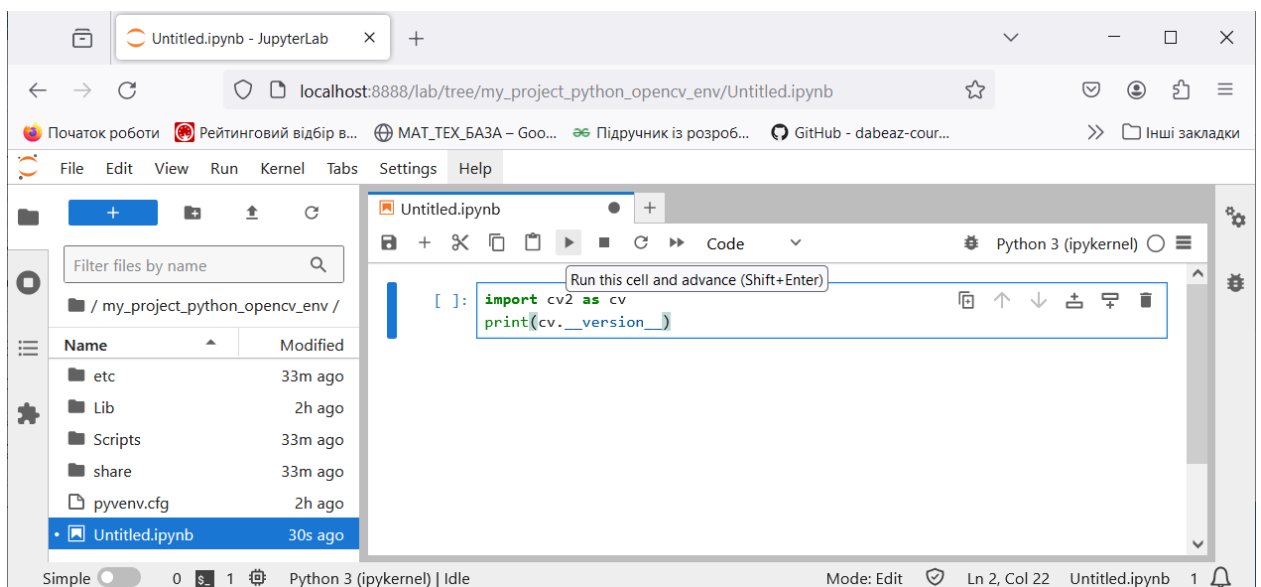


Рис. 14. Формування команд у Jupyter Notebook

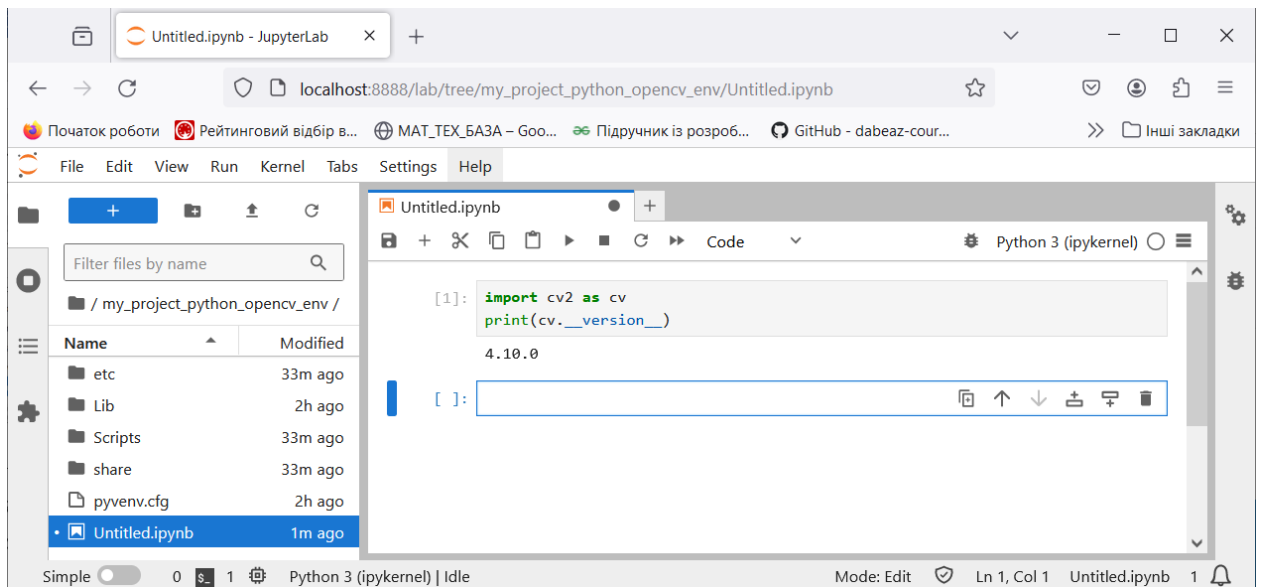


Рис. 15. Запуск на виконання команд у Jupyter Notebook

Пояснення щодо виконання завдань 4-8

Читання, збереження та відображення зображень є основою для цифрової обробки зображень та комп'ютерного зору. Навіть під час обрізання, зміни розміру, обертання або застосування різних фільтрів для обробки зображень його спочатку потрібно прочитати. Тому важливо освоїти ці основні операції.

OpenCV – найбільша бібліотека для цифрової обробки зображень та комп'ютерного зору у світі, має ці три вбудовані функції, які мають наступне призначення:

- 1) `imread()` – читання зображення з файлу;
- 2) `imwrite()` – збереження зображення у файлі;
- 3) `imshow()` – відображення зображення у вікні.

Розглянемо спочатку приклад коду комп'ютерної програми (лістинг 1), що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створюватиме вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Лістинг 1. Код комп'ютерної програми

```
# Підключення бібліотеки OpenCV
import cv2
```

```

# Підключення бібліотеки sys
import sys
# Читання зображення як є (з альфа-каналом)
image = cv2.imread("image.webp", -1)
# або
# image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if image is None:
    sys.exit("Could not read the image.")
# Створення вікна для відображення
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
# Відображення у вікні прочитаного зображення
cv2.imshow("My image", image)
# Організація затримки до натиснення довільної клавіші
cv2.waitKey(0)
# Знищення всіх створених вікон
cv2.destroyAllWindows()
# або знищення створеного вікна
# cv2.destroyWindow("My image")

```

Відмітимо, що всі функції OpenCV містяться у просторі імен cv2. Для виклику функцій потрібно для компілятора явно вказати простір імен, добавивши префікс cv2. до імен функцій. Для практичної зручності простору імен cv2 варто призначити ім'я cv (лістинг 2).

Лістинг 2. Код комп'ютерної програми

```

# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
# Читання зображення як є (з альфа-каналом)
image = cv.imread("image.webp", -1)
# або
# image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
# Перевірка зчитування зображення, виведення повідомлення

```

у разі виникнення помилки та завершення роботи програми
if image is None:

```
    sys.exit("Could not read the image.")
```

Створення вікна для відображення

```
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
```

Відображення у вікні прочитаного зображення

```
cv.imshow("My image", image)
```

Організація затримки до натиснення довільної клавіші

```
cv.waitKey(0)
```

Знищення всіх створених вікон

```
cv.destroyAllWindows()
```

або знищення створеного вікна

```
# cv.destroyWindow("My image")
```

Для перевірки наведених вище кодів можна використати зображення, наведене на рис. 16. Необхідно розмістити це зображення (або довільне інше на свій розсуд) у папці, де буде міститися файл з кодом (рис. 17).



Рис. 16. Оригінальне зображення з ім'ям "image.webp"

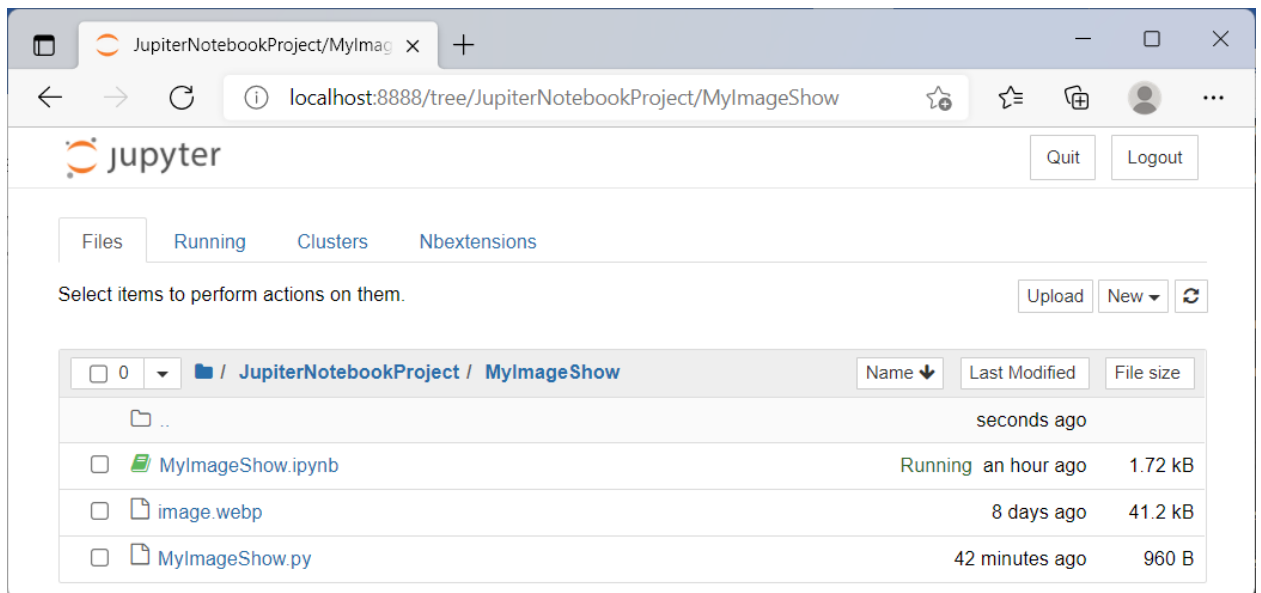


Рис. 17. Вміст папки проекту у Jupyter Notebook

Далі необхідно вставити відповідний код у Jupyter Notebook у новий Python 3 (ipykernel) файл та зберегти з ім'ям MyImageShow.ipynb (рис. 18) і натиснути кнопку «Run». Повинно з'явиться вікно із зображенням (рис. 19).

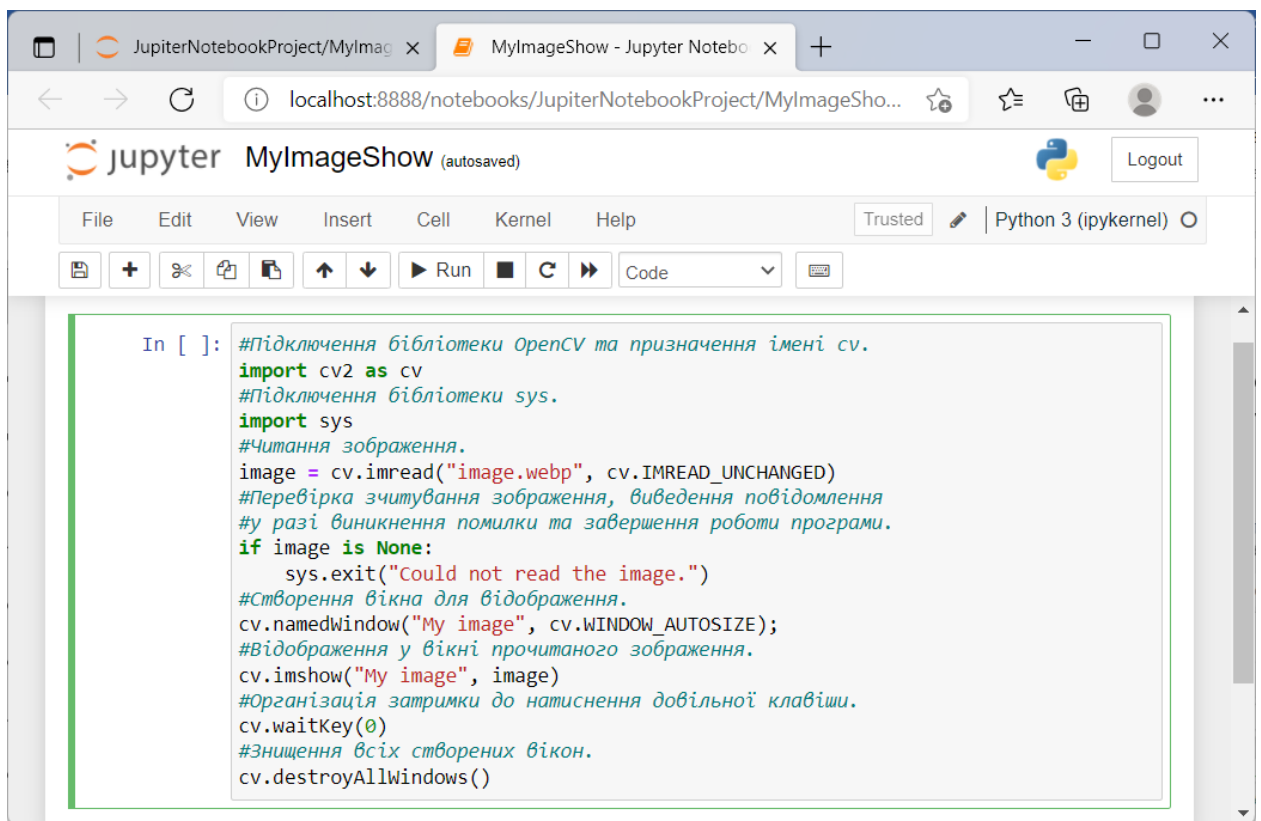


Рис. 18. Python 3 (ipykernel) файл MyImageShow.ipynb у Jupyter Notebook



Рис. 19. Вікно із зображенням з файлу з ім'ям "image.webp"

Також можна відповідний код розбити на блоки та вставити кожен блок у свою комірку у Jupyter Notebook у новому Python 3 (ipykernel) файлі, зберегти з ім'ям MyImageShow2.ipynb (рис. 20) і виконати його натискаючи кнопку «Run» на кожній комірці або командою Cell -> Run All. Повинно з'явиться вікно із зображенням (рис. 19).

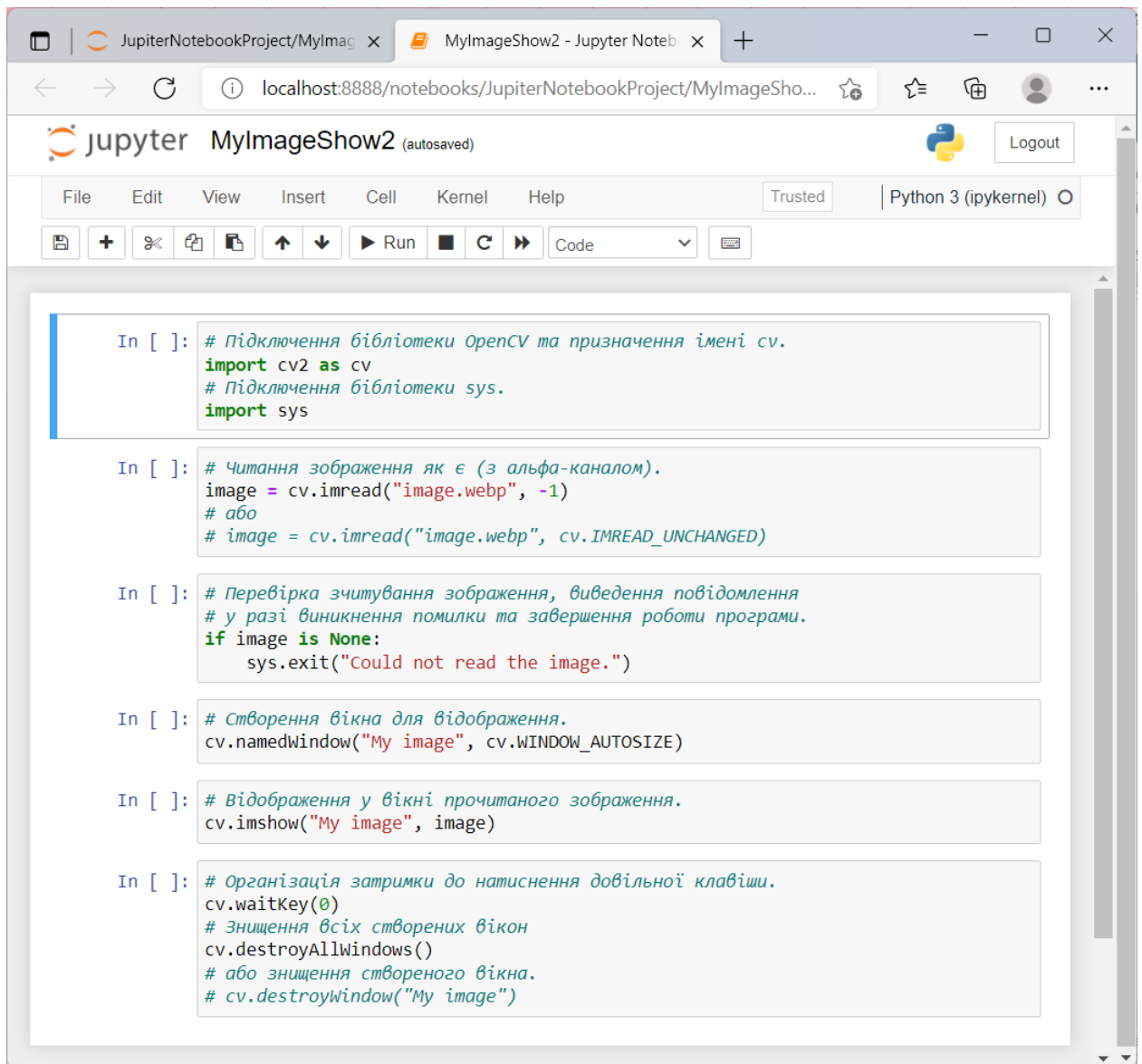


Рис. 20. Python 3 (ipykernel) файл MyImageShow2.ipynb у Jupyter Notebook

Також у Jupyter Notebook можна вставити відповідний код у новий Text File файл та зберегти його з ім'ям MyImageShow.py (рис. 21). Для виконання коду з файлу MyImageShow.py у Jupyter Notebook необхідно створити новий Python 3 (ipykernel) файл, вставити у нього команду (рис. 22):

```
%run ./MyImageShow.py
```

і натиснути кнопку «Run». Аналогічно до попереднього повинне з'явитися вікно із зображенням (рис. 19).

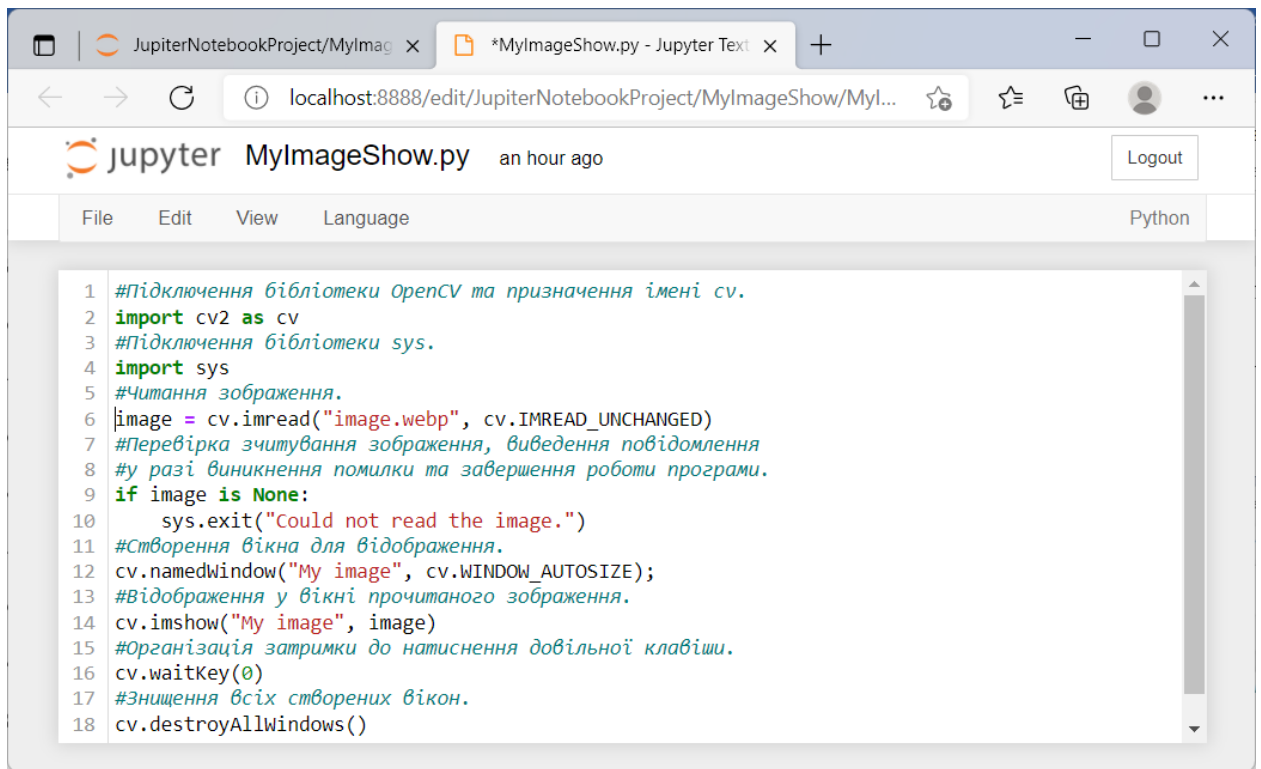


Рис. 21. Text File файл MyImageShow.py у Jupyter Notebook

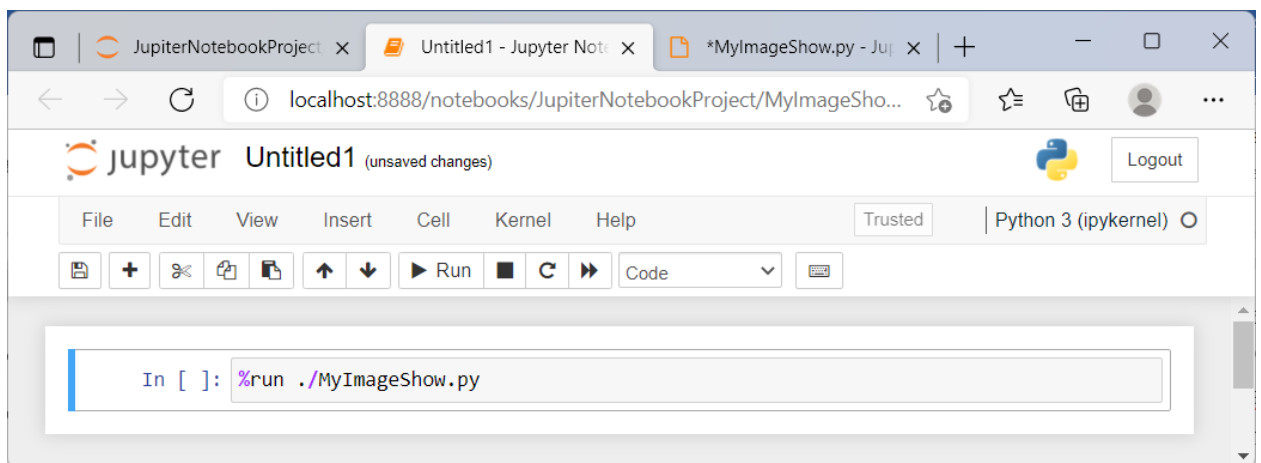


Рис. 22. Запуск на виконання файлу MyImageShow.py у Jupyter Notebook

Далі розглянемо приклад коду комп'ютерної програми (лістинг 3), що за допомогою функцій бібліотеки OpenCV читатиме з поточної папки деяке зображення як ϵ (з альфа-каналом) з файлу, що задається як параметр командного рядка, створюватиме вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Лістинг 3. Код комп'ютерної програми

```

# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
# Читання зображення як є (з альфа-каналом)
image = cv.imread(sys.argv[1], -1)
# або
# image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)
# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if image is None:
    sys.exit("Could not read the image.")
# Створення вікна для відображення
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
# Відображення у вікні прочитаного зображення
cv.imshow("My image", image)
# Організація затримки до натиснення довільної клавіші
cv.waitKey(0)
# Знищення всіх створених вікон
cv.destroyAllWindows()
# або знищення створеного вікна
# cv.destroyWindow("My image")

```

Якщо у Jupyter Notebook вставити відповідний код у новий Text File файл та зберегти його з ім'ям MyImageShow.py (рис. 23). Для виконання коду з файлу MyImageShow.py у Jupyter Notebook необхідно створити новий Python 3 (ipykernel) файл, вставити у нього команду (рис. 24):

```
%run ./MyImageShow.py image.webp
```

і натиснути кнопку «Run». Аналогічно до попереднього повинне з'явитися вікно із зображенням (рис. 19).

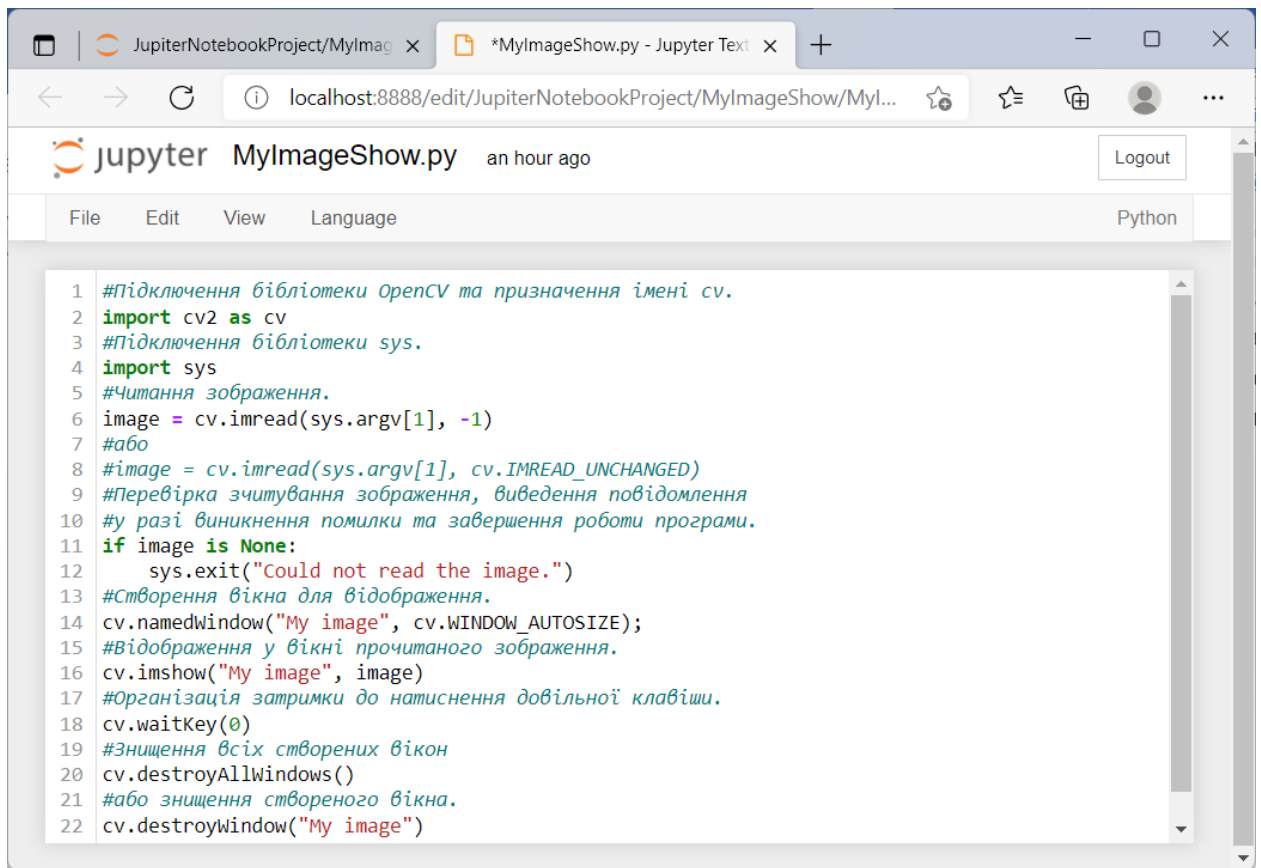


Рис. 23. Text File файл MyImageShow.py у Jupyter Notebook

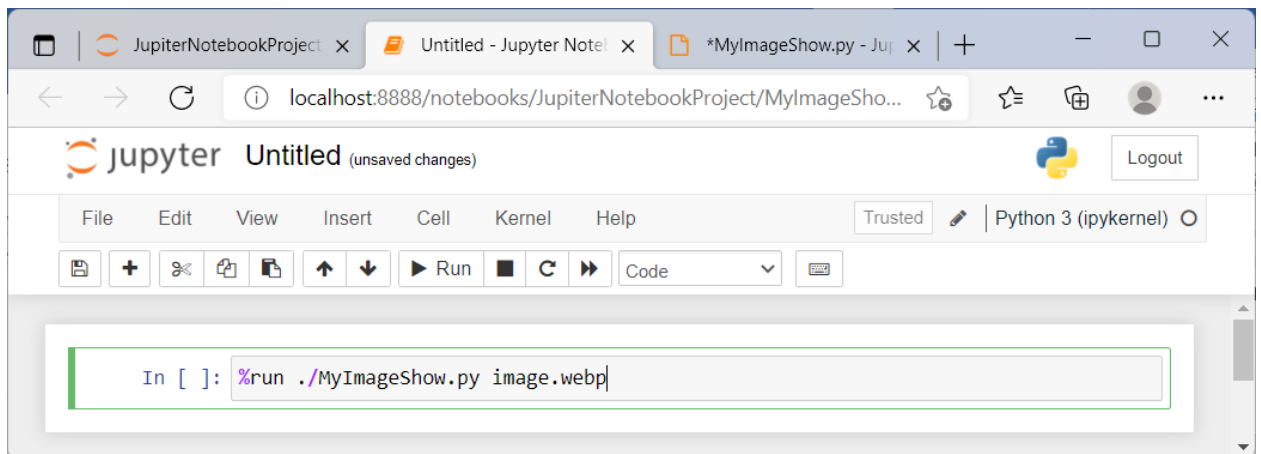


Рис. 24. Запуск на виконання файлу MyImageShow.py у Jupyter Notebook

Далі розглянемо приклад коду комп'ютерної програми (лістинг 4), що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як 3-канальне кольорове з файлу з ім'ям "image.webp" з поточної папки, створюватиме вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразить у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіши.

Лістинг 4. Код комп'ютерної програми

```
# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
# Читання зображення як 3-канальне кольорове
image_color = cv.imread("image.webp", 1)
# або
# image_color = cv.imread("image.webp", cv.IMREAD_COLOR)
# або
# image_color = cv.imread("image.webp")
# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if image_color is None:
    sys.exit("Could not read the image.")
# Створення вікна для відображення
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
# Відображення у вікні прочитаного зображення
cv.imshow("Color image", image_color)
# Організація затримки до натиснення довільної клавіші
cv.waitKey(0)
# Знищення всіх створених вікон
cv.destroyAllWindows()
# або знищення створеного вікна
# cv.destroyWindow("Color image")
```

Якщо у Jupyter Notebook вставити відповідний код у новий Text File файл та зберегти його з ім'ям MyImageShow.py, а потім у Jupyter Notebook створити новий Python 3 (ipykernel) файл з командою %run ./MyImageShow.py і натиснути кнопку «Run», то аналогічно до попереднього повинне з'явитися вікно з відповідним кольоровим зображенням без альфа-каналу (рис. 19).

Далі розглянемо приклад коду комп'ютерної програми (лістинг 5), що за допомогою функцій бібліотеки OpenCV читатиме деяке зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.webp" з поточної папки, зберігатиме його файлі з ім'ям "image_grayscale.webp" у поточній

папці, створюватиме вікно з ім'ям "Grayscale image", дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Лістинг 5. Код комп'ютерної програми

```
# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
# Читання зображення як одноканального у відтінках сірого
image_grayscale = cv.imread("image.webp", 0)
# або
# image_grayscale = cv.imread("image.webp", cv.IMREAD_GRAYSCALE)
# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if image_grayscale is None:
    sys.exit("Could not read the image.")
# Збереження зображення
cv.imwrite("image_grayscale.webp", image_grayscale)
# Створення вікна для відображення.
cv.namedWindow("Grayscale image", cv.WINDOW_NORMAL)
# Відображення у вікні прочитаного зображення
cv.imshow("Grayscale image", image_grayscale)
# Організація затримки до натиснення довільної клавіші
cv.waitKey(0)
# Знищення всіх створених вікон
cv.destroyAllWindows()
# або знищення створеного вікна
# cv.destroyWindow("Grayscale image")
```

Після виконання цього коду у відповідній папці поряд з оригінальним зображенням буде створено відповідне одноканальне зображення у відтінках сірого і на екран монітора буде виведено вікно з цим зображенням (рис. 25).

Розглянемо детальніше останній наведений код. На початку коду виконується підключення бібліотеки OpenCV та призначення імені cv і бібліотеки sys за допомогою директиви import (лістинг 6).



Рис. 25. Результуюче зображення з ім'ям "image_grayscale.webp"

Лістинг 6. Фрагмент коду комп'ютерної програми

```
# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
```

Далі для читання зображення використовується функція `imread()` бібліотеки OpenCV, яка має наступний синтаксис:

```
imread(filename, flags)
```

Функція `imread()` бібліотеки OpenCV має два аргументи:

1) перший аргумент – це ім'я файлу (специфікація) зображення, яке обов'язково має містити розширення (наприклад, `.webp`, `.png`, `.jpg` тощо), оскільки бібліотека OpenCV використовує його для визначення формату файлу;

2) другий аргумент – це необов'язковий прапорець, який дозволяє вказати, як має бути представлено зображення.

Бібліотека OpenCV пропонує різні варіанти значень цього прапорця, які наведені у табл. 1.

Табл. 1. Прапорці функції imread()

Значення		Призначення
IMREAD_UNCHANGED	-1	зображення зчитується як є (з альфа-каналом). Ігнорується орієнтація EXIF
IMREAD_GRAYSCALE	0	зображення конвертується в одноканальне зображення у відтінках сірого (внутрішнє перетворення кодеку)
IMREAD_COLOR	1	зображення конвертується в 3-канальне кольорове зображення BGR
IMREAD_ANYDEPTH	2	зображення конвертується у 16-розрядне/32-розрядне, коли вхідне має відповідну глибину, інакше конвертується у 8-бітове
IMREAD_ANYCOLOR	4	зображення зчитується в будь-якому кольоровому форматі
IMREAD_LOAD_GDAL	8	використовується драйвер gdal для завантаження образу
IMREAD_REDUCED_GRAYSCALE_2	16	зображення конвертується в одноканальне зображення у відтінках сірого, а розмір зображення зменшено на 1/2
IMREAD_REDUCED_COLOR_2	17	зображення конвертується в 3-канальне кольорове зображення BGR, а розмір зображення зменшено на 1/2

Значення		Призначення
IMREAD_REDUCED_GRAYSCALE_4	32	зображення конвертується в одноканальне зображення у відтінках сірого, а розмір зображення зменшено на 1/4
IMREAD_REDUCED_COLOR_4	33	зображення конвертується в 3-канальне кольорове зображення BGR, а розмір зображення зменшено на 1/4
IMREAD_REDUCED_GRAYSCALE_8	64	зображення конвертується в одноканальне зображення у відтінках сірого, а розмір зображення зменшено на 1/8
IMREAD_REDUCED_COLOR_8	65	зображення конвертується в 3-канальне кольорове зображення BGR, а розмір зображення зменшено на 1/8
IMREAD_IGNORE_ORIENTATION	128	ігнорується орієнтація EXIF

Найпоширенішими з них є:

- 1) IMREAD_UNCHANGED або -1;
- 2) IMREAD_GRAYSCALE або 0;
- 3) IMREAD_COLOR або 1.

Значенням за замовчуванням для прапорця є IMREAD_COLOR або 1, яке означає, що зображення буде прочитане як 3-канальне кольорове зображення BGR. Якщо зображення необхідно прочитати у певному форматі, то необхідно вказати відповідний прапорець.

Важливо також зазначити, що бібліотека OpenCV читає кольорові зображення у форматі BGR, тоді як більшість інших бібліотек цифрової обробки зображень та комп'ютерного зору використовують порядок форматів

каналів RGB. Тому, використовуючи бібліотеку OpenCV з іншими наборами інструментів, потрібно не забувати міняти місцями канали синього та червоного кольорів під час переходу з однієї бібліотеки на іншу.

Нижче наведено фрагмент коду для читання зображення як одноканального у відтінках сірого (лістинг 7).

Лістинг 7. Фрагмент коду комп'ютерної програми

```
# Читання зображення як одноканального у відтінках сірого
image_grayscale = cv.imread("image.webp", 0)
# або
# image_grayscale = cv.imread("image.webp", cv.IMREAD_GRAYSCALE)
```

Нижче наведено фрагмент коду для читання деякого зображення у найуживаніших варіантах.

Лістинг 8. Фрагменти коду

```
# Читання зображення з файлу "image.png" як є (з альфа-каналом)
image_unchanged = cv.imread("image.webp", -1)
# або
# image_unchanged = cv.imread("image.webp",
#                               cv.IMREAD_UNCHANGED)
# Читання зображення з файлу "image.jpg"
# як одноканального у відтінках сірого
image_grayscale = cv.imread("image.jpg", 0)
# або
# image_grayscale = cv.imread("image.jpg", cv.IMREAD_GRAYSCALE)
# Читання зображення з файлу "image.webp"
# як 3-канального кольорового зображення BGR
image_color = cv.imread("image.webp", 1)
# або
# image_color = cv.imread("image.webp", cv.IMREAD_COLOR)
# або
# image_color = cv.imread("image.webp")
```

Далі здійснюється перевірка на коректність зчитування зображення. У разі, коли змінна `image_grayscale` не отримує значення, виводиться

повідомлення про помилку читання зображення. Нижче наведено відповідний фрагмент коду (лістинг 9).

```
Лістинг 9. Фрагмент коду комп'ютерної програми
# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if image_grayscale is None:
    sys.exit("Could not read the image.")
```

Далі для збереження зображення використовується функція `imwrite()` бібліотеки `OpenCV`, яка має наступний синтаксис:

```
imwrite(filename, image, flags)
```

Функція `imwrite()` має три аргументи:

1) перший аргумент – це ім'я (специфікація) файлу, яке обов'язково (як і у випадку функції `imread()`) має містити розширення імені файлу (наприклад, `.webp`, `.png`, `.jpg` тощо), оскільки бібліотека `OpenCV` використовує його для визначення формату файлу;

2) другий аргумент – зображення, яке потрібно зберегти;

3) третій аргумент – задає необов'язкові додаткові параметри для вихідного зображення за допомогою одного або кількох аргументів пари “прапорць–значення” (див. табл. 2).

Функція повертає значення `True`, якщо зображення успішно збережено.

Табл. 2. Прапорці функції `cv.imwrite()`

Значення		Призначення
<code>IMWRITE_JPEG_QUALITY</code>	1	Для JPEG задає якість зображення від 0 до 100 (чим вище, тим краще). Значення за замовчуванням – 95.
<code>IMWRITE_JPEG_PROGRESSIVE</code>	2	Для JPEG дозволяє включити режим прогресивного стиснення, 0 або 1, за замовчуванням встановлено значення <code>False</code> .

Значення		Призначення
IMWRITE_JPEG_OPTIMIZE	3	Для JPEG дозволяє включити режим оптимізації, 0 або 1, за замовчуванням встановлено значення False.
IMWRITE_JPEG_LUMA_QUALITY	5	Для JPEG дозволяє включити окремий рівень якості яскравості, 0 – 100, за замовчуванням 0 – не використовувати.
IMWRITE_JPEG_CHROMA_QUALITY	6	Для JPEG дозволяє включити окремий рівень якості кольорів, 0 – 100, за замовчуванням 0 – не використовувати.
IMWRITE_PNG_COMPRESSION	16	Для PNG задає рівень стиснення від 0 до 9. Більше значення означає менший розмір і довший час стиснення.
IMWRITE_WEBP_QUALITY	64	Для WEBP задає якість зображення від 1 до 100 (чим вище, тим краще). За замовчуванням (без будь-якого параметра) і для якості 100 використовується стиснення без втрат.

Нижче наведено фрагмент коду (лістинг 10), що демонструє збереження зображення. Для цього необхідно вказати назву файлу з потрібним розширенням (з-за необхідності з будь-яким бажаним шляхом до нього) та ім'я змінної, яка містить дане зображення.

```
Лістинг 10. Фрагмент коду комп'ютерної програми
# Збереження зображення
cv.imwrite("image_grayscale.webp", image_grayscale)
```

Нижче наведено фрагмент коду (лістинг 11), що демонструє можливість зміни якості збереження результуючого зображення шляхом зміни третього параметра функції `imwrite()`.

Лістинг 11. Фрагмент коду

Збереження зображення

```
cv.imwrite("image_grayscale.webp", image_grayscale,  
          [cv.IMWRITE_WEBP_QUALITY, 100])
```

Потім для створення вікна комп'ютерної програми з заданими назвою та розміром, призначеного для відображення зчитаного зображення, використовується функція `namedWindow()` бібліотеки `OpenCV`, яка має наступний синтаксис:

```
namedWindow(window_name, flag)
```

Ця функція має два аргументи:

1) перший аргумент `window_name` – це назва вікна, у якому буде відображатися зображення;

2) другий аргумент `flag` – прапорець – показує, чи розмір вікна автоматично встановлюється або регулюється.

Деякі із можливих значень аргумента `flag`:

1) `WINDOW_NORMAL` – дозволяє користувачеві вручну змінювати розмір вікна (без обмежень) та режим вікна: звичайний або повноекранний;

2) `WINDOW_AUTOSIZE` – автоматично встановлює розмір вікна у відповідності до розміру зображення, користувачеві не можна змінити вручну розмір вікна (за замовчуванням);

3) `WINDOW_FULLSCREEN` – дозволяє користувачеві змінювати режим вікна: звичайний або повноекранний, але розмір вікна змінити не можна.

Нижче наведено фрагмент коду (лістинг 12), що демонструє можливість створення вікна для відображення, в якому користувачу дозволено змінити розмір вікна та режим вікна зі звичайного на повноекранний і навпаки.

Лістинг 12. Фрагмент коду комп'ютерної програми

Створення вікна для відображення.

```
cv.namedWindow("Grayscale image", cv.WINDOW_NORMAL)
```

Далі для відображення зображення у вікні комп'ютерної програми використовується функція `imshow()` бібліотеки `OpenCV`, яка має наступний синтаксис:

```
imshow(window_name, image)
```

Ця функція також має два аргументи:

1) перший аргумент `window_name` – це назва вікна, у якому буде відображатися зображення;

2) другий аргумент `image` – це зображення, яке потрібно відобразити.

Нижче наведено приклад фрагменту коду (лістинг 13), що демонструє відображення прочитаних зображень.

Лістинг 13. Фрагмент коду комп'ютерної програми

```
# Відображення у вікні прочитаного зображення.  
cv.imshow("Grayscale image", image_grayscale)
```

Щоб одночасно відобразити кілька зображень, необхідно за допомогою функції `namedWindow()` створити кілька вікон з різними назвами та за допомогою функції `imshow()` їх відобразити, вказавши назви вікон для кожного зображення.

Функція `imshow()` призначена для використання разом з функцією `waitKey()` і функцією `destroyWindow()` або `destroyAllWindows()`.

Функція `waitKey()` є функцією для організації затримки для показу вікна, призначеного для відображення зображення. Має єдиний аргумент – час (у мілісекундах), протягом якого буде відображатися вікно. Якщо протягом цього часу користувач натисне будь-яку клавішу, програма продовжиться. Якщо значенням аргументу задано 0, програма чекає натискання будь-якої клавіші.

Також можна реалізувати функцію для виявлення конкретних натискань клавіш на клавіатурі, таких як `Q` або `ESC`, тим самим чіткіше повідомляючи, яка клавіша повинна викликати таку поведінку.

Функція `destroyWindow()` служить для знищення вікна, створеного у ході виконання коду і призначеного для відображення зображення. Має єдиний аргумент – назву вікна, що має бути знищене. Функція `destroyAllWindows()` призначена для знищення всіх вікон, створених у ході виконання коду і призначених для відображення зображень. Якщо потрібно знищити певне вікно, необхідно вказати відповідне ім'я вікна як аргумент. Використання

функцій `destroyWindow()` та `destroyAllWindows()` також дозволяє очистити основну пам'ять системи від наявних у ній зображень.

Нижче наведено фрагмент коду (лістинг 14), що демонструє можливість організації затримки до натиснення довільної клавіші та знищення створених вікон.

Лістинг 14. Фрагмент коду комп'ютерної програми

```
# Організація затримки до натиснення довільної клавіші
cv.waitKey(0)
# Знищення всіх створених вікон
cv.destroyAllWindows()
# або знищення створеного вікна
# cv.destroyWindow("Grayscale image")
```

Далі розглянемо приклад коду (лістинг 15), реалізований з використанням об'єктно-орієнтованого підходу, комп'ютерної програми, що за допомогою функцій бібліотеки `OpenCV` читатиме деяке зображення як одноканальне у відтінках сірого з файлу з ім'ям `"image.webp"` з поточної папки, зберігатиме його файлі з ім'ям `"image_grayscale.webp"` у поточній папці, створюватиме вікно з ім'ям `"Grayscale image"`, дозволивши вручну змінювати розміри вікна, відображатиме у цьому вікні зчитане зображення та знищуватиме створене вікно після натиснення будь-якої клавіші.

Лістинг 15. Код комп'ютерної програми

```
# Підключення бібліотеки OpenCV та призначення імені cv
import cv2 as cv
# Підключення бібліотеки sys
import sys
# Опис класу Image
class Image:
    # Конструктор для зчитування зображення
    # як одноканального у відтінках сірого із файлу
    def __init__(self, path):
        self.image = cv.imread(path, 0)
        # або
        # self.image = cv.imread(path, cv.IMREAD_GRAYSCALE)
```

```

# Перевірка зчитування зображення, виведення повідомлення
# у разі виникнення помилки та завершення роботи програми
if self.image is None:
    sys.exit("Could not read the image.")
# Метод для збереження зображення у файл
def write(self, path):
    cv.imwrite(path, self.image)
# Метод для створення вікна і відображення у ньому зображення
def show(self, caption):
    cv.namedWindow(caption, cv.WINDOW_NORMAL)
    cv.imshow(caption, self.image)
# Деструктор для організації затримки до натиснення довільної
# клавіши і знищення всіх створених вікон
def __del__(self):
    cv.waitKey(0)
    cv.destroyAllWindows()
# Код, який розміщений після рядка if __name__ == "__main__":,
# запускається лише, якщо цей модуль виконується як "основний" файл
# (не запускається при імпорті цього модуля)
if __name__ == "__main__":
    # Створення об'єкта image_grayscale та зчитування у поле image
    # зображення з файлу з іменем "image.webp" з поточної папки
    image_grayscale = Image("image.webp")
    # Збереження зображення, що міститься у полі image
    # об'єкта image_grayscale у файлі з іменем "image_grayscale.webp"
    # в поточній папці
    image_grayscale.write("image_grayscale.webp")
    # Створення вікна з ім'ям "Grayscale image" і відображення у ньому
    # зображення, що міститься у полі image об'єкта image_grayscale
    image_grayscale.show("Grayscale image")
    # Знищення об'єкта image_grayscale та очищення пам'яті
    del image_grayscale

```

II. Опрацюйте наступні тестові завдання для перевірки засвоєння матеріалу (підготуйтеся до тестування):

Питання першого рівня складності (необхідно вибрати одну відповідь із запропонованого списку):

1. У комп'ютерних програмах мовою програмування Python для імпорту бібліотеки OpenCV використовується директива:

- `import cv2`
- `import cv`
- `import opencv`
- `import opencv2`
- `import opencv as cv`
- `import opencv2 as cv2`

2. У комп'ютерних програмах мовою програмування Python для зчитування зображення використовується функція бібліотеки OpenCV:

- `imwrite()`
- `imread()`
- `imload()`
- `imWrite()`
- `imRead()`
- `imLoad()`

3. У комп'ютерних програмах мовою програмування Python для виконання збереження зображення використовується функція бібліотеки OpenCV:

- `imwrite()`
- `imread()`
- `imsave()`
- `imWrite()`
- `imRead()`
- `imSave()`

4. У комп'ютерній програмі мовою програмування Python для створення вікна з відповідною назвою та розміром, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

- `newWindow()`
- `newwindow()`
- `createWindow()`

- `createwindow()`
- `namedWindow()`
- `namedwindow()`

5. У комп'ютерних програмах мовою програмування Python для відображення зображення у створеному вікні використовується функція бібліотеки OpenCV:

- `imwrite()`
- `imprint()`
- `imshow()`
- `imWrite()`
- `imPrint()`
- `imShow()`

6. У комп'ютерній програмі мовою програмування Python для організації затримки до натиснення будь-якої клавіші для показу вікна, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

- `waitKey()`
- `waitkey()`
- `wait()`
- `pauseKey()`
- `pausekey()`
- `pause()`

7. У комп'ютерній програмі мовою програмування Python для організації затримки до натиснення будь-якої клавіші для показу вікна, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

- `waitKey()`
- `waitChar()`
- `wait()`
- `pauseKey()`
- `pauseChar()`
- `pause()`

8. У комп'ютерній програмі мовою програмування Python для знищення вікна, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

- `destroyAllWindows()`
- `destroyWindows()`
- `destroyWindow()`
- `deleteAllWindows()`
- `deleteWindows()`
- `deleteWindow()`

9. У комп'ютерній програмі мовою програмування Python для знищення всіх вікон, призначених для відображення зображень, використовується функція бібліотеки OpenCV:

- `destroyAllWindows()`
- `destroyWindows()`
- `destroyWindow()`
- `deleteAllWindows()`
- `deleteWindows()`
- `deleteWindow()`

10. ...

```
image = cv.imread("image.webp", -1)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `import cv2 as cv`
- `import cv`
- `import opencv`

- import opencv2
- import opencv as cv
- import opencv2 as cv

```
11. import cv2 as cv
```

```
...
```

```
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
```

```
cv.imshow("My image", image)
```

```
cv.waitKey(0)
```

```
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- image = cv.imread("image.webp", -1)
- image = cv.imread("image.webp", -1)
- image = cv.imread("image.webp", 0)
- image = cv.imread("image.webp", 0)
- image = cv.imread("image.webp", 1)
- image = cv.imread("image.webp", 1)

```
12. import cv2 as cv
```

```
...
```

```
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
```

```
cv.imshow("My image", image)
```

```
cv.waitKey(0)
```

```
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image",

автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.webp", cv.IMAGE_UNCHANGED)`
- `image = cv.imread("image.webp", cv.READ_UNCHANGED)`
- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.webp", cv.READ_UNCHANGED)`

```
13. import cv2 as cv
```

```
...
```

```
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)  
cv.imshow("My image", image)  
cv.waitKey(0)  
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.webp", cv.IMREAD_GRAYSCALE)`
- `image = cv.imread("image.webp", cv.IMREAD_COLOR)`
- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.webp", cv.IMREAD_GRAYSCALE)`
- `image = cv.imread("image.webp", cv.IMREAD_COLOR)`

```
14. import cv2 as cv
```

```
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
```

```
...
```

```
cv.imshow("My image", image)  
cv.waitKey(0)
```

```
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
- cv.createWindow("My image", cv.WINDOW_AUTOSIZE)
- cv.namedWindow("My image", cv.WINDOW_NORMAL)
- cv.createWindow("My image", cv.WINDOW_NORMAL)
- cv.namedWindow("My image", cv.WINDOW_AUTO)
- cv.createWindow("My image", cv.WINDOW_AUTO)

```
15. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
...
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.imshow("My image", image)
- cv.imwrite("My image", image)
- cv.imprint("My image", image)
- cv.imshow(image, "My image")
- cv.imwrite(image, "My image")
- cv.imprint(image, "My image")

```
16. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
...
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.waitKey(0)
- cv.waitKey()
- cv.wait(0)
- cv.wait()
- cv.waitKey(0)
- cv.waitKey()

```
17. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
...
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.waitKey(0)
- cv.wait(0)
- cv.waitChar(0)
- cv.pauseKey(0)
- cv.pause(0)
- cv.pauseChar(0)

```
18. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "webp.jpg" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.destroyWindow("My image")
- cv.destroyWindow()
- cv.destroyAllWindows()
- cv.deleteWindow("My image")
- cv.deleteWindow()
- cv.deleteAllWindows()

```
19. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[0], -1)`
- `image = cv.imread(sys.argv[1], 0)`
- `image = cv.imread(sys.argv[0], 0)`
- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[0], 1)`

```
20. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_GRAYSCALE)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_GRAYSCALE)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_COLOR)`


```

21. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.READ_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMAGE_UNCHANGED)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[0], cv.READ_UNCHANGED)`
- `image = cv.imread(sys.argv[0], cv.IMAGE_UNCHANGED)`

```

22. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.READ_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMAGE_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.READ_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMAGE_UNCHANGED)`

23. ...

```
image = cv2.imread("image.webp", -1)
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `import cv2`
- `import cv`
- `import opencv`
- `import opencv2`
- `import opencv as cv2`
- `import opencv2 as cv2`

24. `import cv2`

...

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки

OpenCV2 зчитувала зображення як ϵ (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread("image.webp", -1)`
- `image = cv2.imread("image.webp", 0)`
- `image = cv2.imread("image.webp", 1)`
- `image = cv2.imread("image.webp", -1)`
- `image = cv2.imread("image.webp", 0)`
- `image = cv2.imread("image.webp", 1)`

25. `import cv2`

...

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як ϵ (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.READ_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.IMAGE_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.IMAGE_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.READ_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)`

26. `import cv2`

...

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
```

```
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyAllWindows("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread("image.webp", cv2.IMREAD_GRAYSCALE)`
- `image = cv2.imread("image.webp", cv2.IMREAD_COLOR)`
- `image = cv2.imread("image.webp", cv2.IMREAD_COLOR)`
- `image = cv2.imread("image.webp", cv2.IMREAD_GRAYSCALE)`
- `image = cv2.imread("image.webp", cv2.IMREAD_COLOR)`

```
27. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
...
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyAllWindows("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)`
- `cv2.createWindow("My image", cv2.WINDOW_AUTOSIZE)`
- `cv2.namedWindow("My image", cv2.WINDOW_NORMAL)`
- `cv2.createWindow("My image", cv2.WINDOW_NORMAL)`

- `cv2.namedWindow("My image", cv2.WINDOW_AUTO)`
- `cv2.createWindow("My image", cv2.WINDOW_AUTO)`

```
28. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
...
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv2.imshow("My image", image)`
- `cv2.imwrite("My image", image)`
- `cv2.imprint("My image", image)`
- `cv2.imshow(image, "My image")`
- `cv2.imwrite(image, "My image")`
- `cv2.imprint(image, "My image")`

```
29. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
...
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення,

відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv2.waitKey(0)
- cv2.waitKey()
- cv2.wait(0)
- cv2.wait()
- cv2.waitKeyChar(0)
- cv2.waitKeyChar()

```
30. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
...
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv2.waitKey(0)
- cv2.wait(0)
- cv2.waitKeyChar(0)
- cv2.pauseKey(0)
- cv2.pause(0)
- cv2.pauseChar(0)

```
31. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "webp.jpg" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv2.destroyWindow("My image")
- cv2.destroyWindow()
- cv2.destroyAllWindows()
- cv2.deleteWindow("My image")
- cv2.deleteWindow()
- cv2.deleteAllWindows()

```
32. import cv2
import sys
...
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- image = cv2.imread(sys.argv[1], -1)
- image = cv2.imread(sys.argv[0], -1)
- image = cv2.imread(sys.argv[1], 0)
- image = cv2.imread(sys.argv[0], 0)
- image = cv2.imread(sys.argv[1], 1)
- image = cv2.imread(sys.argv[0], 1)

```

33. import cv2
import sys
...
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)
- image = cv2.imread(sys.argv[1], cv2.READ_UNCHANGED)
- image = cv2.imread(sys.argv[1], cv2.IMAGE_UNCHANGED)
- image = cv2.imread(sys.argv[0], cv2.IMREAD_UNCHANGED)
- image = cv2.imread(sys.argv[0], cv2.READ_UNCHANGED)
- image = cv2.imread(sys.argv[0], cv2.IMAGE_UNCHANGED)

```

34. import cv2
import sys
...
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру

зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.READ_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMAGE_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.READ_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMAGE_UNCHANGED)`

```
35. import cv2
```

```
import sys
```

```
...
```

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
```

```
cv2.imshow("My image", image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV2 зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[0], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_GRAYSCALE)`
- `image = cv2.imread(sys.argv[0], cv2.IMREAD_GRAYSCALE)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_COLOR)`
- `image = cv2.imread(sys.argv[0], cv2.IMREAD_COLOR)`

```
36. import cv2 as cv
```

```
...
```

```
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
```

```
cv.imshow("Color image", image)
```

```
cv.waitKey(0)
```

```
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.png", -1)`
- `image = cv.imread("image.png", -1)`
- `image = cv.imread("image.png", 0)`
- `image = cv.imread("image.png", 0)`
- `image = cv.imread("image.png", 1)`
- `image = cv.imread("image.png", 1)`

```
37. import cv2 as cv
```

```
...
```

```
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
```

```
cv.imshow("Color image", image)
```

```
cv.waitKey(0)
```

```
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.png", cv.IMCOLOR)`
- `image = cv.imread("image.png", cv.IMCOLOR)`
- `image = cv.imread("image.png", cv.IMAGE_COLOR)`
- `image = cv.imread("image.png", cv.IMAGE_COLOR)`
- `image = cv.imread("image.png", cv.IMREAD_COLOR)`
- `image = cv.imread("image.png", cv.IMREAD_COLOR)`

```

38. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
...
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyAllWindows("Color image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
- cv.createWindow("Color image", cv.WINDOW_AUTOSIZE)
- cv.namedWindow("Color image", cv.WINDOW_NORMAL)
- cv.createWindow("Color image", cv.WINDOW_NORMAL)
- cv.namedWindow("Color image", cv.WINDOW_AUTO)
- cv.createWindow("Color image", cv.WINDOW_AUTO)

```

39. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
...
cv.waitKey(0)
cv.destroyAllWindows("Color image")

```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.imshow("Color image", image)

- cv.imshow("My image", image)
- cv.imwrite("Color image", image)
- cv.imwrite("My image", image)
- cv.imread("Color image", image)
- cv.imread("My image", image)

```
40. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
...
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.waitKey(0)
- cv.waitKey()
- cv.wait(0)
- cv.wait()
- cv.waitKey(0)
- cv.waitKey()

```
41. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
...
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки

OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.waitKey(0)
- cv.wait(0)
- cv.waitChar(0)
- cv.pauseKey(0)
- cv.pause(0)
- cv.pauseChar(0)

```
42. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- cv.destroyWindow("Color image")
- cv.destroyWindow("My image")
- cv.destroyWindow()
- cv.deleteWindow("Color image")
- cv.deleteWindow("My image")
- cv.deleteWindow()

```
43. import cv2 as cv
...
cv.imwrite("image_grayscale.jpg", image_grayscale)
```

```
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- image_grayscale = cv.imread("image.jpg", -1)
- image_grayscale = cv.imread("image.jpg", 0)
- image_grayscale = cv.imread("image.jpg", 1)
- image_grayscale = cv.imread("image.jpg", 0)
- image_grayscale = cv.imread("image.jpg", 1)
- image_grayscale = cv.imread("image.jpg", 1)

```
44. import cv2 as cv
...
cv.imwrite("image_grayscale.jpg", image_grayscale)
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image_grayscale = cv.imread("image.jpg", cv.IMAGE_GRAYSCALE)`
- `image_grayscale = cv.imread("image.jpg", cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread("image.jpg", cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread("image.jpg", cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread("image.jpg", cv.READ_GRAYSCALE)`
- `image_grayscale = cv.imread("image.jpg", cv.READ_GRAYSCALE)`

```
45. import cv2 as cv
image_grayscale = cv.imread("image.jpg", 0)
...
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv.imwrite("image_grayscale.jpg", image_grayscale)`
- `cv.imwrite("image_grayscale.webp", image_grayscale)`
- `cv.imwrite("image_grayscale.png", image_grayscale)`
- `cv.imprint("image_grayscale.jpg", image_grayscale)`
- `cv.imprint("image_grayscale.webp", image_grayscale)`
- `cv.imprint("image_grayscale.png", image_grayscale)`

```
46. import cv2 as cv
image_grayscale = cv.imread("image.jpg", 0)
...
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
```

```
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv.imwrite("image_grayscale.jpg", image_grayscale);`
- `cv.imwrite("image_grayscale.jpg", image_grayscale);`
- `cv.imwrite("image_grayscale.jpg", image_grayscale);`
- `cv.imwrite(image_grayscale, "image_grayscale.jpg");`
- `cv.imwrite(image_grayscale, "image_grayscale.jpg");`
- `cv.imwrite(image_grayscale, "image_grayscale.jpg");`

Питання другого рівня складності (необхідно вибрати декілька відповідей із запропонованого списку):

```
1. import cv2 as cv
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.webp", -1)`
- `image = cv.imread("image.webp", 0)`
- `image = cv.imread("image.webp", 1)`
- `image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)`

- `image = cv.imread("image.webp", cv.IMREAD_COLOR)`
- `image = cv.imread("image.webp")`

```
2. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[1], 0)`
- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[1])`

```
3. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My

image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[0], -1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1])`
- `image = cv.imread(sys.argv[0])`

```
4. import cv2 as cv
import sys
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп’ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім’ям “My image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1])`
- `image = cv.imread(sys.argv[1])`

```
5. import cv2 as cv
import sys
image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
```

```
cv.imshow("My image", image)
cv.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv.destroyAllWindows()`
- `cv.destroyWindow("My image")`
- `cv.destroyWindow()`
- `cv.deleteWindow("My image")`
- `cv.deleteWindow()`
- `cv.deleteAllWindows()`

```
6. import cv2
```

```
...
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread("image.webp", -1)`
- `image = cv2.imread("image.webp", 0)`
- `image = cv2.imread("image.webp", 1)`
- `image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)`

- `image = cv2.imread("image.webp", cv2.IMREAD_COLOR)`
- `image = cv2.imread("image.webp")`

7. `import cv2`

`import sys`

...

`cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)`

`cv2.imshow("My image", image)`

`cv2.waitKey(0)`

`cv2.destroyWindow("My image")`

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv2.imread(sys.argv[1], -1)`
- `image = cv2.imread(sys.argv[1], 0)`
- `image = cv2.imread(sys.argv[1], 1)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_COLOR)`
- `image = cv2.imread(sys.argv[1])`

8. `import cv2`

`import sys`

...

`cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)`

`cv2.imshow("My image", image)`

`cv2.waitKey(0)`

`cv2.destroyWindow("My image")`

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається

як параметр командного рядка, з поточної папки, створювала вікно з ім'ям “My image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread(sys.argv[1], -1)`
- `image = cv2.imread(sys.argv[0], -1)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[0], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1])`
- `image = cv2.imread(sys.argv[0])`

```
9. import cv2
```

```
import sys
```

```
...
```

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
```

```
cv2.imshow("My image", image)
```

```
cv2.waitKey(0)
```

```
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям “My image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv2.imread(sys.argv[1], -1)`
- `image = cv2.imread(sys.argv[1], -1)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)`
- `image = cv2.imread(sys.argv[1])`
- `image = cv2.imread(sys.argv[1])`

```
10. import cv2
```

```
import sys
```

```
image = cv2.imread(sys.argv[1], cv2.IMREAD_UNCHANGED)
```

```
cv2.namedWindow("My image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `cv2.destroyAllWindows()`
- `cv2.destroyWindow("My image")`
- `cv2.destroyWindow()`
- `cv2.deleteWindow("My image")`
- `cv2.deleteWindow()`
- `cv2.deleteAllWindows()`

```
11. import cv2 as cv
...
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.jpg" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image = cv.imread("image.jpg", -1)`
- `image = cv.imread("image.jpg", 0)`
- `image = cv.imread("image.jpg", 1)`

- `image = cv.imread("image.jpg", cv.IMREAD_UNCHANGED)`
- `image = cv.imread("image.jpg", cv.IMREAD_COLOR)`
- `image = cv.imread("image.jpg")`

```
12. import cv2 as cv
import sys
...
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], -1)`
- `image = cv.imread(sys.argv[1], 0)`
- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_UNCHANGED)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[1])`

```
13. import cv2 as cv
import sys
...
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу, що задається

як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[0], 1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[0], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[1])`
- `image = cv.imread(sys.argv[0])`

```
14. import cv2 as cv
import sys
...
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядки, які можна вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[1], 1)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[1], cv.IMREAD_COLOR)`
- `image = cv.imread(sys.argv[1])`
- `image = cv.imread(sys.argv[1])`

```
15. import cv2 as cv
...
cv.imwrite("image_grayscale.png", image_grayscale)
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
```



```
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.png" з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

- `image_grayscale = cv.imread("image.png", -1)`
- `image_grayscale = cv.imread("image.png", 0)`
- `image_grayscale = cv.imread("image.png", 1)`
- `image_grayscale = cv.imread("image.png", cv.IMAGE_GRAYSCALE)`
- `image_grayscale = cv.imread("image.png", cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread("image.png", cv.GRAYSCALE)`

```
16. import cv2 as cv
import sys
...
cv.imwrite("image_grayscale.png", image_grayscale)
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні

зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image_grayscale = cv.imread(sys.argv[1], 0)`
- `image_grayscale = cv.imread(sys.argv[0], 0)`
- `image_grayscale = cv.imread(sys.argv[0], cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread(sys.argv[1], cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread(sys.argv[1])`
- `image_grayscale = cv.imread(sys.argv[0])`

```
17. import cv2 as cv
import sys
...
cv.imwrite("image_grayscale.png", image_grayscale)
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувала збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

- `image_grayscale = cv.imread(sys.argv[1], 0)`
- `image_grayscale = cv.imread(sys.argv[1], 0)`
- `image_grayscale = cv.imread(sys.argv[1], cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread(sys.argv[1], cv.IMREAD_GRAYSCALE)`
- `image_grayscale = cv.imread(sys.argv[1])`
- `image_grayscale = cv.imread(sys.argv[1])`

18.

```
1 import cv2 as cv
2 image_grayscale = cv.imread("image.png")
3 cv.imwrite("image.png", image_grayscale)
4 cv.namedWindow("Grayscale image", cv.window_autosize)
5 cv.imshow("Grayscale image", image_grayscale)
6 cv.waitKey(0)
7 cv.deleteWindow("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище кодї комп'ютерної програма мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.png" з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7

19.

```
1 import cv2 as cv
2 image_grayscale = cv.imread("image.png", IMREAD_GRAYSCALE)
3 cv.imwrite("image_grayscale.png", image_grayscale)
4 cv.namedWindow("Grayscale image", cv.WINDOW_NORMAL)
5 cv.imshow("Grayscale image", image_grayscale)
6 cv.waitKey()
7 cv.destroyAllWindows("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище кодї комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.png" з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7

20.

```
1 import cv2 as cv
2 import sys
3 image_grayscale = cv.imread(argv[1], cv.IMREAD_GRAYSCALE)
4 cv.imwrite("image_grayscale.png", image_grayscale)
5 cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
6 cv.imshow("Grayscale image", image_grayscale)
7 cv.waitKey(0)
8 cv.destroyAllWindows("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище кодї комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

21.

```
1 import cv2 as cv
2 import sys
3 image_grayscale = cv.imread(sys.argv[1], -1)
4 cv.imwrite("image_grayscale.png", image_grayscale)
5 cv.namedWindow("Grayscale image", WINDOW_AUTOSIZE)
6 cv.imshow("Grayscale image", image_grayscale)
7 cv.waitKey(0)
8 cv.destroyAllWindows("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище коді комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.png" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображати у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

22.

```
1 import cv2
2 image_grayscale = cv2.imread("image.jpg", 0)
3 cv2.imwrite("image_grayscale.png", image_grayscale)
4 cv2.namedWindow("Grayscale image", WINDOW_AUTOSIZE)
5 cv2.imshow("Grayscale image", image_grayscale)
6 cv2.waitKey()
7 cv2.destroyWindow("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище коді комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7

23.

```
1 import cv2
2 image_grayscale = cv2.imread("image.jpg", cv2.IMAGE_GRAYSCALE)
3 cv2.imwrite("image_grayscale.jpg", image_grayscale)
4 cv2.createWindow("Grayscale image", WINDOW_AUTOSIZE)
5 cv2.imshow("Grayscale image", image_grayscale)
6 cv2.waitKey(0)
7 cv2.deleteWindow("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище кодї комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.jpg" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7

24.

```
1 import cv2
2 import sys
3 image_grayscale = cv2.imread(argv[0], cv2.IMREAD_GRAYSCALE)
4 cv2.imwrite("image_grayscale.webp", image_grayscale)
5 cv2.namedWindow("Grayscale image", WINDOW_AUTOSIZE)
6 cv2.imshow("Grayscale image", image_grayscale)
7 cv2.waitKey()
8 cv2.destroyAllWindows("Grayscale image")
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище кодї комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.webp" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відобразити у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

25.

```
1 import cv2
2 import sys
3 image_grayscale = cv2.imread(sys.argv[1], -1)
4 cv2.imprint("image_grayscale.webp", image_grayscale)
5 cv2.namedWindow("Grayscale image", cv2.WINDOW_AUTOSIZE)
6 cv2.imshow("Grayscale image", image_grayscale)
7 cv2.waitKey(0)
8 cv2.destroyAllWindows()
```

Вкажіть номери рядків, у яких допущено помилки, у наведеному вище коді комп'ютерної програми мовою Python, яка за допомогою функцій бібліотеки OpenCV повинна зчитувати зображення як одноканальне у відтінках сірого з файлу, що задається як параметр командного рядка, з поточної папки, виконувати збереження зображення у файлі з іменем "image_grayscale.webp" в поточній папці, створювати вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображати у цьому вікні зчитане зображення та знищувати створене вікно після натиснення будь-якої клавіші:

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8

Питання третього рівня складності (вказіть правильну відповідь):

1. У комп'ютерних програмах мовою програмування Python для імпорту бібліотеки OpenCV використовується директива:

2. У комп'ютерних програмах мовою програмування Python для зчитування зображень використовується функція бібліотеки OpenCV:

3. У комп'ютерних програмах мовою програмування Python для виконання збереження зображення використовується функція бібліотеки OpenCV:

4. У комп'ютерній програмі мовою програмування PYTHON для створення вікна з відповідною назвою та розміром, призначеного для відображення зчитаного зображення, використовується функція бібліотеки OpenCV:

5. У комп'ютерних програмах мовою програмування Python для відображення зображень використовується функція бібліотеки OpenCV:

6. У комп'ютерній програмі мовою програмування PYTHON для організації затримки до натиснення будь-якої клавіши для показу вікна, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

7. У комп'ютерній програмі мовою програмування PYTHON для знищення вікна, призначеного для відображення зображення, використовується функція бібліотеки OpenCV:

8. У комп'ютерній програмі мовою програмування PYTHON для знищення всіх вікон, призначених для відображення зображень, використовується функція бібліотеки OpenCV:

9. ...

```
image = cv.imread("image.webp", -1)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
10. import cv2 as cv
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як ϵ (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
11. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
...
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як ϵ (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
12. import cv2 as cv
image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
...
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою

функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
13. import cv2 as cv
    image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
    cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
    cv.imshow("My image", image)
    ...
    cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
14. import cv2 as cv
    image = cv.imread("image.webp", cv.IMREAD_UNCHANGED)
    cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
    cv.imshow("My image", image)
    cv.waitKey(0)
    ...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "webp.jpg" з поточної папки, створювала вікно з ім'ям "My

image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

```
15. import cv2 as cv
...
cv.namedWindow("My image", cv.WINDOW_AUTOSIZE)
cv.imshow("My image", image)
cv.waitKey(0)
cv.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп’ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу, що задається як параметр командного рядка, з поточної папки, створювала вікно з ім’ям “My image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіши:

```
16. ...
image = cv2.imread("image.webp", -1)
cv2.namedWindow("My image", WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп’ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім’ям “image.webp” з поточної папки, створювала вікно з ім’ям “My image”, автоматично встановивши його розмір у відповідності до розміру

зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
17. import cv2
...
cv2.namedWindow("My image", WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
18. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
...
cv2.imshow("My image", image)
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
19. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", WINDOW_AUTOSIZE)
...
cv2.waitKey(0)
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
20. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", WINDOW_AUTOSIZE)
cv2.imshow("My image", image)
...
cv2.destroyWindow("My image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "image.webp" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
21. import cv2
image = cv2.imread("image.webp", cv2.IMREAD_UNCHANGED)
cv2.namedWindow("My image", WINDOW_AUTOSIZE)
```

```
cv2.imshow("My image", image)
cv2.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як є (з альфа-каналом) з файлу з ім'ям "webp.jpg" з поточної папки, створювала вікно з ім'ям "My image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
22. import cv2 as cv
...
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
23. import cv2
...
cv2.namedWindow("Color image", cv2.WINDOW_AUTOSIZE)
cv2.imshow("Color image", image)
cv2.waitKey(0)
cv2.destroyWindow("Color image")
```


Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
24. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
...
cv.imshow("Color image", image)
cv.waitKey(0)
cv.destroyAllWindows("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
25. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
...
cv.waitKey(0)
cv.destroyAllWindows("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
26. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
...
cv.destroyWindow("Color image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з файлу з ім'ям "image.png" з поточної папки, створювала вікно з ім'ям "Color image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
27. import cv2 as cv
image = cv.imread("image.png", cv.IMREAD_COLOR)
cv.namedWindow("Color image", cv.WINDOW_AUTOSIZE)
cv.imshow("Color image", image)
cv.waitKey(0)
...
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як 3-канальне кольорове з

файлу з ім'ям “image.png” з поточної папки, створювала вікно з ім'ям “Color image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
28. import cv2 as cv
...
cv.imwrite("image_grayscale.jpg", image_grayscale)
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям “image.jpg” з поточної папки, виконувала збереження зображення у файлі з іменем “image_grayscale.jpg” в поточній папці, створювала вікно з ім'ям “Grayscale image”, автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

```
29. import cv2 as cv
image_grayscale = cv.imread("image.jpg", 0)
...
cv.namedWindow("Grayscale image", cv.WINDOW_AUTOSIZE)
cv.imshow("Grayscale image", image_grayscale)
cv.waitKey(0)
cv.destroyWindow("Grayscale image")
```

Для наведеного вище коду вкажіть рядок, який треба вставити на місці ..., щоб комп'ютерна програма мовою програмування Python за допомогою

функцій бібліотеки OpenCV зчитувала зображення як одноканальне у відтінках сірого з файлу з ім'ям "image.jpg" з поточної папки, виконувала збереження зображення в файлі з іменем "image_grayscale.jpg" у поточній папці, створювала вікно з ім'ям "Grayscale image", автоматично встановивши його розмір у відповідності до розміру зображення, відображала у цьому вікні зчитане зображення та знищувала створене вікно після натиснення будь-якої клавіші:

ЛІТЕРАТУРА

1. Download the latest version for Windows. Python 3.12.4. URL: <https://www.python.org/downloads/>
2. Методичні вказівки до виконання лабораторної роботи на тему «Читання, відображення та збереження зображень за допомогою функцій бібліотеки OpenCV» у Jupyter Notebook мовою програмування Python з освітньої компоненти «Цифрова обробка зображень» для здобувачів вищої освіти другого (магістерського) рівня за освітньо-професійною програмою «Прикладна математика» спеціальності 113 «Прикладна математика» і за освітньо-професійною програмою «Прикладна інформатика» спеціальності 122 «Комп'ютерні науки» денної та заочної форм навчання [Електронне видання] / Ю.Є. Климюк. Рівне : НУВГП, 2022. 88 с.
3. Chung Bryan WC. Pro Processing for Images and Computer Vision with OpenCV. Apress, 2017. 301 p.
4. Rosebrock Adrian. Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision. PyImageSearch, 2016. 166 p.
5. Szeliski Richard. Computer Vision: Algorithms and Applications. 2nd Edition. Springer, 2021. 1230 p.
6. Villán Alberto Fernández. Mastering OpenCV 4 with Python : A practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7. Packt Publishing : Birmingham – Mumbai, 2019. 650 p.