МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Костромской государственный университет»

(КГУ)

ИАСТ

Кафедра автоматизированных систем и технологий

09.03.02

Направление подготовки/Специальность Информационные системы и технологии

Дисциплина Технологии компьютерного зрения

# Лабораторная №0.

# Загрузка и сохранение изображений.

Выполнили студенты Копосов Лев Владимирович

Копосов Владимир Владимирович

Группа 22-ИСбо-1б

Проверил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_

Кострома

Вопросы:

1. Какой способ представления изображения был использован при выполнении работы?

При выполнении работы было использовано растровое хранение изображений в формате RGB. Каждый пиксель кодируется тремя значениями - интенсивностью красного, зеленого и синего цветов.

2. Как при использовании массивов numpy узнать их размер по той или иной оси?

При использовании массивов numpy, чтобы узнать их размер по той или иной оси, можно использовать атрибут shape. Пример реализации:

h, w, ch = image.shape.

3. Как создать пустой массив numpy размером Y строк и X столбцов, элементы которого являются байтами (0..255)?

Чтобы создать пустой массив numpy размером Y строк и X столбцов, элементы которого являются байтами (0..255), нужно использовать функцию np.zeros, с параметром dtype=np.uint8, чтобы был диапазон (0..255). Пример реализации:

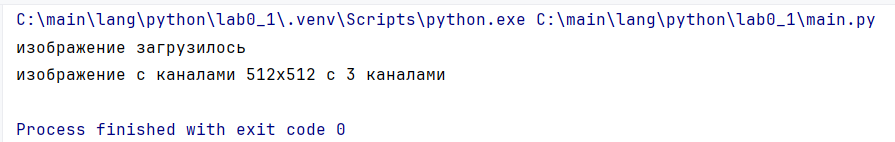
bytearray = np.zeros((Y, X), dtype=np.uint8).

Вывод в консоль.

Задание 1.

Создайте приложение на Python, которое загружает изображение из заданного файла и выводит в консоль его размер в формате "столбцы x строки".

Для загрузки изображения используйте функцию **cv2.imread()**. Укажите желаемую цветовую систему как **cv2.IMREAD\_COLOR**.



Задание 2.

Модифицируйте вашу программу так, чтобы загрузка изображения происходила в 2 этапа:

1. загрузка заданного файла как последовательности байт в массив numpy с помощью функции **numpy.fromfile()**.
2. разбор этой последовательности байт как изображения с помощью функции **cv2.imdecode()**. Укажите желаемую цветовую систему как **cv2.IMREAD\_COLOR**.

Выделите код загрузки в отдельную функцию, принимающую строку с путём к изображению и возвращающую объект изображения (массив numpy). Проверьте работу программы на файлах с именами в ascii (латиницей) и с кириллицей.



Задание 3.

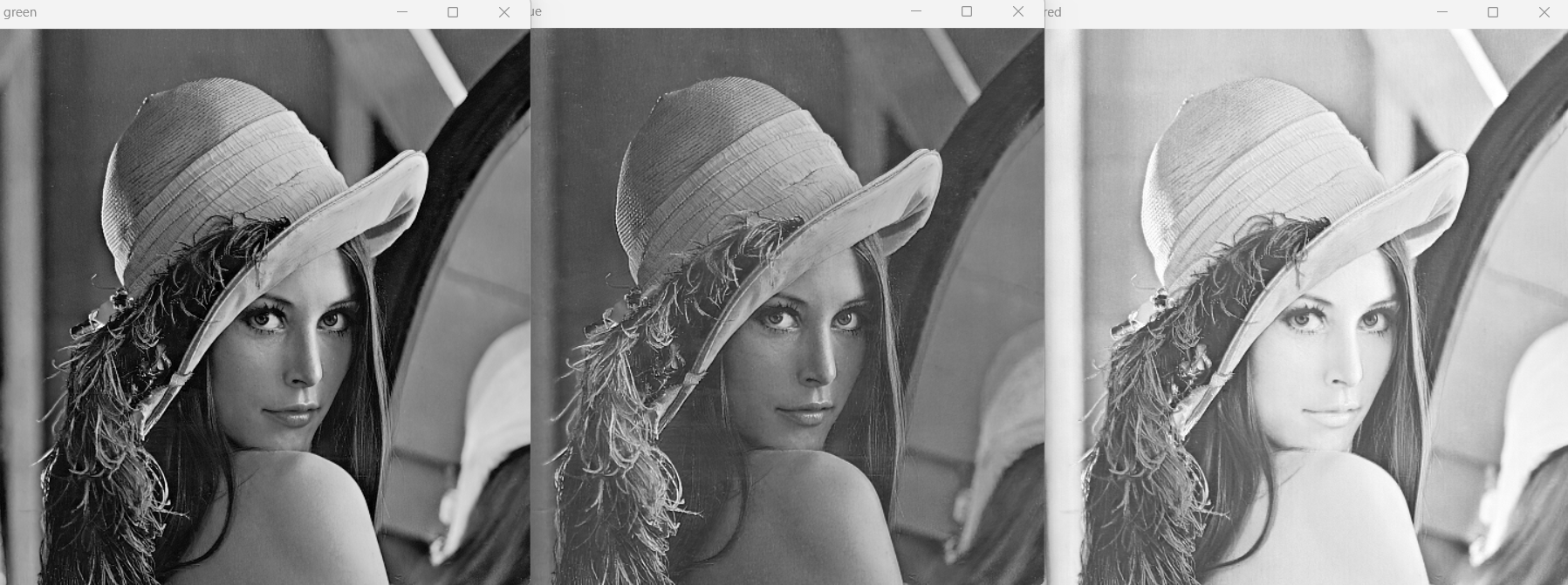
Модифицируйте вашу программу, чтобы после загрузки изображения оно было показано на экране с помощью функции **cv2.imshow()**. Чтобы задержать выполнение программы до нажатия клавиши, используйте функцию **cv2.waitKey()**. Имя окна должно совпадать с именем загруженного файла.



Задание 4.

Модифицируйте вашу программу следующим образом:

1. после загрузки изображения разделите его на три канала (0 - синий, 1 - зеленый, 2 - красный).
2. каждый канал выведите в отдельном окне с пометкой BLUE/GREEN/RED.
3. закодируйте каждый канал в содержимое файла формата ".png" с помощью функции **cv2.encode()** и запишите его в отдельный файл.



Дополнительное задание.

Используя функцию **cv2.selectROI()**, напишите программу, которая позволяет пользователю выделить прямоугольную область на изображении. Эта область должна быть сохранена в отдельный файл, имя которого вводится с клавиатуры.

