



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА №4

«Использование библиотек NumPy, SciPy, matplotlib для
вычислений»

ДИСЦИПЛИНА: «Перспективные языки программирования»

Выполнил: студент гр. ИУК4-32Б

Зудин (Подпись) (Зудин Д.В.)
(Ф.И.О.)

Проверил:

Пчелинцева (Подпись) (Пчелинцева Н.И.)
(Ф.И.О.)

Дата сдачи (защиты):

30.11.22

Результаты сдачи (защиты):

100

- Балльная оценка:

- Оценка:

заг-ено

Калуга, 2022 г.

Цель: формирование практических навыков процедурного программирования, разработки и отладки программ, овладение методами и средствами разработки и оформления технической документации.

Задачи:

1. Научиться загружать дополнительные библиотеки в среду разработки;
2. Изучить особенности и возможности библиотек NumPy, SciPy, Matplotlib и Pillow;
3. Изучить типовые алгоритмы решения задач с использованием дополнительных библиотек.

Вариант №1

Формулировка задания №1

Создайте матрицу размером 6х6, на границах которой расположены 1, внутри пространство заполнено 0. Создайте вторую матрицу размером 6х6, на границах которой расположены 0, внутри пространство заполнено 1. Перемножьте две созданные матрицы. Для каждой из трёх матриц определите ранг. Составьте круговую диаграмму, отражающую ранги каждой из матриц. От общей диаграммы отделите часть, соответствующую наибольшему значению ранга.

Листинг программы для задания №1

```
import numpy as np
import matplotlib.pyplot as plt

# Матрица 6х6 с 1 на границе и 0 внутри
matrix_1 = np.ones((6, 6))
matrix_1[1:-1, 1:-1] = 0
rank_1 = np.linalg.matrix_rank(matrix_1)
print("Первая матрица:")
print(matrix_1)
print(f"Ранг первой матрицы: {rank_1}")

# Матрица 6х6 с 0 на границе и 1 внутри
matrix_2 = np.zeros((6, 6))
matrix_2[1:-1, 1:-1] = 1
rank_2 = np.linalg.matrix_rank(matrix_2)
print("\nВторая матрица:")
```

```

print(matrix_2)
print(f"Ранг второй матрицы: {rank_2}")

# Третья матрица - произведение первых двух
matrix_3 = np.dot(matrix_1, matrix_2)
rank_3 = np.linalg.matrix_rank(matrix_3)
print("\nТретья матрица:")
print(matrix_3)
print(f"Ранг третьей матрицы: {rank_3}")

# Построение круговой диаграммы рангов матриц
labels = ['Матрица №1', 'Матрица №2', 'Матрица №3']
sizes = [rank_1, rank_2, rank_3]
max_index = sizes.index(max(sizes))
colors = ['red', 'green', 'blue']
explode = tuple([0 if i != max_index else 0.1 for i in range(len(sizes))])

def absolute_value(val):
    """Вычисление абсолютного значения из процентного"""
    a = int(np.round(val / 100 * np.array(sizes).sum(), 0))
    return a

plt.pie(sizes, explode=explode, labels=labels, colors=colors, shadow=True,
autopct=absolute_value, startangle=90)
plt.title("Круговая диаграмма рангов матриц")
plt.show()

```

Результат выполнения программы для задания №1

Первая матрица:

```

[[1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1. 1.]]

```

Ранг первой матрицы: 2

Вторая матрица:

```

[[0. 0. 0. 0. 0. 0.]
 [0. 1. 1. 1. 1. 0.]

```

```
[0. 1. 1. 1. 1. 0.]
[0. 1. 1. 1. 1. 0.]
[0. 1. 1. 1. 1. 0.]
[0. 0. 0. 0. 0. 0.]]
```

Ранг второй матрицы: 1

Третья матрица:

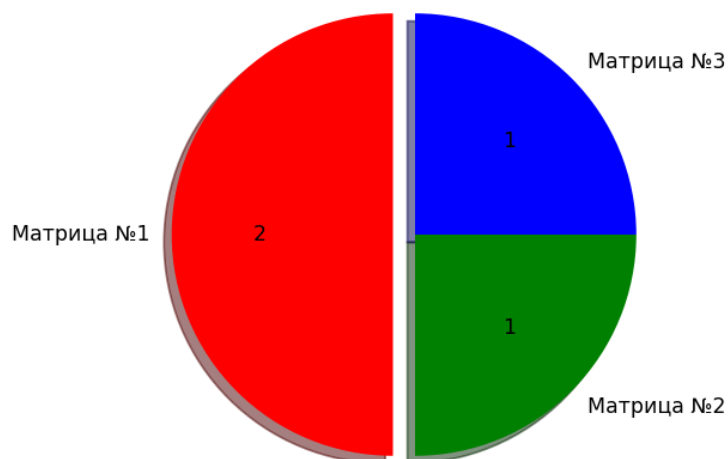
```
[[0. 4. 4. 4. 4. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0.]
 [0. 4. 4. 4. 4. 0.]]
```

Ранг третьей матрицы: 1

Figure 1

— □ ×

Круговая диаграмма рангов матриц



Формулировка задания №2

Дано изображение HOUSE.JPG размером 2560x1600 пикселей. Уменьшите изображение в 4 раза. Пересохраните.

Листинг программы для задания №2

```
from PIL import Image
```

```

image_path = 'House.jpg'
img = Image.open(image_path)
width, height = img.size
print(f"Изначальный размер изображения: {width} x {height}")

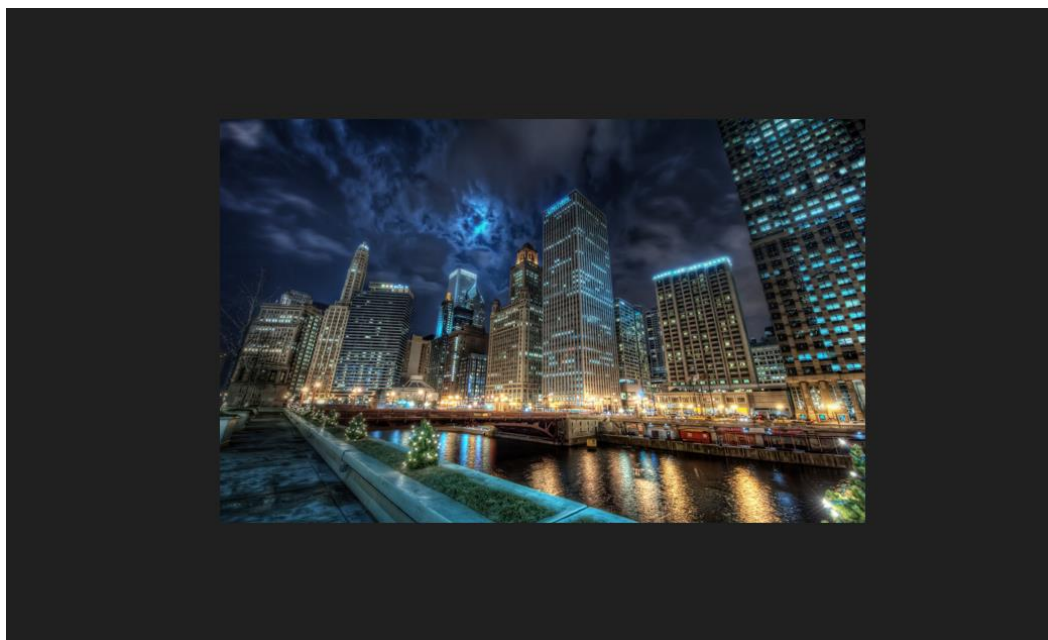
sizes = (int(width / 4), int(height / 4))
img.thumbnail(size=sizes)
width, height = img.size
print(f"Изменённый размер изображения: {width} x {height}")
img.show()
img.save('House.jpg')

```

Результат выполнения программы для задания №2

Изначальный размер изображения: 2560 x 1600

Изменённый размер изображения: 640 x 400



Формулировка задания №3

С помощью библиотек NumPy и Matplotlib постройте графики двух линейных функций – возрастающей и убывающей, и закрасьте образованные ими углы двумя разными цветами (т.е. вертикальные углы закрашиваются одним цветом). Цвета закрашенных углов не должны совпадать с цветами линий линейных функций.

Листинг программы для задания №3

```

import matplotlib.pyplot as plt
import numpy as np

```

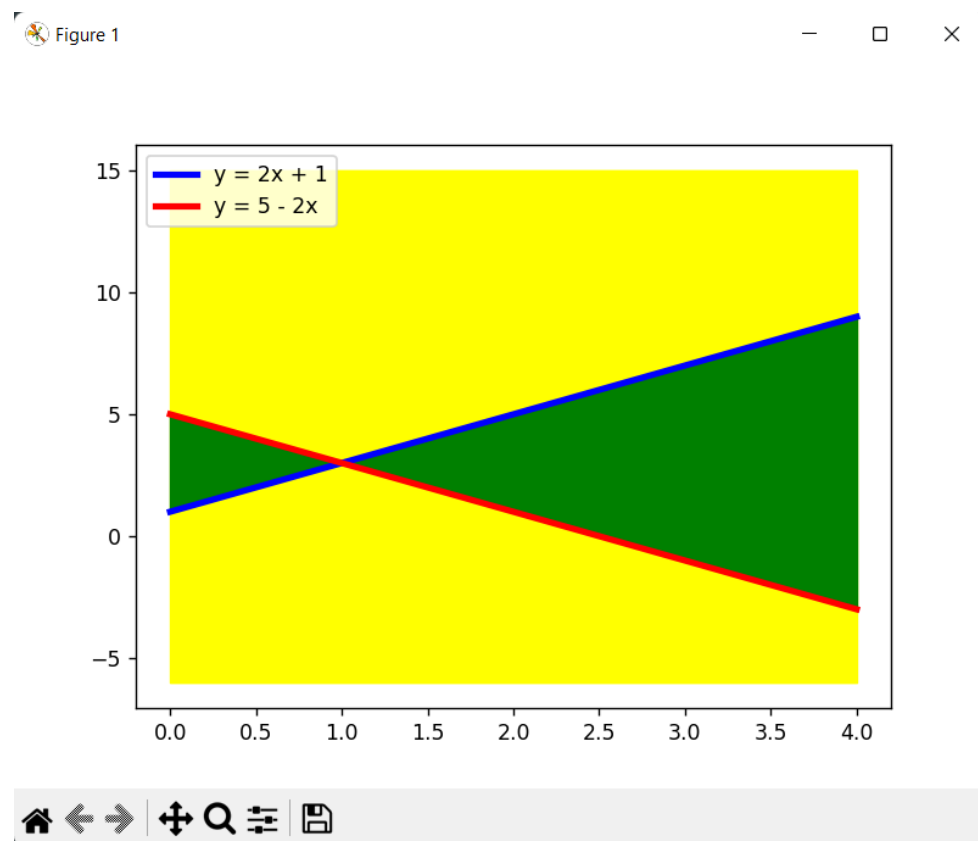
```

x = np.arange(0, 5)
y1 = 2 * x + 1
y2 = -2 * x + 5
fig, ax = plt.subplots()

ax.plot(x, y1, color='blue', label='y = 2x + 1', lw=3)
ax.plot(x, y2, color='red', label='y = 5 - 2x', lw=3)
ax.fill_between(x, y2, 9 + y2 + y1, color='yellow')
ax.fill_between(x, y2, 0 - y1 - y2, color='yellow')
ax.fill_between(x, y1, y2, color='green')
ax.legend(loc='upper left')
plt.show()

```

Результат выполнения программы для задания №3



Выводы:

В ходе работы были сформированы практические навыки процедурного программирования, разработки и отладки программ, овладения методами и средствами разработки и оформления технической документации.