

ЛАБОРАТОРНА РОБОТА №4

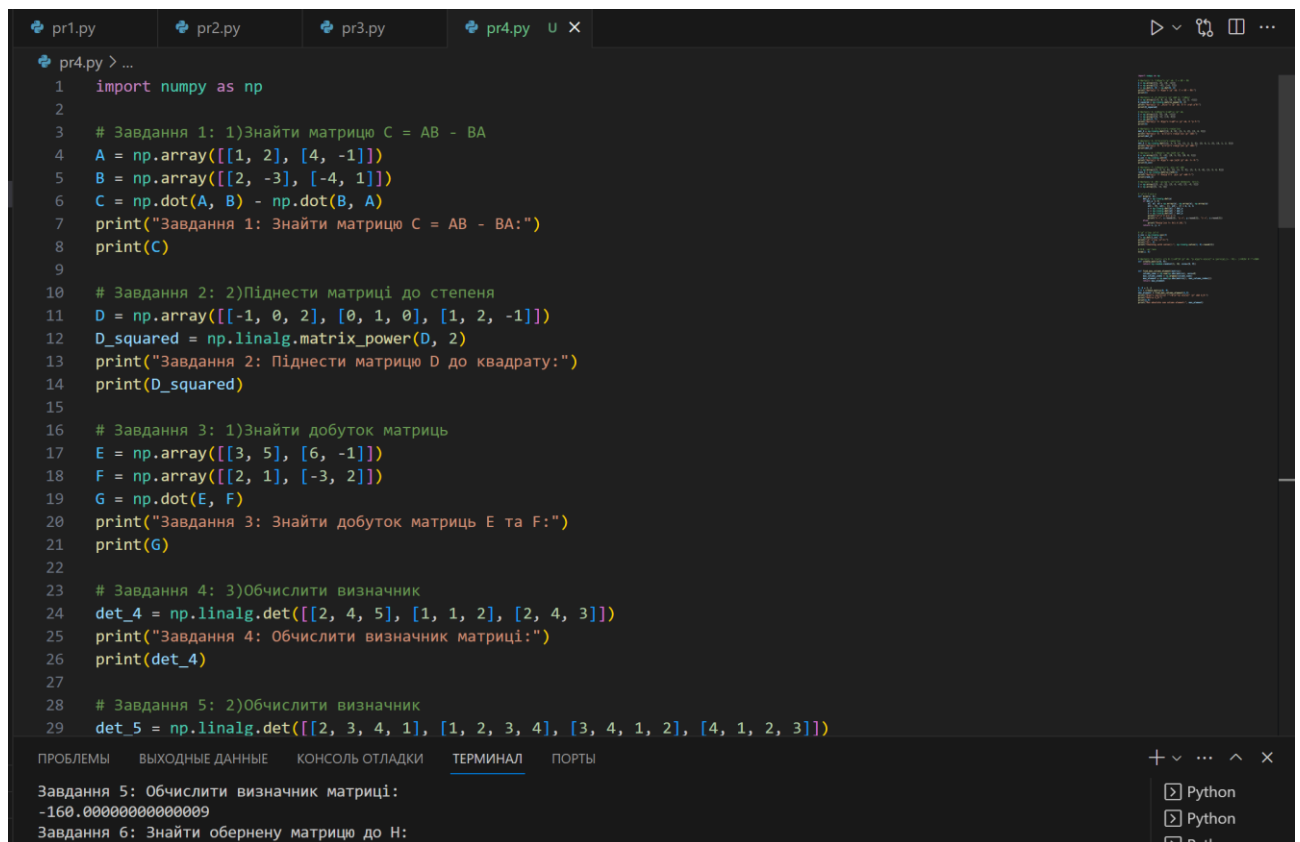
Тема: «Елементи лінійної алгебри. Розв'язання систем лінійних алгебраїчних рівнянь»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:



```
pr1.py pr2.py pr3.py pr4.py U X
pr4.py > ...
1 import numpy as np
2
3 # Завдання 1: 1)Знайти матрицю C = AB - BA
4 A = np.array([[1, 2], [4, -1]])
5 B = np.array([[2, -3], [-4, 1]])
6 C = np.dot(A, B) - np.dot(B, A)
7 print("Завдання 1: Знайти матрицю C = AB - BA:")
8 print(C)
9
10 # Завдання 2: 2)Піднести матриці до степеня
11 D = np.array([[ -1, 0, 2], [0, 1, 0], [1, 2, -1]])
12 D_squared = np.linalg.matrix_power(D, 2)
13 print("Завдання 2: Піднести матрицю D до квадрату:")
14 print(D_squared)
15
16 # Завдання 3: 1)Знайти добуток матриць
17 E = np.array([[3, 5], [6, -1]])
18 F = np.array([[2, 1], [-3, 2]])
19 G = np.dot(E, F)
20 print("Завдання 3: Знайти добуток матриць E та F:")
21 print(G)
22
23 # Завдання 4: 3)Обчислити визначник
24 det_4 = np.linalg.det([[2, 4, 5], [1, 1, 2], [2, 4, 3]])
25 print("Завдання 4: Обчислити визначник матриці:")
26 print(det_4)
27
28 # Завдання 5: 2)Обчислити визначник
29 det_5 = np.linalg.det([[2, 3, 4, 1], [1, 2, 3, 4], [3, 4, 1, 2], [4, 1, 2, 3]])
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Завдання 5: Обчислити визначник матриці:
-160.00000000000009

Завдання 6: Знайти обернену матрицю до H:

Python Python Python

```
pr1.py pr2.py pr3.py pr4.py u ●
pr4.py > ...
28 # Завдання 5: 2)Обчислити визначник
29 det_5 = np.linalg.det([[2, 3, 4, 1], [1, 2, 3, 4], [3, 4, 1, 2], [4, 1, 2, 3]])
30 print("Завдання 5: Обчислити визначник матриці:")
31 print(det_5)
32
33 # Завдання 6: 1)Знайти обернену матрицю
34 H = np.array([[1, 2, -3], [0, 1, 2], [0, 0, 1]])
35 H_inv = np.linalg.inv(H)
36 print("Завдання 6: Знайти обернену матрицю до H:")
37 print(H_inv)
38
39 # Завдання 7: 1)Визначити ранг матриці
40 I = np.array([[1, 2, 3, 4], [3, -1, 2, 5], [1, 2, 3, 4], [1, 3, 4, 5]])
41 rank_I = np.linalg.matrix_rank(I)
42 print("Завдання 7: Визначити ранг матриці I:")
43 print(rank_I)
44
45 # Завдання 8: 29)Розв'язати систему лінійних рівнянь
46 J = np.array([[2, -1, 1], [3, 4, -2], [1, -3, 1]])
47 K = np.array([5, -3, 4])
48
49
50 # Метод Крамера
51 def kram(a, b):
52     det_a = np.linalg.det(a)
53     if det_a != 0:
54         a1, a2, a3 = np.array(a), np.array(a), np.array(a)
55         a1[:, 0], a2[:, 1], a3[:, 2] = b, b, b
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Завдання 5: Обчислити визначник матриці:
-160.00000000000009

Завдання 6: Знайти обернену матрицю до H:

Python Python Python

```
pr1.py pr2.py pr3.py pr4.py u ●
pr4.py > ...
54     a1, a2, a3 = np.array(a), np.array(a), np.array(a)
55     a1[:, 0], a2[:, 1], a3[:, 2] = b, b, b
56     x = np.linalg.det(a1) / det_a
57     y = np.linalg.det(a2) / det_a
58     z = np.linalg.det(a3) / det_a
59     print("Метод Крамера:")
60     print("x =", x.round(3), "y =", y.round(3), "z =", z.round(3))
61 else:
62     print("Визначник дорівнює нулю")
63     return x, y, z
64
65
66 # Матричний метод
67 J_inv = np.linalg.inv(J)
68 X = np.dot(J_inv, K)
69 print("Матричний метод:")
70 print("X=", X)
71 print("Checking with solve():", np.linalg.solve(J, K).round(3))
72
73 # Всі розв'язки
74 kram(J, K)
75
76
77 # Завдання 9: 1)Створити прямокутну матрицю та знайти елемент з максимальною сумою модулів у стовпці
78 def create_matrix(N, M):
79     return np.random.randint(1, 10, size=(N, M))
80
81
82 def find_max_column_element(matrix):
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

Завдання 5: Обчислити визначник матриці:
-160.00000000000009

Завдання 6: Знайти обернену матрицю до H:

Python Python Python

```
pr1.py pr2.py pr3.py pr4.py U
pr4.py > ...
75
76
77 # Завдання 9: 1) Створити прямокутну матрицю та знайти елемент з максимальною сумою модулів у стовпці
78 def create_matrix(N, M):
79     return np.random.randint(1, 10, size=(N, M))
80
81
82 def find_max_column_element(matrix):
83     column_sums = np.sum(np.abs(matrix), axis=0)
84     max_column_index = np.argmax(column_sums)
85     max_element = np.max(np.abs(matrix[:, max_column_index]))
86     return max_element
87
88
89 N, M = 3, 4
90 A_9 = create_matrix(N, M)
91 max_element = find_max_column_element(A_9)
92 print("Знайти найменший стовпчастий елемент матриці A_9:")
93 print("Matrix A_9:")
94 print(A_9)
95 print("Max absolute sum column element:", max_element)
96
```

ПРОБЛЕМИ ВИХОДНІ ДАННІ КОНСОЛЬ ОТЛАДКИ **ТЕРМИНАЛ** ПОРТЫ

Завдання 5: Обчислити визначник матриці:
-160.00000000000009

Завдання 6: Знайти обернену матрицю до H:

Python Python

```
pr1.py pr2.py pr3.py pr4.py U
pr4.py > ...
ПРОБЛЕМИ ВИХОДНІ ДАННІ КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ

```
PS C:\Users\38066\Desktop\chiselni-metody> & C:/Python311/python.exe c:/Users/38066/Desktop/chiselni-metody/pr4.py
Завдання 1: Знайти матрицю C = AB - BA:
[[4 -8]
 [12 -4]]
Завдання 2: Піднести матрицю D до квадрату:
[[3 4 -4]
 [0 1 0]
 [-2 0 3]]
Завдання 3: Знайти добуток матриць E та F:
[[-9 13]
 [15 4]]
Завдання 4: Обчислити визначник матриці:
4.0
Завдання 5: Обчислити визначник матриці:
-160.00000000000009
Завдання 6: Знайти обернену матрицю до H:
[[1. -2. 7.]
 [0. 1. -2.]
 [0. 0. 1.]]
Завдання 7: Визначити ранг матриці I:
2
Матричний метод:
X= [1.0000000e+00 -4.4408921e-16 3.0000000e+00]
Checking with solve(): [1. -0. 3.]
Метод Крамера:
x = 1.0 y = 0.0 z = 3.0
Знайти найменший стовпчастий елемент матриці A_9:
Matrix A_9:
[[5 1 2 1]
 [3 6 2 8]
 [2 7 7 5]]
Max absolute sum column element: 7
PS C:\Users\38066\Desktop\chiselni-metody>
```



Python Python Python Python Python Python Python


```

Код зі скріншотів:

```
import numpy as np

# Завдання 1: 1) Знайти матрицю C = AB - BA
A = np.array([[1, 2], [4, -1]])
B = np.array([[2, -3], [-4, 1]])
```

```

C = np.dot(A, B) - np.dot(B, A)
print("Завдання 1: Знайти матрицю C = AB - BA:")
print(C)

# Завдання 2: 2)Піднести матриці до степеня
D = np.array([[ -1, 0, 2], [0, 1, 0], [1, 2, -1]])
D_squared = np.linalg.matrix_power(D, 2)
print("Завдання 2: Піднести матрицю D до квадрату:")
print(D_squared)

# Завдання 3: 1)Знайти добуток матриць
E = np.array([[3, 5], [6, -1]])
F = np.array([[2, 1], [-3, 2]])
G = np.dot(E, F)
print("Завдання 3: Знайти добуток матриць E та F:")
print(G)

# Завдання 4: 3)Обчислити визначник
det_4 = np.linalg.det([[2, 4, 5], [1, 1, 2], [2, 4, 3]])
print("Завдання 4: Обчислити визначник матриці:")
print(det_4)

# Завдання 5: 2)Обчислити визначник
det_5 = np.linalg.det([[2, 3, 4, 1], [1, 2, 3, 4], [3, 4, 1, 2], [4, 1, 2, 3]])
print("Завдання 5: Обчислити визначник матриці:")
print(det_5)

# Завдання 6: 1)Знайти обернену матрицю
H = np.array([[1, 2, -3], [0, 1, 2], [0, 0, 1]])
H_inv = np.linalg.inv(H)
print("Завдання 6: Знайти обернену матрицю до H:")
print(H_inv)

# Завдання 7: 1)Визначити ранг матриці
I = np.array([[1, 2, 3, 4], [3, -1, 2, 5], [1, 2, 3, 4], [1, 3, 4, 5]])
rank_I = np.linalg.matrix_rank(I)
print("Завдання 7: Визначити ранг матриці I:")
print(rank_I)

# Завдання 8: 29)Розв'язати систему лінійних рівнянь
J = np.array([[2, -1, 1], [3, 4, -2], [1, -3, 1]])
K = np.array([5, -3, 4])

# Метод Крамера
def kram(a, b):
    det_a = np.linalg.det(a)
    if det_a != 0:
        a1, a2, a3 = np.array(a), np.array(a), np.array(a)
        a1[:, 0], a2[:, 1], a3[:, 2] = b, b, b
        x = np.linalg.det(a1) / det_a
        y = np.linalg.det(a2) / det_a
        z = np.linalg.det(a3) / det_a

```

```

        print("Метод Крамера:")
        print("x =", x.round(3), "y =", y.round(3), "z =", z.round(3))
    else:
        print("Визначник дорівнює нулю")
    return x, y, z

# Матричний метод
J_inv = np.linalg.inv(J)
X = np.dot(J_inv, K)
print("Матричний метод:")
print("X=", X)
print("Checking with solve():", np.linalg.solve(J, K).round(3))

# Всі розв'язки
kram(J, K)

# Завдання 9: 1)Створити прямокутну матрицю та знайти елемент з максимальною сумою
модулів у стовпці
def create_matrix(N, M):
    return np.random.randint(1, 10, size=(N, M))

def find_max_column_element(matrix):
    column_sums = np.sum(np.abs(matrix), axis=0)
    max_column_index = np.argmax(column_sums)
    max_element = np.max(np.abs(matrix[:, max_column_index]))
    return max_element

N, M = 3, 4
A_9 = create_matrix(N, M)
max_element = find_max_column_element(A_9)
print("Знайти найменший стовпчастий елемент матриці A_9:")
print("Matrix A_9:")
print(A_9)
print("Max absolute sum column element:", max_element)

```