

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра програмних систем і технологій

Дисципліна
« Спеціалізоване програмування автоматизованих систем»

Практичне заняття №2
на тему:
«Масиви та лінійна алгебра»

Виконав:	Якімечко Артем Анатолійович	Перевірила:	Ковтун Оксана Іванівна
Група	ІПЗ-31	Дата перевірки	
Форма навчання	Денна	Оцінка	
Спеціальність	121		
2021			

- **Основні теоретичні відомості**

Numpy — бібліотека мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Будь-який алгоритм, який може бути виражений в основному як послідовність операцій над масивами і матрицями, працює так само швидко, як еквівалентний код, написаний на C. В роботі застосовується підмодуль linalg (лінійна алгебра) для роботи із матрицями.

- **Постановка задачі**

2.1. Знайдіть $f(A)$, якщо:

2.1.3. $A = \begin{pmatrix} 2 & 1 \\ 0 & -1 \end{pmatrix}$, $f(x) = x^3 + x^2 - 7x + 3$.

2.2. Розв'яжіть матричні рівняння.

2.2.3. $A \cdot X \cdot B = C$, $A = \begin{pmatrix} 2 & 3 \\ 5 & 8 \end{pmatrix}$, $B = \begin{pmatrix} 5 & 4 & 1 \\ 1 & 1 & 7 \\ 6 & 5 & 9 \end{pmatrix}$, $C = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$

2.3. Знайдіть обернену матрицю A^{-1} , якщо:

2.3.3. $A = \begin{pmatrix} -1 & 2 & -2 \\ -3 & 5 & -4 \\ 1 & 0 & 1 \end{pmatrix}$.

2.4. Знайдіть ранг матриці.

2.4.3. $\begin{pmatrix} -2 & 2 & 1 & -1 & -3 \\ 3 & 1 & 0 & 1 & 0 \\ -1 & 1 & 2 & 2 & 1 \\ 0 & 4 & 3 & 2 & -2 \end{pmatrix}$.

- **Хід роботи**

```
• import numpy as np

def task_2_1():
    A = np.array([[2, 1], [0, -1]])

    f = np.linalg.matrix_power(A, 3) + np.linalg.matrix_power(A, 2) -
    np.dot(A, 7) + np.diag(np.array([3]))

    print('task2_1 \n', f, '\n')

def task_2_2():
    A = np.array([[2, 3],
                  [5, 8]])

    B = np.array([
        [5, 4, 1],
        [1, 1, 7],
        [6, 5, 9]])

    C = np.array([
        [1, 1, 0],
        [0, 1, 1]])

    A_invert = np.linalg.inv(A)
    B_invert = np.linalg.inv(B)

    result = np.dot(np.dot(A_invert, C), B_invert)

    #X = 1/A * C * 1/B

    print('task2_2 \n', result, '\n')

def task_2_3():
    A = np.array([
        [-1, 2, -2],
        [-3, 5, -4],
        [1, 0, 1]])

    A_inv = np.linalg.inv(A)

    print('task2_3 \n', A_inv, '\n')

def task_2_4():
    A = np.array([
        [-2, 2, 1, -1, -3],
        [3, 1, 0, 1, 0],
        [-1, 1, 2, 2, 1],
        [0, 4, 3, 2, -2]])

    rank = np.linalg.matrix_rank(A)

    print('task2_4 \n', rank, '\n')
```

- **Результати виконання**

```
task2_1
[[ 1  0]
 [ 3 10]]

task2_2
[[-40. -50.  43.]
 [ 29.  36. -31.]]

task2_3
[[ 1.66666667 -0.66666667  0.66666667]
 [-0.33333333  0.33333333  0.66666667]
 [-1.66666667  0.66666667  0.33333333]]

task2_4
3
```

- **Висновок**

Виконання лабораторного завдання закріпило навички роботи із модулем для роботи із масивами Numpy мови програмування Python. Для обчислень використовувались вбудовані у модуль методи лінійної алгебри. В ході паралельного виконання були помічені розбіжності у результатах виконання деяких обчислень.