

Приклад скрипту до роботи №1

```
import sys
import numpy as np
from numpy import linalg as LA
print("Приклад роботи № 1")
try:
    n=int(input('вкажіть розмір квадратної матриці'))
except ValueError:
    print('\n*** Це не число. Аварійне завершення...')
    input("pres any key...")
    sys.exit()
a=np.matrix(np.random.rand(n, n))
print('a=\n', a)
a=np.round_(a,2)
print('a=\n', a)
filename='file_lab1.txt'
np.savetxt(filename,a,fmt='%-10.2f', header='матриця a', footer='-----')
print("виведення блоку вихідної матриці")
print('блок задається "адресою" верхнього лівого елемента матриці i j')
print("i кількістю рядків i стовпців, що виводяться kstr krow" )
try:
    i1, j1 = map(int, input('вкажіть значення ij верхнього лівого елемента матриці').split())
    kstr, krow = map(int, input('вкажіть кількість рядків, що виводяться, i стовпців kstr krow').split())
except ValueError:
    print('***\nЦе не числа. Організується виведення блоку a[0,0] 2x2 ...\n')
    i1 = 0; j1=0; kstr = 2; krow=2
f = open(filename, 'at')

print('виведення блоку матриці починаючи з a[', i1, ', ', j1, '] розміру ', kstr, 'x', krow, sep="")
print(a[i1:i1+kstr,j1:j1+krow])
```

```

print('виведення блоку матриці починаючи з a[' , i1, ', ', j1, '] розміру ', kstr, 'x', krow, sep="",
file=f)
print(a[i1:i1+kstr,j1:j1+krow], file=f)

det=np.linalg.det(a)
print("Визначник матриці", det)
print("Визначник матриці", det, file=f)
if abs(det) < 1e-8:
    print("вихідна матриця близька до виродженої .....")
    sys.exit()
a1=np.linalg.inv(a)
print("Зворотна матриця")
for i in range(0,n):
    for j in range(0,n):
        print('%-10.2f' % a1[i,j], end="")
    print()

print("Зворотна матриця", file=f)
for i in range(0,n):
    for j in range(0,n):
        print('%-10.2f' % a1[i,j], end="", file=f)
    print( file = f)

a1=np.zeros((n,n))
np.matmul(a,a1, out=a1)

print("твір вихідної та зворотної матриці")
for i in range(0,n):
    for j in range(0,n):
        print('%-10.2f' % a1[i,j], end="")
    print()
print("твір вихідної та зворотної матриці", file=f)
for i in range(0,n):

```

```

        for j in range(0,n):
            print('%-10.2f' % aal[i,j], end="", file=f)
        print( file = f)

kcond=np.linalg.cond(a) число обумовленості матриці.
print("Число обумовленості матриці", kcond)
print("Кількість обумовленості матриці", kcond, file=f)

print("qr - розкладання матриці a")
print("qr - розкладання матриці a", file = f)
q=np.zeros((n,n)); r=np.zeros((n,n))
q,r = np.linalg.qr(a)
print("матриці q\n", q)
print("матриці q\n", q, file=f)
print("матриці r\n", r)
print("матриці r\n", r, file=f)
axx=np.matmul(q,r)
print("твір матриці q*r\n", axx)
print("Початкова матриця a = q * r \ n", a)
print("твір матриці q*r\n", axx, file=f)
print("Початкова матриця a = q * r \ n", a, file = f)

# Власні значення матриці.
w,v = LA.eig(a)
print('Власні числа матриці a\n', w)
print('Власні вектори матриці a\n', v)

print('Власні числа матриці a\n', w, file=f)
print('Власні вектори матриці a\n', v, file=f)

print( ''' перевірка:
Власне число квадратної матриці a
та її власний вектор
повинні бути пов'язані рівністю

```

```

np.dot(a, v) = w*v.
'''
print( ''' перевірка:
Власне число квадратної матриці a
та її власний вектор
повинні бути пов'язані рівністю
np.dot(a, v) = w*v.
''', file = f)
x1=np.dot(a,v)
x2=w*v
print('np.dot(a,v) =\n', x1)
print('w * v = \ n', x2)

print('np.dot(a,v) =\n', x1, file=f)
print( w * v = \ n, x2, file = f)

print("|dot(a,v) - w*v| < 1e-8\n",np.abs(x1-x2)< 1e-8)
print("|dot(a,v) - w*v| < 1e-8\n", np.abs(x1-x2)

f.close()
input("скрипт завершений\npres any key...")

```