

test-01_matrix

October 16, 2020

1 Analiza i projektiranje računalom: 1. laboratorijska vježba

1.1 Test klase Matrix

1.1.1 Priprema

```
[1]: import os

CD_KEY = "--MATRIX_TEST_IN_ROOT"

[2]: if (
    CD_KEY not in os.environ
    or os.environ[CD_KEY] is None
    or len(os.environ[CD_KEY]) == 0
    or os.environ[CD_KEY] == "false"
):
    %cd ..
else:
    print(os.getcwd())

os.environ[CD_KEY] = "true"
```

/mnt/data/projekti/faks/AIPR/dz/dz-01

1.1.2 Učitavanje paketa

```
[3]: import numpy as np

from src.matrices.matrix import Matrix
```

1.1.3 Provjere pristupa

```
[4]: SHAPE = (5, 5)
DTYPE = float
FILL_VALUE = 17.29

[5]: our_matrix = Matrix.full(5, 5, fill_value=FILL_VALUE, dtype=DTYPE)
print(our_matrix)
```

```
[
    [17.290, 17.290, 17.290, 17.290, 17.290],
    [17.290, 17.290, 17.290, 17.290, 17.290],
    [17.290, 17.290, 17.290, 17.290, 17.290],
    [17.290, 17.290, 17.290, 17.290, 17.290],
    [17.290, 17.290, 17.290, 17.290, 17.290]
]
```

Provjera manipulacije elemenata

```
[6]: our_matrix[1][3] = 941

print(f"Oblik: {our_matrix.shape}")
print(f"Matrica: {our_matrix}")
```

```
Oblik: (5, 5)
Matrica: [
    [ 17.290,  17.290,  17.290,  17.290,  17.290],
    [ 17.290,  17.290,  17.290, 941.000,  17.290],
    [ 17.290,  17.290,  17.290,  17.290,  17.290],
    [ 17.290,  17.290,  17.290,  17.290,  17.290],
    [ 17.290,  17.290,  17.290,  17.290,  17.290]
]
```

```
[7]: our_matrix.int()
print(f"Matrica pretvorena u cijele brojeve: {our_matrix}")
```

```
Matrica pretvorena u cijele brojeve: [
    [ 17,  17,  17,  17,  17],
    [ 17,  17,  17, 941,  17],
    [ 17,  17,  17,  17,  17],
    [ 17,  17,  17,  17,  17],
    [ 17,  17,  17,  17,  17]
]
```

```
[8]: our_matrix.float()
print(f"Matrica pretvorena natrag u float: {our_matrix}")
```

```
Matrica pretvorena natrag u float: [
    [ 17.000,  17.000,  17.000,  17.000,  17.000],
    [ 17.000,  17.000,  17.000, 941.000,  17.000],
    [ 17.000,  17.000,  17.000,  17.000,  17.000],
    [ 17.000,  17.000,  17.000,  17.000,  17.000],
    [ 17.000,  17.000,  17.000,  17.000,  17.000]
]
```

```
[9]: print(f"Matrica nula: {Matrix.zeros(3, 3)}")
```

```
Matrica nula: [
```

```

    [0.000, 0.000, 0.000],
    [0.000, 0.000, 0.000],
    [0.000, 0.000, 0.000]
]

```

```
[10]: print(f"Jedinična matrica: {Matrix.eye(5, 6, int)}")
```

```

Jedinična matrica: [
  [1, 0, 0, 0, 0, 0],
  [0, 1, 0, 0, 0, 0],
  [0, 0, 1, 0, 0, 0],
  [0, 0, 0, 1, 0, 0],
  [0, 0, 0, 0, 1, 0]
]

```

Provjera podmatrica

```
[11]: MATRIX_TO_USE = [
      [1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]
]

numpy_matrix_2 = np.array(MATRIX_TO_USE, dtype=float)
our_matrix_2 = Matrix.from_array(MATRIX_TO_USE)

```

```
[12]: print(our_matrix_2.diagonal())
```

```
[[1, 5, 9]]
```

```
[13]: print(our_matrix_2.reverse_diagonal())
```

```
[[3, 5, 7]]
```

```
[14]: print(our_matrix_2.row(1))
```

```
[[4, 5, 6]]
```

```
[15]: print(our_matrix_2.column(1))
```

```

[
  [2],
  [5],
  [8]
]

```

1.1.4 Provjera aritmetike

Zbrajanje

```
[16]: base_arithmetic_matrix = Matrix.from_array(  
      [  
          [1, 2, 3],  
          [4, 5, 6],  
          [7, 8, 9]  
      ]  
      )
```

```
[17]: print(base_arithmetic_matrix + 1)
```

```
[  
  [ 2,  3,  4],  
  [ 5,  6,  7],  
  [ 8,  9, 10]  
]
```

```
[18]: print(base_arithmetic_matrix + 3.14)
```

```
[  
  [ 4.140,  5.140,  6.140],  
  [ 7.140,  8.140,  9.140],  
  [10.140, 11.140, 12.140]  
]
```

```
[19]: print(base_arithmetic_matrix + Matrix.eye(3, 3, int))
```

```
[  
  [ 2,  2,  3],  
  [ 4,  6,  6],  
  [ 7,  8, 10]  
]
```

Oduzimanje

```
[20]: print(base_arithmetic_matrix - 1)
```

```
[  
  [0, 1, 2],  
  [3, 4, 5],  
  [6, 7, 8]  
]
```

```
[21]: print(base_arithmetic_matrix - 3.14)
```

```
[  
  [-2.140, -1.140, -0.140],  
  [ 0.860,  1.860,  2.860],  
  [ 3.860,  4.860,  5.860]  
]
```

```
[22]: print(base_arithmetic_matrix - Matrix.eye(3, 3, int))
```

```
[
  [0, 2, 3],
  [4, 4, 6],
  [7, 8, 8]
]
```

Množenje

```
[23]: print(base_arithmetic_matrix * 3)
```

```
[
  [ 3,  6,  9],
  [12, 15, 18],
  [21, 24, 27]
]
```

```
[24]: print(base_arithmetic_matrix * 3.14)
```

```
[
  [ 3.140,  6.280,  9.420],
  [12.560, 15.700, 18.840],
  [21.980, 25.120, 28.260]
]
```

```
[25]: print(base_arithmetic_matrix * Matrix.eye(3, 3, int))
```

```
[
  [1, 0, 0],
  [0, 5, 0],
  [0, 0, 9]
]
```

Dijeljenje

```
[26]: print(base_arithmetic_matrix / 3)
```

```
[
  [0.333, 0.667, 1.000],
  [1.333, 1.667, 2.000],
  [2.333, 2.667, 3.000]
]
```

```
[27]: print(base_arithmetic_matrix / 3.14)
```

```
[
  [0.318, 0.637, 0.955],
  [1.274, 1.592, 1.911],
]
```

```
[2.229, 2.548, 2.866]  
]
```

```
[28]: print(base_arithmetic_matrix / base_arithmetic_matrix)
```

```
[  
  [1.000, 1.000, 1.000],  
  [1.000, 1.000, 1.000],  
  [1.000, 1.000, 1.000]  
]
```