

Programmeerimine

10. loeng

Täna loengus

- Mitmemõõtmelised massiivid
- Listide muteerimine ja aliased

Kahemõõtmelised massiivid

Kahemõõtmelised massiivid

Mitmemõõtmelised massiivid

- Massiivid võivad olla mitmemõõtmelised.
 - Kahemõõtmeline = massiiv massiividest.
 - Kolmemõõtmeline = massiiv kahemõõtmelistest massiividest.
 - ...
- Üldjuhul võivad sisemised massiivid olla erineva suurusega.
- **Maatriks** on kahemõõtmeline massiiv, kus kõik sisemised massiivid on sama pikkusega.

Kahemõõtmelised massiivid



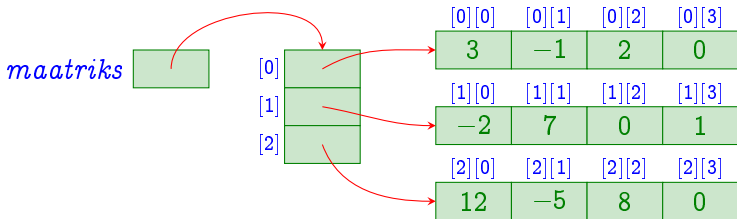
```
maatriks = [ [ 3, -1, 2, 0 ],  
               [ -2, 7, 0, 1 ],  
               [ 12, -5, 8, 0 ]  
             ]  
maatriks[1][2] = maatriks[0][0] - 2
```

Kahemõõtmelised massiivid

$$\text{maatriks} = \begin{bmatrix} [3, -1, 2, 0], \\ [-2, 7, 0, 1], \\ [12, -5, 8, 0] \end{bmatrix}$$



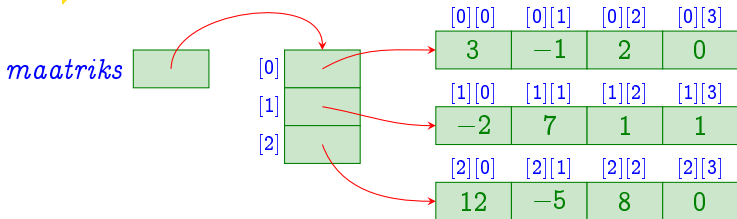
$$\text{maatriks}[1][2] = \text{maatriks}[0][0] - 2$$



Kahemõõtmelised massiivid

$$\text{maatriks} = \begin{bmatrix} [3, -1, 2, 0], \\ [-2, 7, 0, 1], \\ [12, -5, 8, 0] \end{bmatrix}$$

$$\text{maatriks}[1][2] = \text{maatriks}[0][0] - 2$$



Kahemõõtmelised massiivid

Maatriksi väljastamine ridade kaupa

```
def printMatrixRows(a):  
    for row in a:  
        for col in row:  
            print(col, end=' ')  
        print()
```

Näide

```
A = [[1,2,2], [7,5,9]]  
printMatrixRows(A)      # Trükkib: 1 2 2  
                        #          7 5 9
```

Listide muteerimine

Listide muteerimine

Listide muteerimine ja aliased

- Kuna listimuutuja on viit listile, siis ühe listimuutuja omistamisel teisele luuakse **alias**.
- Ühe listimuutuja kaudu listi muutmisel "muteerub" ka teine.
- Ka protseduuri parameetrina antud list edastatakse viida kaudu.
- Seetõttu on protseduuri sees tehtavad argumentlisti muutmised nähtavad ka protseduuri väljakutse kohas.

Listide muteerimine



$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$

Listide mureerimine



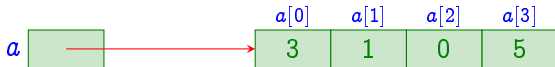
$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$



Listide mutesimine



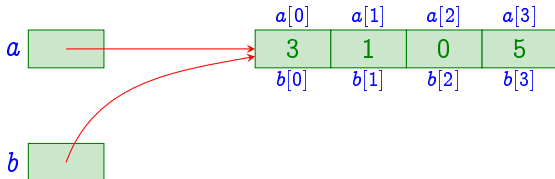
$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$



Listide mütterimine

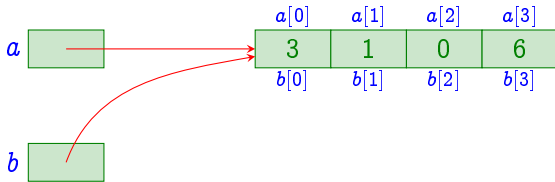
$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$



Listide mutesimine

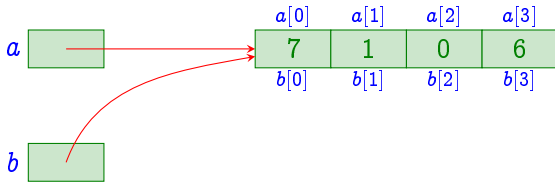
$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$



Listide mouteerimine

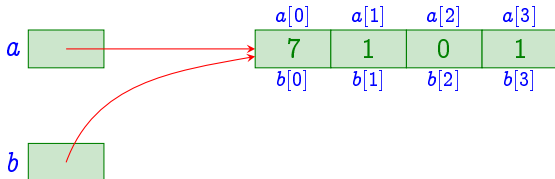
$a = [3, 1, 0, 5]$

$b = a$

$b[3] = 6$

$b[0] = a[3] + 1$

$a[3] = 1$



Listide muteerimine

Listide kopeerimine

- Listide kopeerimiseks võib kasutada *viilutamist* või *tüübiteisendust*.
- Mõlemal juhul kopeeritakse aineult kõige ülemine tase!
 - nn. **madal** (ingl. *shallow*) koopia.

Näide

```
a = [1, 2, [3, 4]]  
b = a[:]  
c = list(a)
```

```
a[0] = 0           # a=[0, 2, [3,4]]  
                  # b=[1, 2, [3,4]]; c=[1, 2, [3,4]]  
b[2][0] = 1        # a=[0, 2, [1,4]]  
                  # b=[1, 2, [1,4]]; c=[1, 2, [1,4]]
```

Listide muteerimine

Listide kopeerimine

- Listi **sügavaks** (ingl. *deep*) kopeerimiseks saab kasutada funktsiooni *deepcopy*.
 - on eeldefineeritud moodulis *copy*.

Näide

```
import copy
```

```
a = [1, 2, [3, 4]]
```

```
b = copy.deepcopy(a)
```

```
a[0] = 0           # a=[0, 2, [3,4]]
```

```
# b=[1, 2, [3,4]]
```

```
b[2][0] = 1        # a=[0, 2, [3,4]]
```

```
# b=[1, 2, [1,4]]
```

Suur tänu osalemast

ja

kohtumiseni!