

Lógica de Programação II

Matrizes

Matrizes

- Variável composta **multidimensional**
 - É equivalente a um vetor, contudo permite a utilização de diversas dimensões acessadas via diferentes índices
 - Pode ser pensada como um vetor cujo tipo é outro vetor, recursivamente
 - Em diversas situações matrizes são necessárias para correlacionar informações

Exemplo Motivacional

- Assumindo que **um aluno é avaliado com cinco notas**, seria necessário um vetor de três posições para guardar as notas de um aluno...

	0	1	2	3	4
notas	10.0	7.0	9.0	5.5	6.0

E se fossem 3 alunos?

- Contudo, assumindo que **uma turma tem três alunos**, seria necessária uma matriz bidimensional para guardar as notas de todos os alunos de uma turma...

notas

	0	1	2	3	4
0	5.0	4.5	7.0	5.2	6.1
1	2.1	6.5	8.0	7.0	6.7
2	8.6	7.0	9.1	8.7	9.3

alunos

```
turma = [[5.0, 4.5, 7.0, 5.2, 6.1], [2.1, 6.5, 8.0, 7.0, 6.7], [8.6, 7.0, 9.1, 8.7, 9.3]]
```

Exemplo motivacional

- Na verdade, na memória turma seria algo assim...

0	5.0
1	4.5
2	7.0
3	5.2
4	6.1
5	

...

0	
1	7.0
2	9.1
3	8.7
4	9.3

Acesso aos valores: [linha][coluna]

- Segunda nota do primeiro aluno

```
>>> turma[0][1]
```

```
4.5
```

- Quinta nota do terceiro aluno

```
>>> turma[2][4]
```

```
9.3
```

		notas				
		0	1	2	3	4
alunos	0	5.0	4.5	7.0	5.2	6.1
	1	2.1	6.5	8.0	7.0	6.7
	2	8.6	7.0	9.1	8.7	9.3

Calcular a média da turma

```
turma = [[5.0, 4.5, 7.0, 5.2, 6.1], [2.1, 6.5, 8.0, 7.0, 6.7], [8.6, 7.0, 9.1, 8.7, 9.3]]
#calcula a média
media = 0
#for para percorrer as linhas
for i in range(3):
    #for para percorrer as colunas
    for j in range(5):
        media = media + turma[i][j]
media = media / 15
print(media)
```

Preencher a matriz por leitura

```
turma = []
for i in range(3):
    # cria linha vazia
    linha = []
    for j in range(5):
        #vai adicionando as notas na linha
        linha.append(eval(input('Digite a
nota[' + str(i) + ', ' + str(j) + ']:')))
    #adiciona a linha na matriz turma
    turma.append(linha)
```


Exemplo

- Programa que cria uma matriz **n x m** preenchida com zeros

```
n = eval(input('Digite a dimensão n da
matriz: '))
m = eval(input('Digite a dimensão m da
matriz: '))
matriz = []
for i in range(n):
    linha = []
    for j in range(m):
        linha.append(0)
    matriz.append(linha)
print(matriz)
```

Imprimir em forma de matriz

- Programa que cria uma matriz **n x m** preenchida com zeros e a **imprime no formato de matriz**

```
n = eval(input('Digite a dimensão n da matriz: '))
m = eval(input('Digite a dimensão m da matriz: '))
matriz = []
for i in range(n):
    matriz.append([0]*m)
#imprimir em formato de matriz
for i in range(n):
    print(matriz[i])
```

Exemplo Contar Pares

- Programa que lê uma matriz 3x3 digitada pelo usuário e conta quantos números pares existem na matriz, imprimindo na tela o resultado e a matriz.

Exemplo Contar Pares

```
matriz = []
for i in range(3):
    linha = []
    for j in range(3):
        linha.append(eval(input('Digite o valor de [' + str(i) + ', ' +
                                str(j) + ']:')))
    matriz.append(linha)
#contar pares
pares = 0
for i in range(3):
    for j in range(3):
        if matriz[i][j] % 2 == 0:
            pares = pares + 1
#imprimir em formato de matriz
for i in range(3):
    print(matriz[i])
#imprimir qtde de números pares
print('A matriz contém', pares, 'números pares')
```

Variação Exemplo Contar Pares

```
matriz = []
for i in range(3):
    linha = []
    for j in range(3):
        linha.append(eval(input('Digite o valor de [' + str(i) + ', ' +
                                str(j) + ']:')))
    matriz.append(linha)
#contar pares
pares = 0
for linha in matriz:
    for valor in linha:
        if valor % 2 == 0:
            pares = pares + 1
#imprimir em formato de matriz
for i in range(3):
    print(matriz[i])
#imprimir qtde de números pares
print('A matriz contém', pares, 'números pares')
```

Python permite misturar tipos em uma matriz

- Exemplo: programa que armazena os nomes e idades de 10 pessoas em uma matriz, e imprime o nome da pessoa mais nova

Encontra a pessoa mais nova

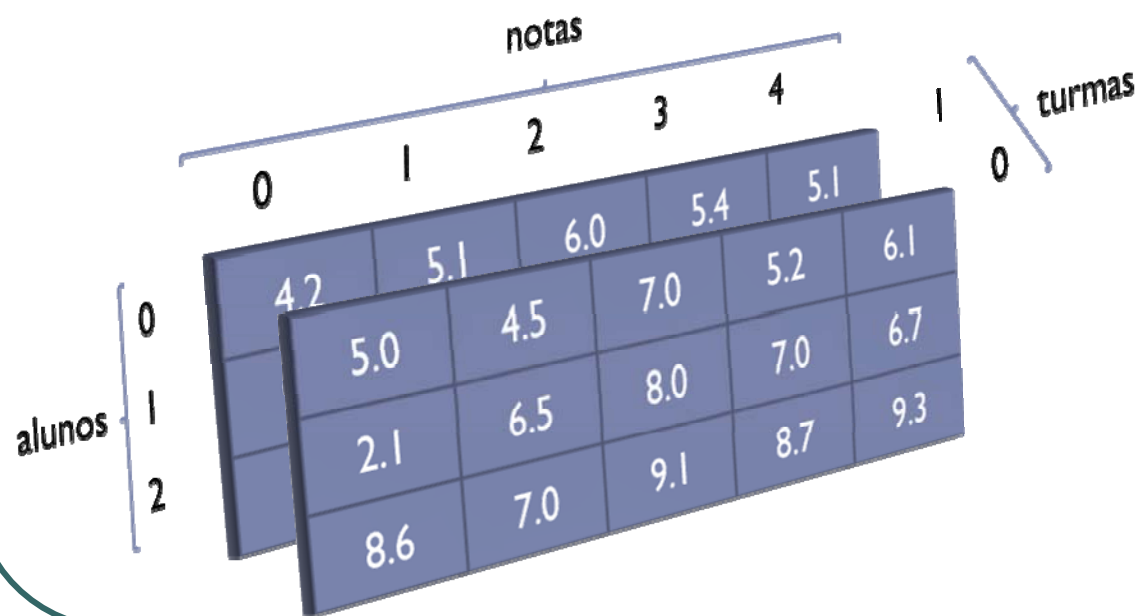
```
m = []
#preenche a matriz
for i in range(10):
    linha = []
    linha.append(input('Digite o nome da pessoa ' + str(i) + ':'))
    linha.append(eval(input('Digite a idade de ' + linha[0] + ':')))
    m.append(linha)
#procura a pessoa mais nova
menor = m[0][1]
pos = 0
for i in range(10):
    if m[i][1] < menor:
        menor = m[i][1]
        pos = i
#imprime a matriz
for i in range(10):
    print(m[i])
print('A pessoa mais nova é', m[pos][0])
```

Matrizes

- Uma matriz pode ter um número qualquer de dimensões! Basta usar um índice para cada dimensão.

Exemplo motivacional

- Ainda, assumindo que **um curso tem duas turmas**, seria necessária uma matriz tridimensional para guardar as notas de todos os alunos de todas as turmas do curso.



Atribuição

```
>>> m = [[5.0, 4.5, 7.0, 5.2,  
5.1], [2.1, 6.5, 8.0, 7.0, 6.7], [8.6, 7.0, 9.1, 8.7,  
, 9.3]], [[4.2, 5.1, 6.0, 5.4, 5.1], [9.0, 8.0, 7.5,  
8.1, 8.8], [2.3, 4.4, 6.7, 6.6, 7.0]]]
```

Aluno

notas					turmas
0	1	2	3	4	
0	4.2	5.1	6.0	5.4	5.1
1	5.0	4.5	7.0	5.2	6.1
2	2.1	6.5	8.0	7.0	6.7
	8.6	7.0	9.1	8.7	9.3

Atribuição

```
>>> m = [[5.0, 4.5, 7.0, 5.2, 5.1], [2.1, 6.5, 8.0, 7.0, 6.7], [8.6, 7.0, 9.1, 8.7, 9.3]], [[4.2, 5.1, 6.0, 5.4, 5.1], [9.0, 8.0, 7.5, 8.1, 8.8], [2.3, 4.4, 6.7, 6.6, 7.0]]]
```

Turma

		notas						
		0	1	2	3	4	1	turmas
alunos	0	4.2	5.1	6.0	5.4	5.1		
	1	5.0	4.5	7.0	5.2	6.1		
	2	2.1	6.5	8.0	7.0	6.7		
		8.6	7.0	9.1	8.7	9.3		

Acesso a elemento

```
>>> m = [[5.0, 4.5, 7.0, 5.2, 5.1], [2.1, 6.5, 8.0, 7.0, 6.7], [8.6, 7.0, 9.1, 8.7, 9.3]]  
, [[4.2, 5.1, 6.0, 5.4, 5.1], [9.0, 8.0, 7.5, 8.1, 8.8], [2.3, 4.4, 6.7, 6.6, 7.0]]]  
>>> print(m[0][1][0])
```

2.1

Turma
Aluno
Nota

notas					turmas
0	1	2	3	4	
4.2	5.1	6.0	5.4	5.1	0
5.0	4.5	7.0	5.2	6.1	
2.1	6.5	8.0	7.0	6.7	
8.6	7.0	9.1	8.7	9.3	1
					2

Referências

- Slides baseados no curso de Programação de Computadores I da Prof. Vanessa Braganholo