

ESDC5: Estruturas de Dados II

Aula 05: Algoritmos de Ordenações Elementares em Java - *Selection Sort, Insertion Sort e Shell Sort*

Lista de Exercício – Listex 04

Breno Lisi Romano

<http://sites.google.com/site/blromano>

Instituto Federal de São Paulo – IFSP São João da Boa Vista
Bacharelado em Ciência da Computação – 3º Semestre



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
SÃO PAULO
Campus São João da Boa Vista



Trabalhos para Casa (1)

- Exercício 01 – Com base na proposta apresentada em aula para ordenação de objetos comparáveis, construa um projeto em Java para aplicar o algoritmo de ordenação *Selection Sort()*, permitindo ordenar quaisquer tipos de objetos.
 - Ilustre a ordenação de pelo menos dois tipos de objetos. Exemplo: Inteiros e Strings

		a[]										
i	min	0	1	2	3	4	5	6	7	8	9	10
		S	O	R	T	E	X	A	M	P	L	E
0	6	S	O	R	T	E	X	A	M	P	L	E
1	4	A	O	R	T	E	X	S	M	P	L	E
2	10	A	E	R	T	O	X	S	M	P	L	E
3	9	A	E	E	T	O	X	S	M	P	L	R
4	7	A	E	E	L	O	X	S	M	P	T	R
5	7	A	E	E	L	M	X	S	O	P	T	R
6	8	A	E	E	L	M	O	S	X	P	T	R
7	10	A	E	E	L	M	O	P	X	S	T	R
8	8	A	E	E	L	M	O	P	R	S	T	X
9	9	A	E	E	L	M	O	P	R	S	T	X
10	10	A	E	E	L	M	O	P	R	S	T	X
		A	E	E	L	M	O	P	R	S	T	X

Entradas em preto são examinadas para se encontrar a menor chave

Entradas em vermelho são o a[min]

Entradas em cinza estão na posição final

Dica para todos os exercícios: Para imprimir o array

```
private static void show(Comparable[] a) {  
    for (int i = 0; i < a.length; i++) {  
        StdOut.println(a[i]);  
    }  
}
```



Trabalhos para Casa (2)

- Exercício 02 – Com base na proposta apresentada em aula para ordenação de objetos comparáveis, construa um projeto em Java para aplicar o algoritmo de ordenação *Insertion Sort()*, permitindo ordenar quaisquer tipos de objetos.
 - Ilustre a ordenação de pelo menos dois tipos de objetos. Exemplo: Inteiros e Strings

		a[]											
i	j	0	1	2	3	4	5	6	7	8	9	10	
		S	O	R	T	E	X	A	M	P	L	E	Entradas em cinza não se movem
1	0	O	S	R	T	E	X	A	M	P	L	E	
2	1	O	R	S	T	E	X	A	M	P	L	E	
3	3	O	R	S	T	E	X	A	M	P	L	E	Entradas em vermelho está em [j]
4	0	E	O	R	S	T	X	A	M	P	L	E	
5	5	E	O	R	S	T	X	A	M	P	L	E	
6	0	A	E	O	R	S	T	X	M	P	L	E	
7	2	A	E	M	O	R	S	T	X	P	L	E	Entradas em preto se movem uma posição para a direita por inserção
8	4	A	E	M	O	P	R	S	T	X	L	E	
9	2	A	E	L	M	O	P	R	S	T	X	E	
10	2	A	E	E	L	M	O	P	R	S	T	X	



Trabalhos para Casa (3)

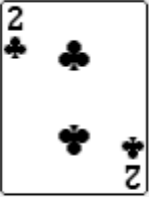
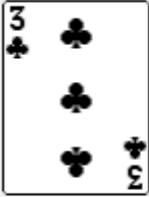
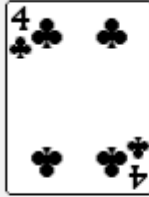
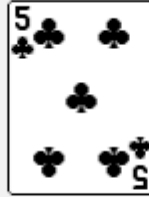
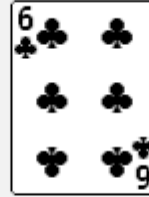
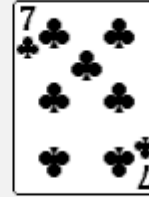
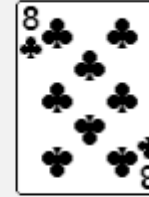


- **Exercício 03** – Com base na proposta apresentada em aula para ordenação de objetos comparáveis, construa um projeto em Java para aplicar o algoritmo de ordenação *ShellSort()*, permitindo ordenar quaisquer tipos de objetos.
 - Ilustre a ordenação de pelo menos dois tipos de objetos. Exemplo: Inteiros e Strings

input	S	H	E	L	L	S	O	R	T	E	X	A	M	P	L	E
13-sort	P	H	E	L	L	S	O	R	T	E	X	A	M	S	L	E
4-sort	L	E	E	A	M	H	L	E	P	S	O	L	T	S	X	R
1-sort	A	E	E	E	H	L	L	L	M	O	P	R	S	S	T	X



Trabalhos para Casa (4)

- **Exercício 04 – Dado como entrada um Array com objetos ordenados de forma crescente, produzir uma saída que embaralhe (*shuffle*) destes objetos:**
 - Dicas:
 - Gere um número randômico para cada uma das entradas do Array
 - Ordene os elementos do Array com base nos números randômicos gerados para cada uma das entradas

								
0.8003	0.9706	0.9157	0.9649	0.1576	0.4854	0.1419	0.4218	0.9572