

1 - Estrutura de dados é o ramo da computação que estuda os diversos mecanismos de organização de dados para atender aos diferentes requisitos de processamento. As estruturas de dados definem a organização, métodos de acesso e opções de processamento para a informação manipulada pelo programa.

2 - Dados Primitivos: são grupos de valores indivisíveis (como os tipos básicos int, boolean, char e float, double, short do JAVA). Exemplo: uma variável do tipo boolean pode assumir o valor verdadeiro ou o valor falso, e nenhum outro valor.

Dados Abstratos: modelo matemático, acompanhado das operações definidas sobre modelo. Exemplo: o conjunto dos inteiros acompanhado das operações de adição, subtração e multiplicação.

- 7 -**
- a. Pelo menos um ponteiro para indicar a última posição da lista.
 - b. Operação de inserir, retirar, localizar e alterar.
 - c. `public void insere (Object x) throws Exception {`
`if (this . ultimo >= this .item.length) throw new Exception ("Erro : A lista`
`esta cheia");`
`else { this .item[this . ultimo] = x;`
`this . ultimo = this . ultimo + 1;`
`}`
`}`
 - d. Um nó é uma estrutura que armazena a informação a ser gerenciada por uma lista.

9 -

índice	0	1	2	3	4	5	6	7	8	
valor	2	3	6	7	8					
	prim				ult					
índice	0	1	2	3	4	5	6	7	8	
valor	1	2	3	6	7	8				Inser(1);
	prim					ult				
índice	0	1	2	3	4	5	6	7	8	
valor	1	2	3	6	7	8	15			Inser(15);
	prim						ult			
índice	0	1	2	3	4	5	6	7	8	
valor	1	2	6	7	8	15				Remove(3);
	prim					ult				
índice	0	1	2	3	4	5	6	7	8	
valor	1	2	4	6	7	8	15			Inser(4);
	prim						ult			

índice	0	1	2	3	4	5	6	7	8	
valor	1	2	4	6	8	15				Remove(7);
	prim					ult				
índice	0	1	2	3	4	5	6	7	8	
valor										LimpaLista();
	prim	ult								
índice	0	1	2	3	4	5	6	7	8	
valor	10									Insere(10);
	prim	ult								
índice	0	1	2	3	4	5	6	7	8	
valor	9	10								Insere(9);
	prim	ult								
índice	0	1	2	3	4	5	6	7	8	
valor	5	9	10							Insere(5);
	prim		ult							

- 10 - a. $A + B + C \Rightarrow AB+C-$
b. $(A+B)*(C-D) / (E*F) \Rightarrow AB+CD-*EF*/$
c. $\{(A-B)/(C*D)\}/E \Rightarrow AB-CD**E/$
d. $(A+ \{[(B-C)*(D-E)+F]/G\})/(H-J) \Rightarrow (A+\{[BC-DE-*F+]/G\})/(HJ-) \Rightarrow$
 $(A+\{BC-DE-*F+G\}/(HJ-) \Rightarrow BC-DE-*F+G/A+/(HJ-) \Rightarrow BC-DE-*F+G/A+HJ-/$

11. a.

+	-	
B	C	
A	(A+B)	A+B - C

b.

		-	+		
-	+	E	F	/	
B	C	D	(D - E)	(D - E+F)	
A	(A - B)	(A - B+C)	(A - B+C)	(A - B+C)	(A - B+C)/(D - E+F)

c.

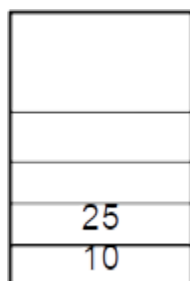
		/		
*	/	E	+	
B	C	D	(D/E)	
A	(A*B)	(A*B)/C	(A*B)/C	{(A*B)/C}+(D/E)

d.

	*		
/	D	+	
B	C	(C*D)	
A	(A/B)	(A/B)	(A/B)+(C*D)

12 - a. Para a manipulação de uma pilha, é necessário apenas um ponteiro para indicar o topo da pilha, pois tanto a inserção e a remoção acontecem no topo da pilha.

b. Tanto a remoção e a inserção da pilha acontece no topo, porém quando você insere um elemento o ponteiro é incrementado, e quando acontece a remoção o ponteiro é decrementado.



```
Desempilha();
Empilha (10);
Empilha(25);
Empilha(81);
```

```
Empilha => topo ++;
Desempilha => topo --;
```

```
c. public void ImprimirPilha(){
    for(int i = 0; i <= topo; i++){
        System.out.println("\nImprimindo Pilha.....\n");
        System.out.println("Posição [" + i + "] => Nome: " + pilha[i].getNome());
    }
}
```

13 - a. Para a manipulação de uma fila, são necessários dois ponteiros, um para indicar o início e outro para indicar o fim da fila.

b.

0	1	2	3	4	5	6	7	8	
12									Inser(12);
In		Fim							
0	1	2	3	4	5	6	7	8	
12	3								Inser(3);
In	Fim								
0	1	2	3	4	5	6	7	8	
12	3	25							Inser(25);
In		Fim							
0	1	2	3	4	5	6	7	8	
12	3								Remove(25);
In	Fim								
0	1	2	3	4	5	6	7	8	
12	3	8							Inser(8);
In		Fim							

c.

```
public void ImprimirFila(){
    for(int i = 0; i <= fim; i++){
        System.out.println("\nImprimindo Fila.....\n");
        System.out.println("Posição [" + i + "] => Nome: " + fila[i].getNome());
    }
}
```
