

PILHA:

```
import java.util.ArrayList;
import java.util.Vector;
```

```
public class Pilha implements Incrementavel{ //Implementa a interface Acessivel com metodos
comuns a pilha, fila e lista.
```

```
    private Vector<Object> vect;
    private Object elemento;
    private int topo;
    private ArrayList<String> listaStr;
```

```
    public Pilha(){
        this.vect = new Vector<Object>();
        this.topo = -1;
    }
```

```
//Insere elemento na
```

```
pilha+++++++++
```

```
    @Override
    public void inserirElemento(Object ele){
        if(isEmpty()){
            vect = new Vector<Object>();
            vect.add(0, ele);
            topo++;
        }
        else{
            vect.add(ele);
            topo++;
        }
    }
```

```
//Remove e retorna elemento
```

```
removido+++++++++
```

```
    public Object removerElemento(){
        elemento = null;
        if(isEmpty()){
            return null;
        }
        else{
            elemento = vect.get(topo);
            vect.remove(topo);
            topo--;
            return elemento;//Retorna elemento removido.
        }
    }
```

```
//Esvazia pilha e mostra elementos do topo à base+++++++++
```

```
    public ArrayList<String> esvaziarMostrar(){
        elemento = null;
        listaStr = new ArrayList<String>();
        while(topo > -1){
            elemento = vect.get(topo);
            vect.remove(topo);
            topo--;
            listaStr.add("Elemento " + (topo+1) + ": " + elemento);
        }
    }
```

```
    return listaStr;//Retorna lista com os elementos removidos e seus indices.
```

```
}
```

```
//Verifica se a pilha está
```

```
vazia+++++++++
```

```
    public boolean isEmpty(){ //estáVazia()
        if(topo == -1){
            return true;
        }
        return false;
    }
```

```
}
```

```
import java.util.ArrayList;
import java.util.Vector;
```

```
private Vector<Object> vect;  
private Object elemento;  
private int inicio, fim;  
private ArrayList<String> listaStr;
```

```
public File(){
    this.vect = new Vector<Object>();
    this.inicio = -1;
    this.fim = -1;
}
```

```

fila+++++

```

```
@Override
public void inserirElemento(Object ele){
    if(isEmpty()){
        vect = new Vector<Object>[];
        this.inicio++;
        vect.add(this.inicio, ele);
        fim++;
    }
    else{
        vect.add(fim+1, ele);
        fim++;
    }
}
```

+++++

```
@Override
public Object removerElemento() {
    elemento = null;
    if(isEmpty()){
        return null;
    }
    else{
        if(fim == 0){ //Se na fila existir apenas um elemento.
            elemento = vect.get(inicio);
            vect.remove(inicio);
            fim--;
            inicio--;
        }
        else{ //Se na fila houver mais de um elemento.
            elemento = vect.get(inicio);
            vect.remove(inicio);
            fim--;
        }
        return(elemento);
    }
}
```

```
fim++++++++++++++++++++++++++++++++++++++//
```

```
@Override  
public ArrayList<String> esvaziarMostrar() {  
    elemento = null;  
    listaStr = new ArrayList<String>();  
    int index = 0;  
    while(fim > -1){ //Enquanto existir algum elemento.  
        elemento = removerElemento();  
        listaStr.add("Elemento " + (index) + ": " + elemento);  
        index++;  
    }  
  
    return listaStr; //Retorna lista com os elementos removidos e seus indices.  
}
```

```

//Verifica se a fila está
vazia+++++++//
@Override
public boolean isEmpty(){ //estáVazia()
    if(this.fim == -1){
        return true;
    }
    return false;
}
}

```

LISTA:

```

import java.util.ArrayList;
import java.util.Collections;
import java.util.Vector;

```

```

public class Lista implements Incrementavel {

```

```

    private Vector<Object> vect;
    private Object elemento;
    private int inicio, fim;
    private ArrayList<String> listaStr;

```

```

    public Lista(){
        this.vect = new Vector<Object>();
        this.inicio = -1;
        this.fim = -1;
    }

```

//Métodos de

```

inserção+++++++//
/

```

```

//Inserção simples
@Override
public void inserirElemento(Object ele) {
    if(isEmpty()){
        vect = new Vector<Object>();
        inicio++;
        vect.add(this.inicio, ele);
        fim++;
    }
    else{
        vect.add(fim+1, ele);
        fim++;
    }
}

```

```

//Inserção em índice
public void inserirElemento(Object ele, int index){
    if(isEmpty()){
        vect = new Vector<Object>();
        inicio++;
        vect.add(this.inicio, ele);
        fim++;
    }
    else{
        if(index == fim){
            vect.add(fim, ele);
            fim++;
        }
        else{
            vect.add(index, ele);
            fim++;
        }
    }
}
}

```

## //Métodos de remoção

+++++

### //Remoção simples

#### @Override

```
public Object removerElemento() {
    elemento = null;
    if(isEmpty()){
        return null;
    }
    else{
        if(fim == 0){ //Se na lista existir apenas um elemento.
            elemento = vect.get(inicio);
            vect.remove(inicio);
            fim--;
            inicio--;
        }
        else{ //Se na lista houver mais de um elemento.
            elemento = vect.get(inicio);
            vect.remove(inicio);
            fim--;
        }
        return(elemento);
    }
}
```

### //Remoção em índice

```
public Object removerElemento(int index){
    elemento = null;
    if(isEmpty()){
        return null;
    }
    else{
        if(index > fim || index <= {-1}){
            return null;
        }
        else{
            if(fim == 0){
                elemento = vect.get(inicio);
                vect.remove(inicio);
                fim--;
                inicio--;
            }
            else{
                elemento = vect.get(index);
                vect.remove(index);
                fim--;
            }
            return(elemento);
        }
    }
}
```

## //Métodos de esvaziamento e

ordenação+++++

#### @Override

```
public ArrayList<String> esvaziarMostrar() {
    elemento = null;
    listaStr = new ArrayList<String>();
    int i = 0;
    while(fim > -1){ //Enquanto existir algum elemento.
        elemento = removerElemento();
        listaStr.add("Elemento " + (i) + ": " + elemento);
        i++;
    }

    return listaStr; //Retorna lista com os elementos removidos e seus índices.
}
```

### //Ordernar alfabeticamente

```
public void ordenarPorNome(){
    listaStr = new ArrayList<String>();
    int i = fim;
    while(i > -1){ //Enquanto existir algum elemento.
        listaStr.add((String)vect.get(i));
        i--;
    }
}
```

```
Collections.sort(listaStr); // Ordena os elementos alfabeticamente.
changePositions(listaStr);
```

```
}
```

```
// Verifica se a lista está
```

```
vazia ++++++
```

```
@Override
```

```
public boolean isEmpty() {
```

```
    if (this.fim == -1) {
```

```
        return true;
```

```
    }
```

```
    return false;
```

```
}
```

```
//
```

```
private void changePositions(ArrayList<String> strList) { /* Muda a posição dos elementos do  
vetor
```

```
conforme as posições da lista de nomes  
passadas por parâmetro */
```

```
Vector<Object> aux = vect;
```

```
int j;
```

```
for (int i = 0; i < strList.size(); i++) {
```

```
    j = 0;
```

```
    System.out.println(strList.get(i));
```

```
    while (!strList.get(i).equals(aux.get(j).toString())) {
```

```
        j++;
```

```
    }
```

```
    vect.add((i), aux.get(j));
```

```
}
```

```
}
```

```
}
```

```
INCREMENTÁVEL:
```

```
import java.util.ArrayList;
```

```
public interface Incrementavel {
```

```
    public void inserirElemento(Object elemento);
```

```
    public Object removerElemento();
```

```
    public ArrayList<String> esvaziarMostrar();
```

```
    public boolean isEmpty();
```

```
}
```