

# **Unidade V:**

## **Estruturas de Dados Básicas**

### **com Alocação Flexível - Pilha**

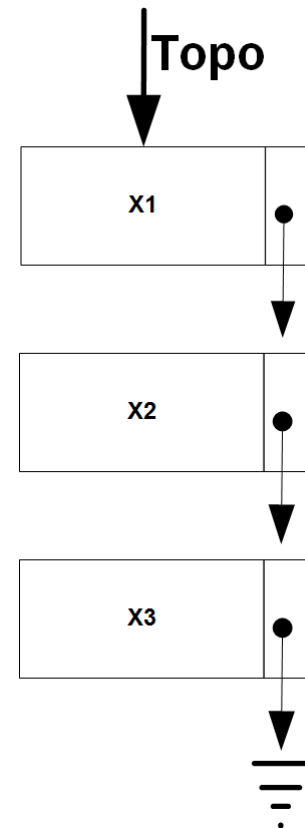
**Prof. Max do Val Machado**



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Curso de Ciência da Computação

- [PrincipalPilha.java](#), igual ao da estrutura sequencial
- [Pilha.java](#), criará instâncias como:

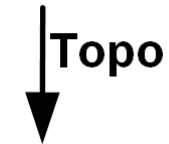


# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

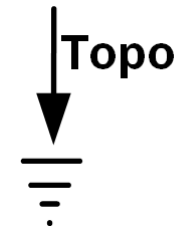
# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



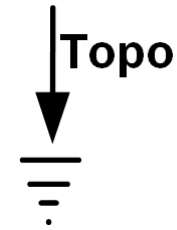
# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



# Inserir (ou Empilhar)

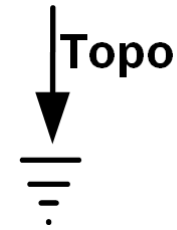
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



```
public void inserir(int x) {  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

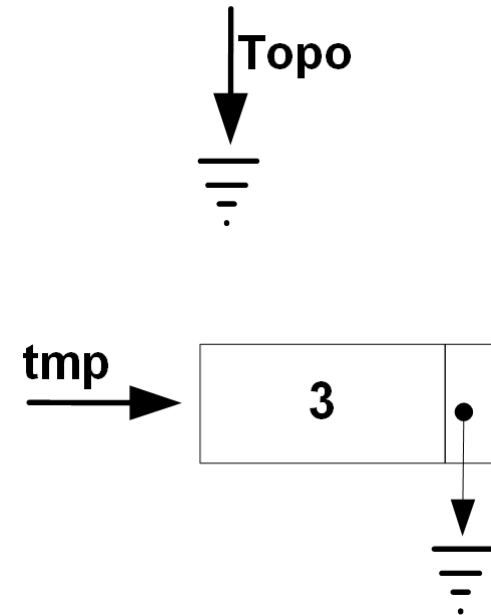


```
public void inserir(int x) { //Inserir(3)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(3)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

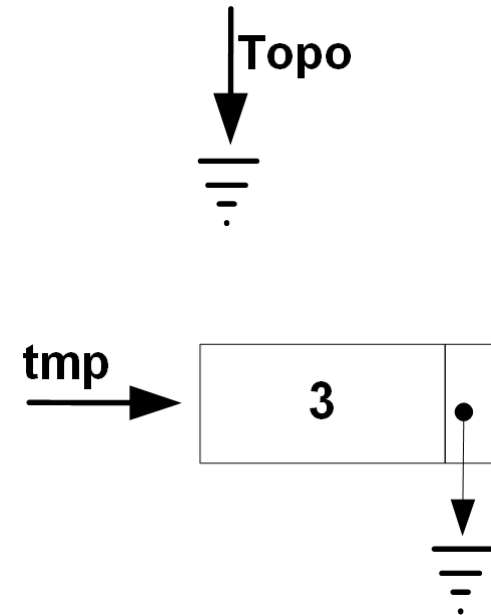




# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(3)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

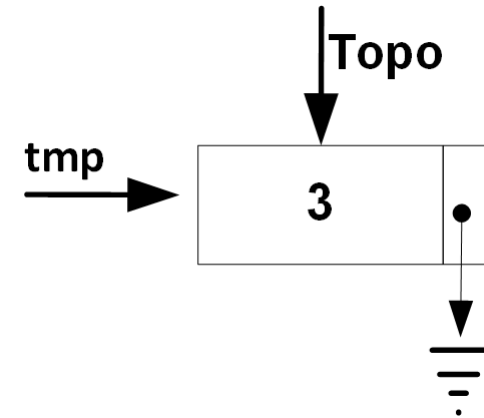


Como topo aponta para null, tmp.prox continua apontando para null

# Inserir (ou Empilhar)

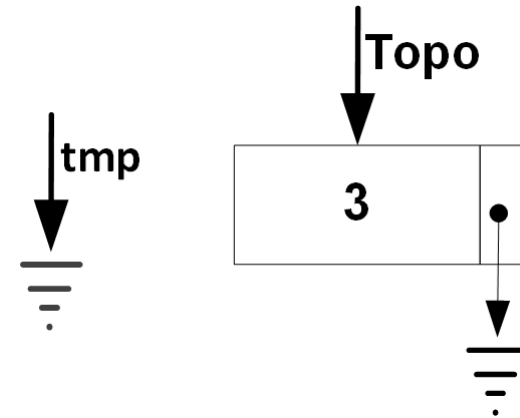
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(3)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

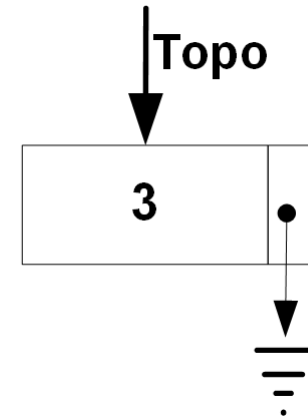
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



```
public void inserir(int x) { //Inserir(3)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

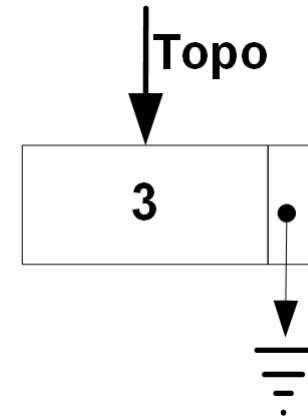
# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

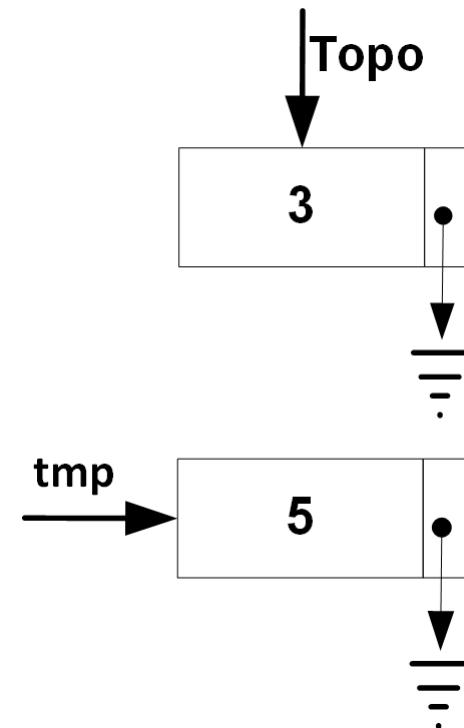


```
public void inserir(int x) { //Inserir(5)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

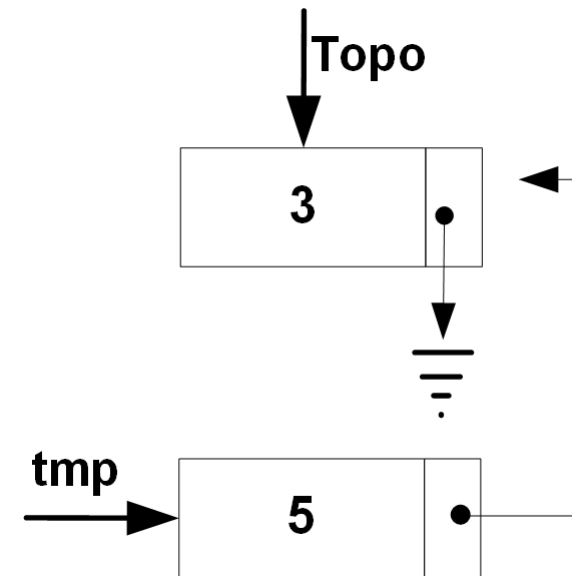
```
public void inserir(int x) { //Inserir(5)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

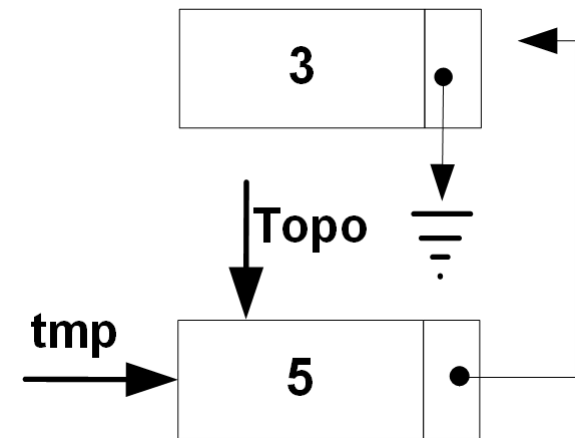
```
public void inserir(int x) { //Inserir(5)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(5)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

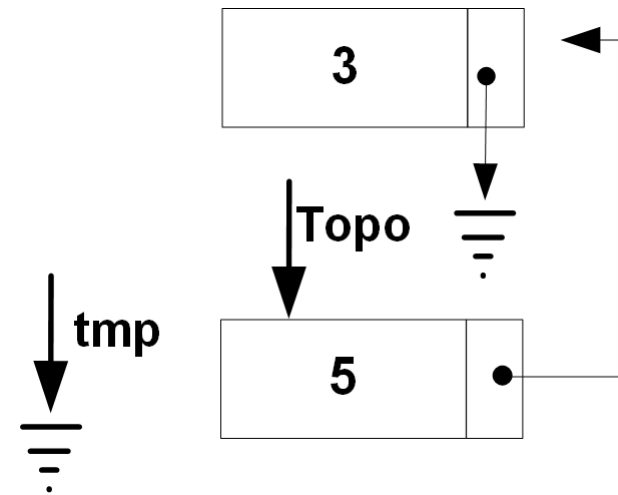




# Inserir (ou Empilhar)

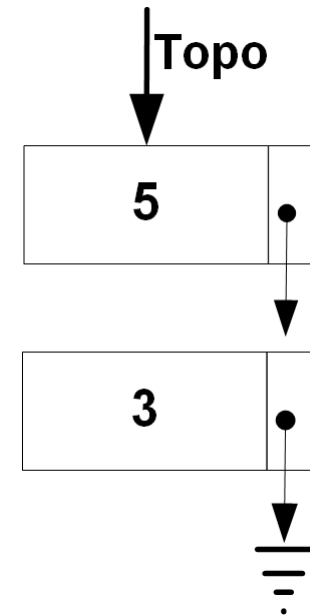
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(5)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Classe Pilha

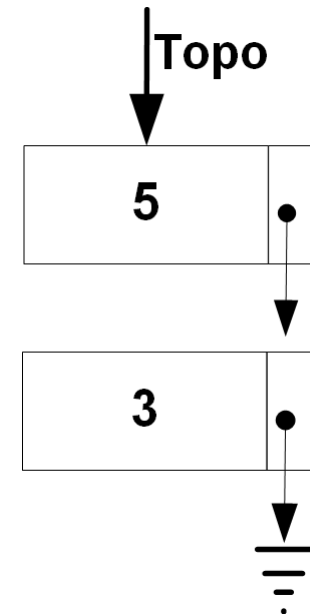
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

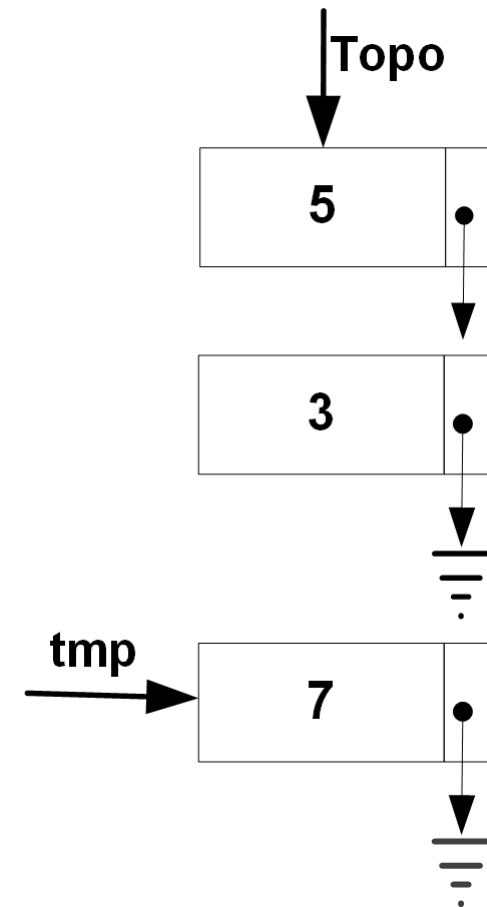
```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

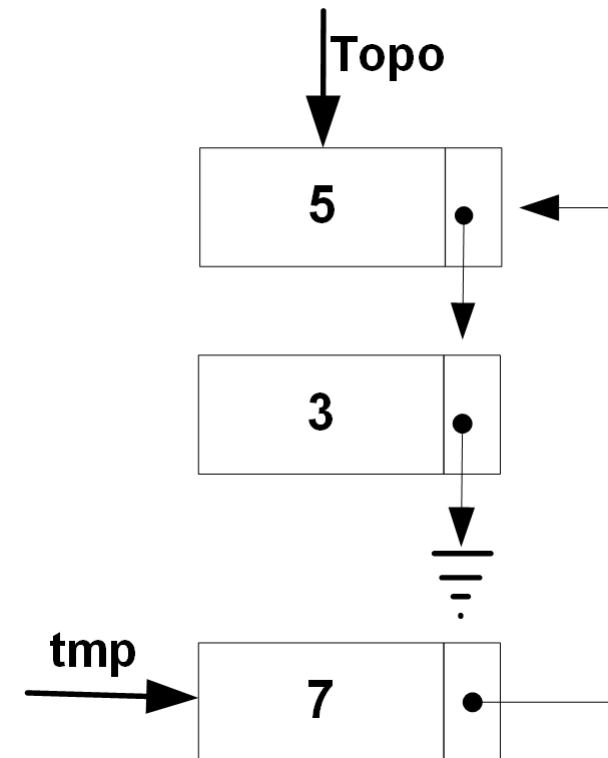
```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

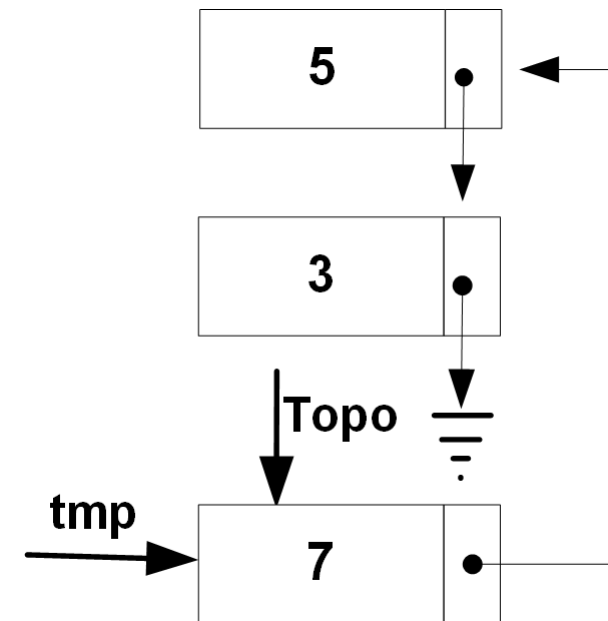
```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

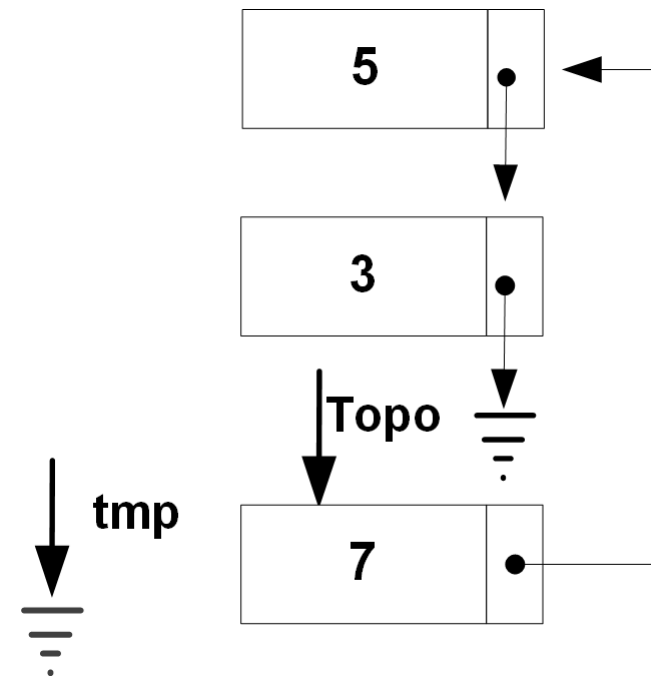
```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Inserir (ou Empilhar)

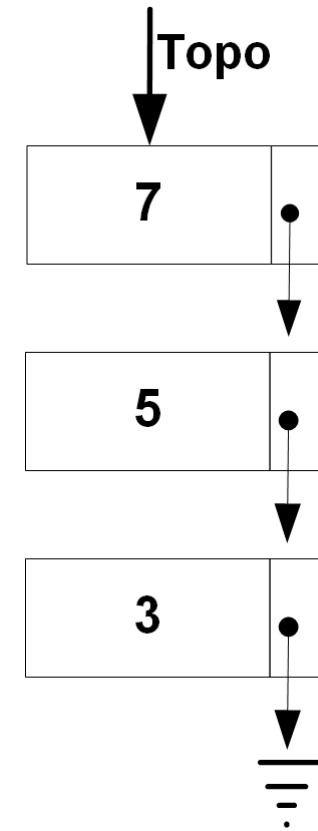
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```



# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

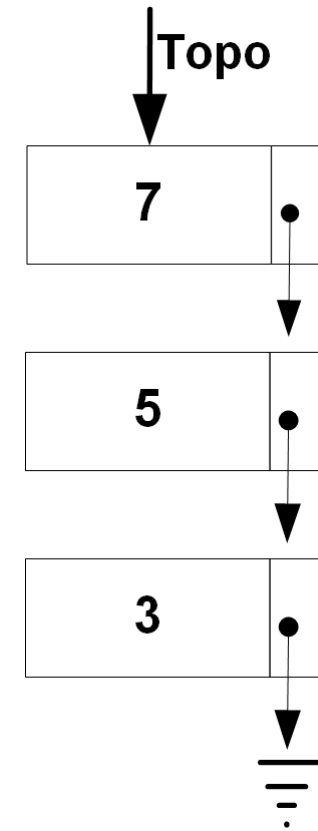




# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public int remover() throws Exception {
```

```
    if (topo == null)
```

```
        throw new Exception("Erro!");
```

```
    int elemento = topo.elemento;
```

```
    Celula tmp = topo;
```

```
    topo = topo.prox;
```

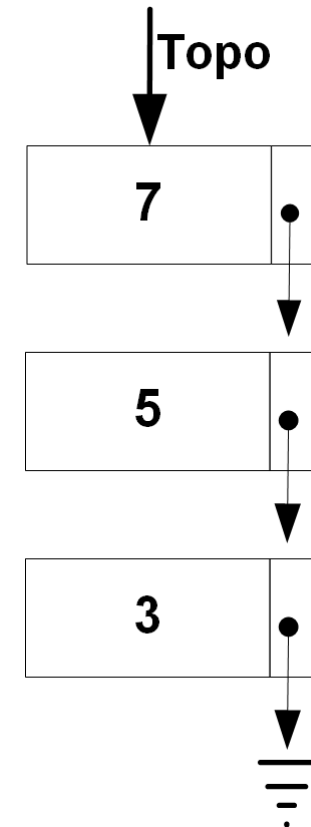
```
    tmp.prox = null;
```

```
    tmp = null;
```

```
    return elemento;
```

```
}
```

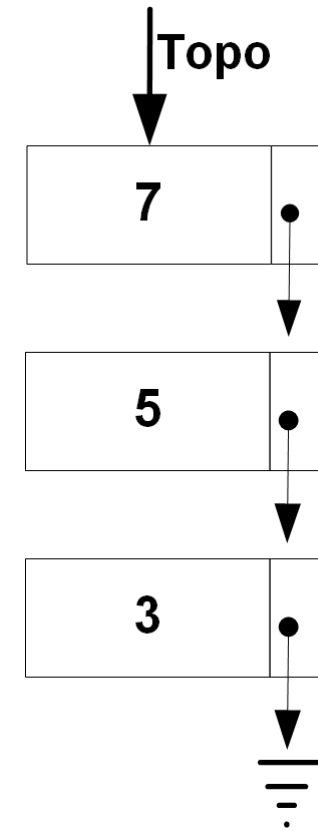
false



# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



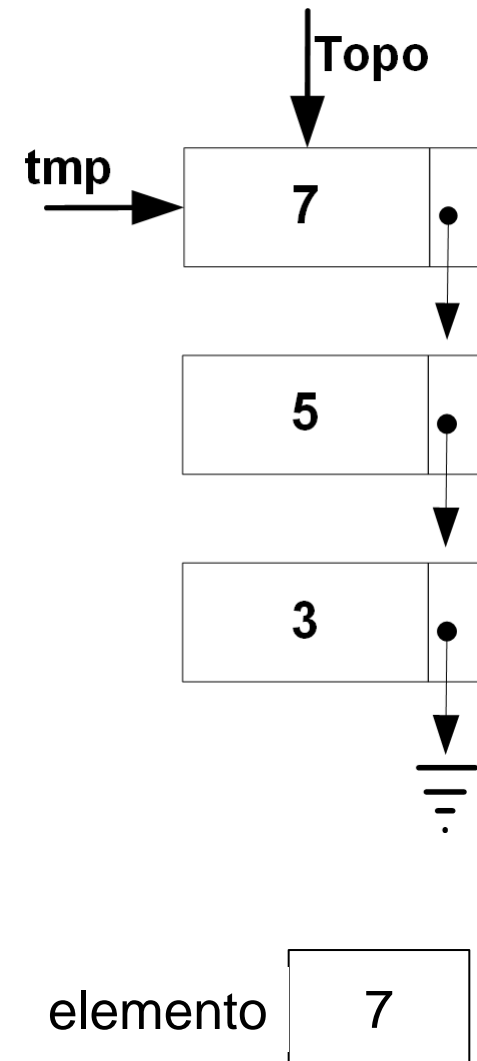
elemento 

7
---

# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

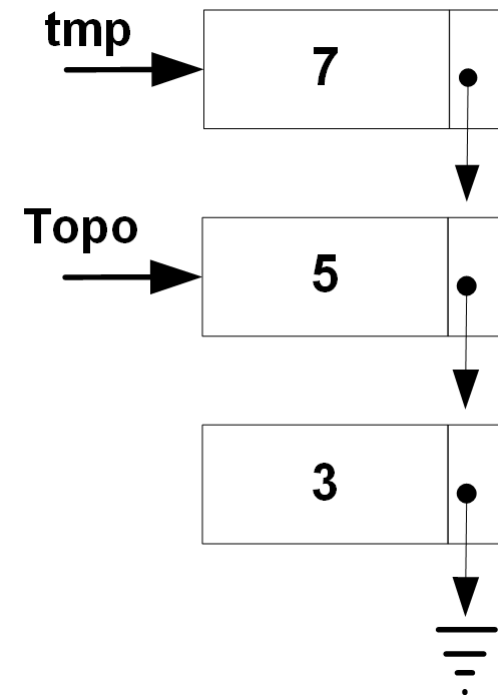
```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



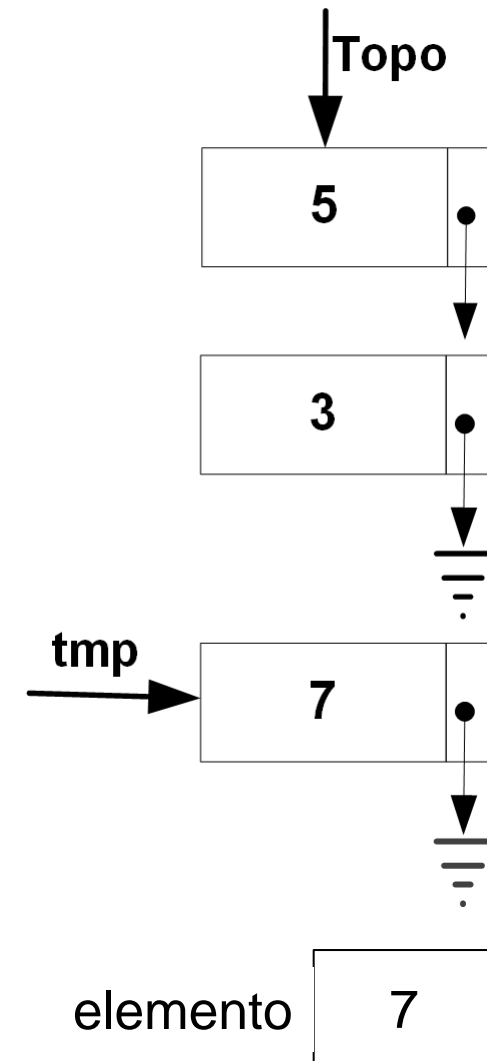
elemento 

7
---

# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

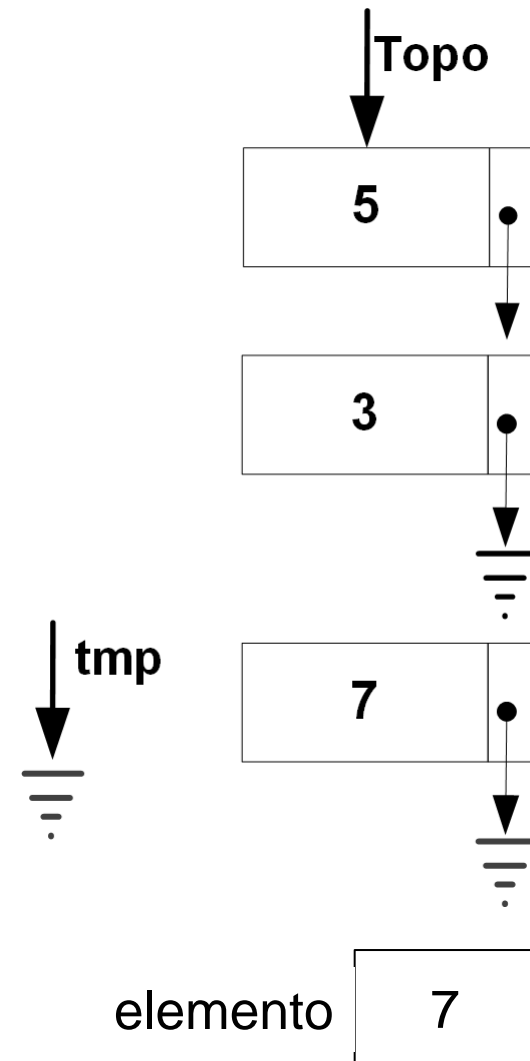
```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

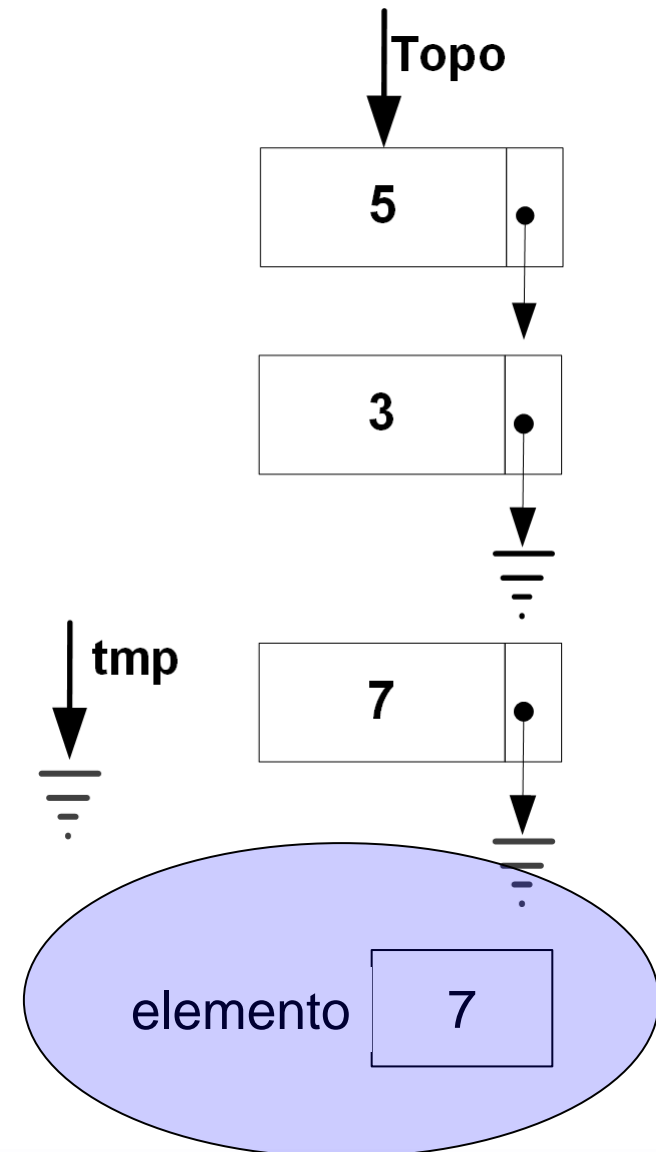
```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```



# Remover (ou Desempilhar)

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

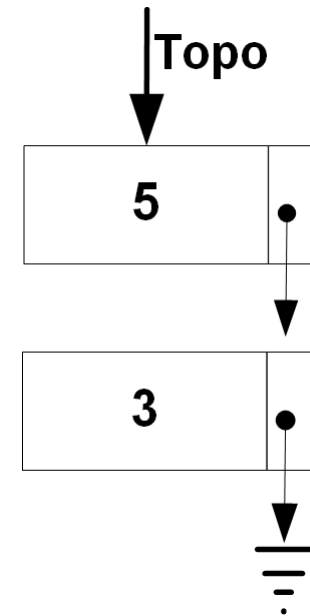
```
public int remover() throws Exception {  
    if (topo == null)  
        throw new Exception("Erro!");  
    int elemento = topo.elemento;  
    Celula tmp = topo;  
    topo = topo.prox;  
    tmp.prox = null;  
    tmp = null;  
    return elemento;  
}
```





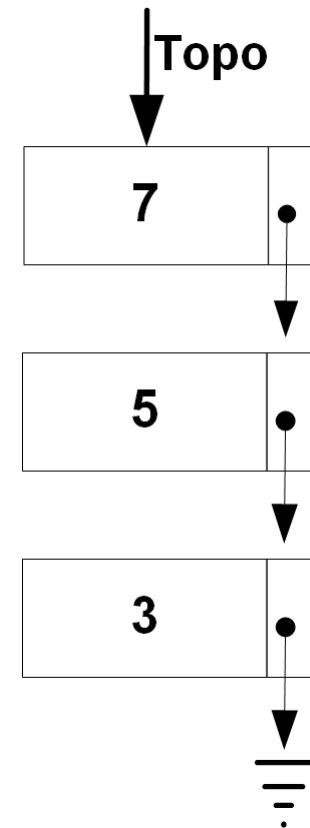
# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



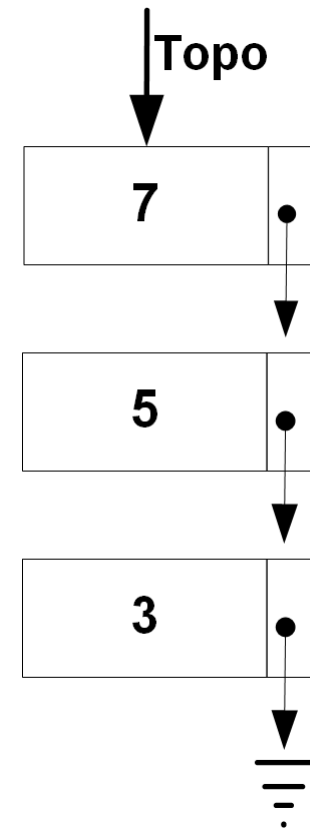
# Classe Pilha

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

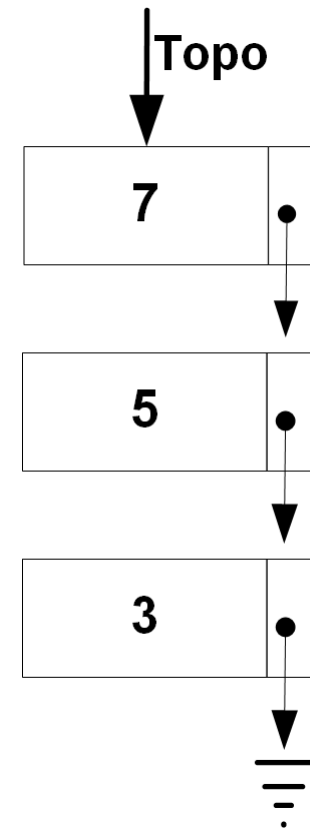
```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



Saída  
na tela

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

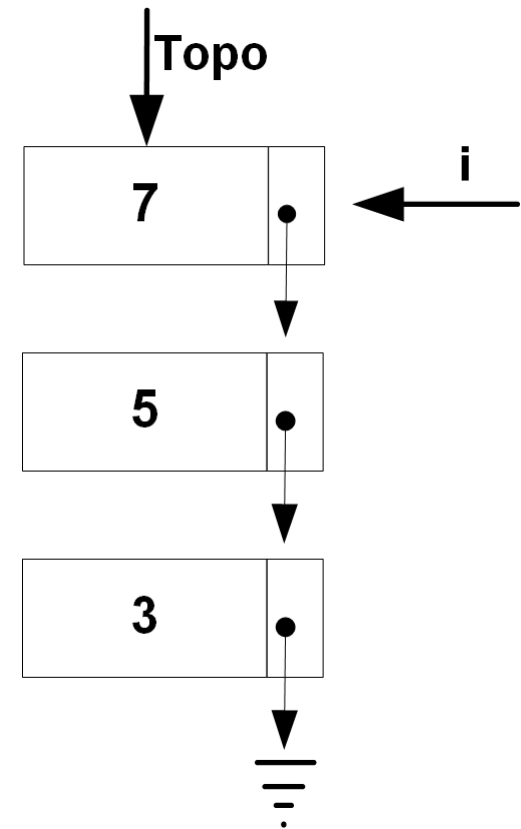


Saída  
na tela

[

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



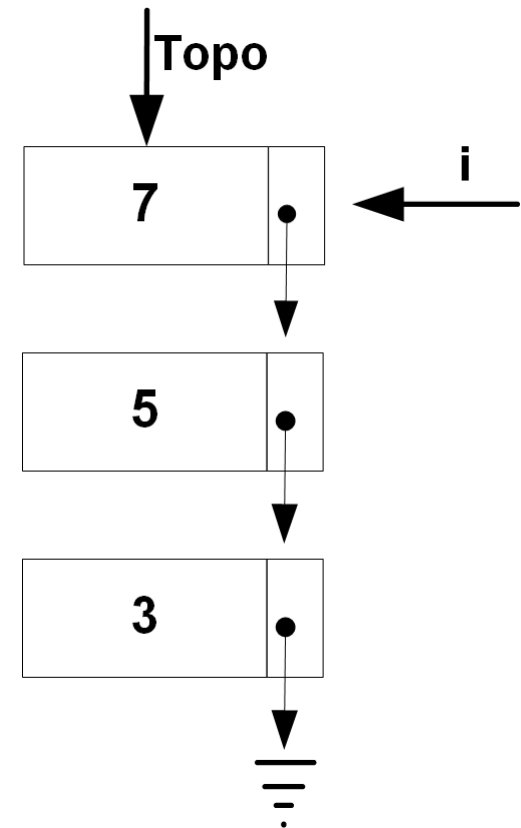
Saída  
na tela

[

# Mostrar

```
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}
```

```
public void mostrar() {
    System.out.print("[ ");
    for (Celula i = topo; i != null; i = i.prox){
        System.out.print(i.elemento + " ");
    }
    System.out.println("]");
}
```

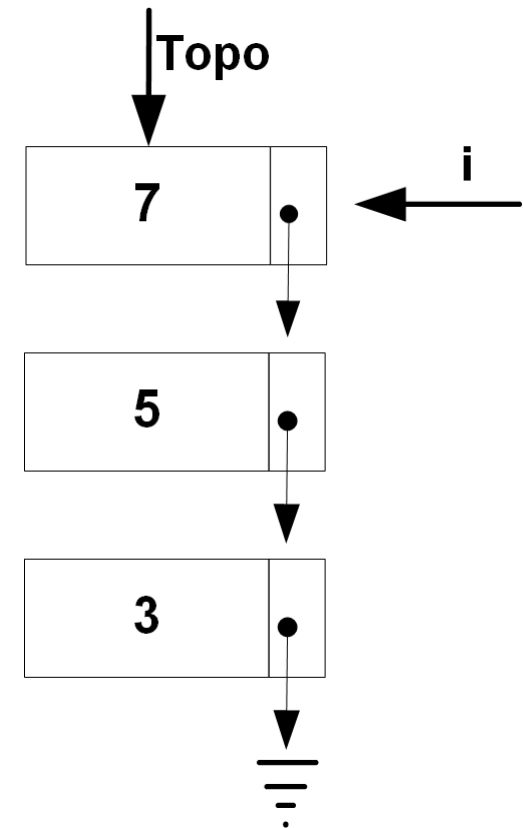


Saída  
na tela

[

```
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}
```

```
public void mostrar() {
    System.out.print("[ ");
    for (Celula i = topo; i != null; i = i.prox){
        System.out.print(i.elemento + " ");
    }
    System.out.println("]");
}
```

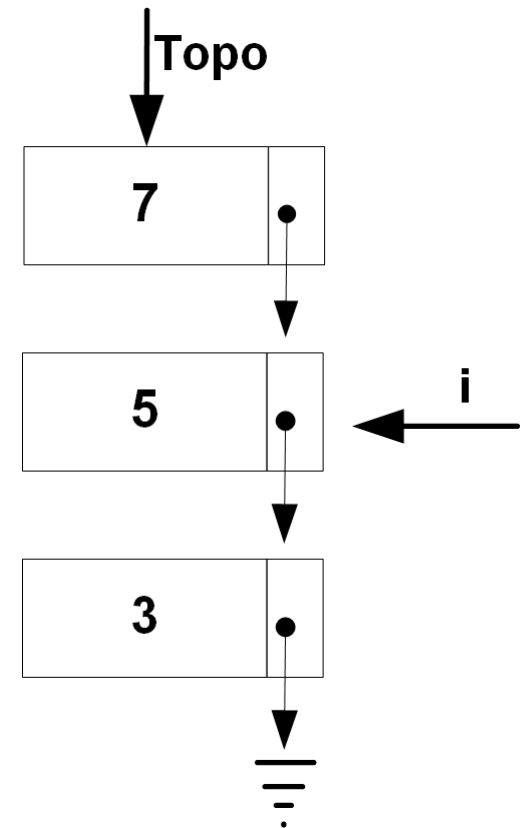


Saída  
na tela

[ 7

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



Saída  
na tela

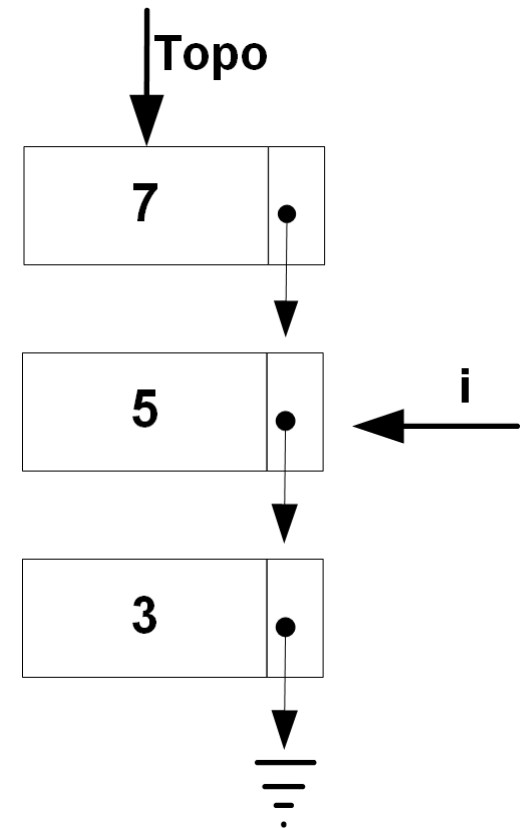
[ 7



# Mostrar

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

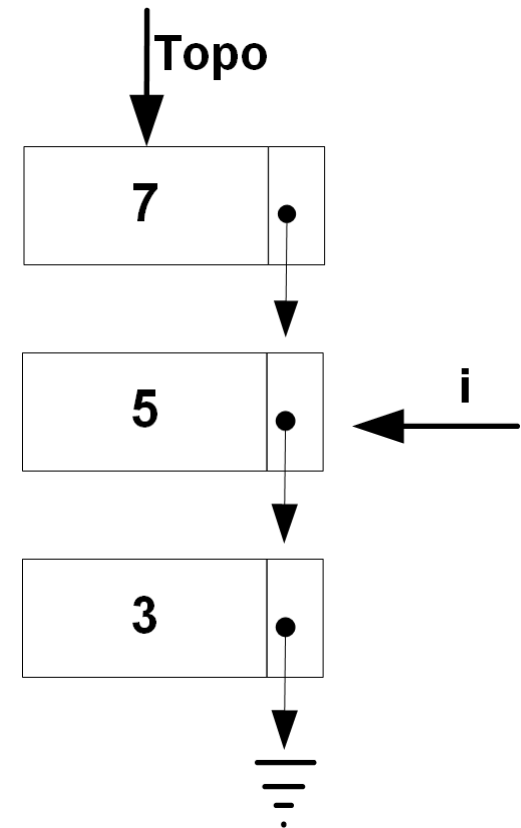


Saída  
na tela

[ 7

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

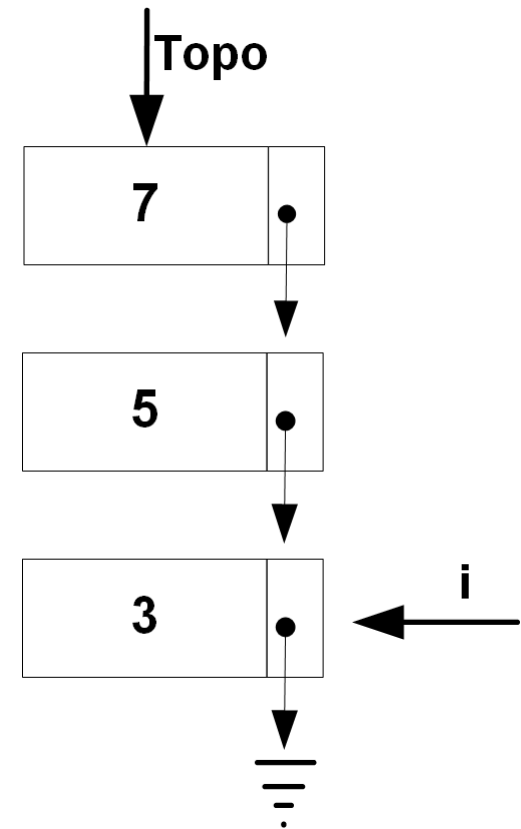


Saída  
na tela

[ 7 5

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

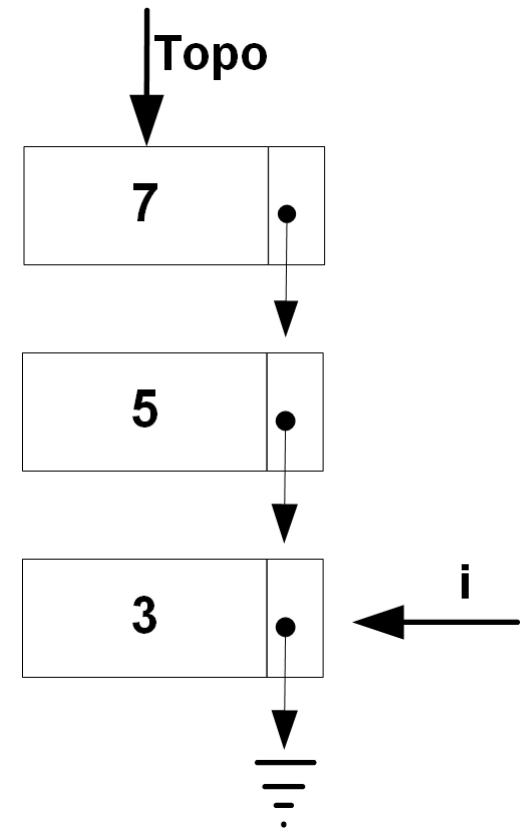


Saída  
na tela

[ 7 5

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

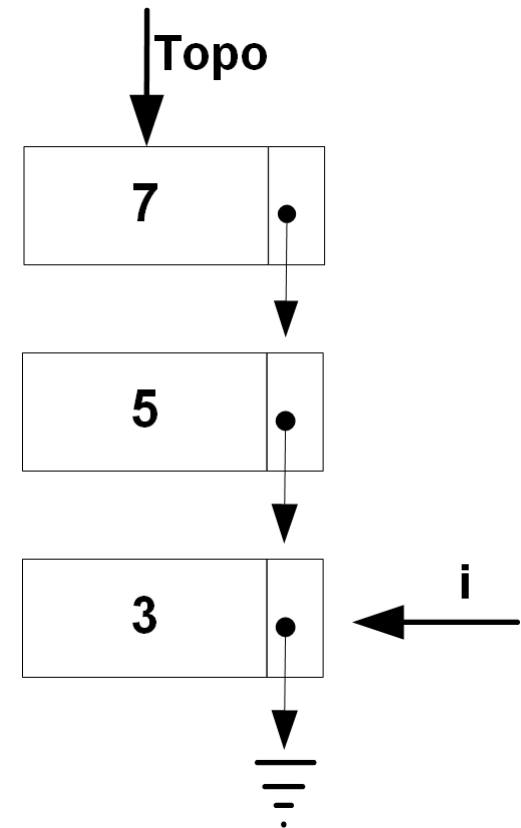


Saída  
na tela

[ 7 5

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



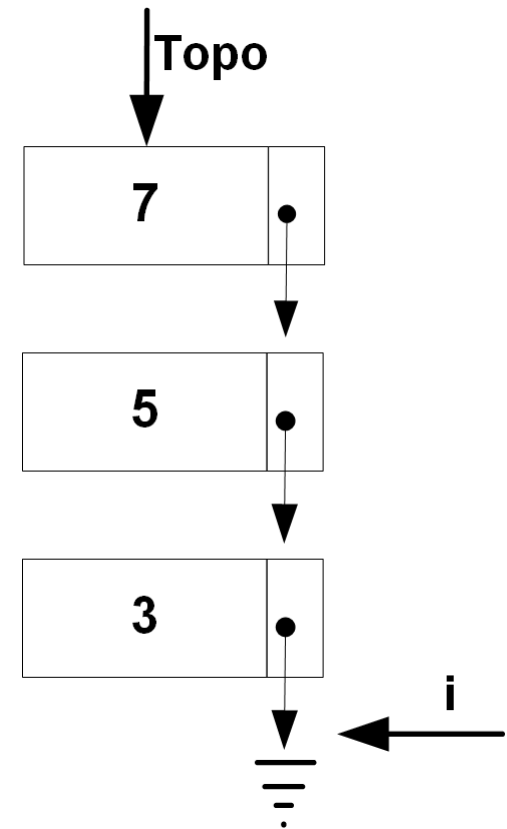
Saída  
na tela

[ 7 5 3

# Mostrar

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



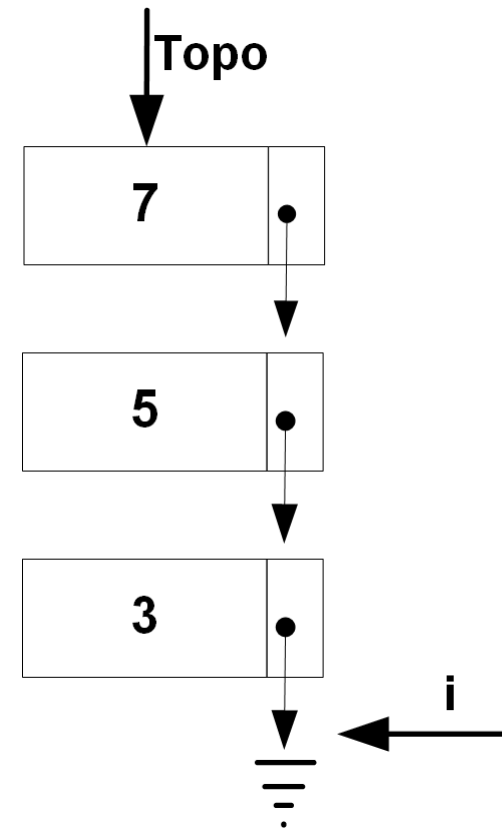
Saída  
na tela

[ 7 5 3

# Mostrar

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```

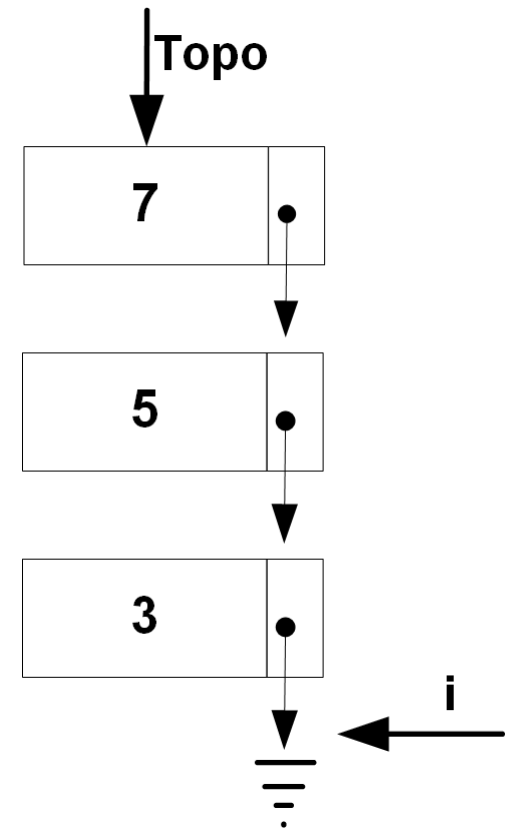


Saída  
na tela

[ 7 5 3

```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```

```
public void mostrar() {  
    System.out.print("[ ");  
    for (Celula i = topo; i != null; i = i.prox){  
        System.out.print(i.elemento + " ");  
    }  
    System.out.println("]");  
}
```



Saída  
na tela

[ 7 5 3 ]



# Exercício

- Seja nossa Pilha, faça um método que soma o conteúdo dos elementos contidos na mesma

# Exercício

- Seja nossa Pilha, faça um método que soma o conteúdo dos elementos contidos na mesma

```
int somar() {  
    int resp = 0;  
    for (Celula i = topo; i != null; i = i.prox) {  
        resp += i.elemento;  
    }  
    return resp;  
}
```

# Exercício

- Seja nossa Pilha, faça um método RECURSIVO que soma o conteúdo dos elementos contidos na mesma

# Exercício

- Seja nossa Pilha, faça um método que retorna o maior elemento contido na pilha

# Exercício

- Seja nossa Pilha, faça um método RECURSIVO que retorna o maior elemento contido na pilha

# Exercício

- Seja nossa Pilha, faça um método RECURSIVO para mostrar os elementos da pilha na ordem em que os mesmos serão removidos

# Exercício

- Seja nossa Pilha, faça um método RECURSIVO para mostrar os elementos da pilha na ordem em que os mesmos foram inseridos

# Exercício

- Seja nossa Pilha, faça um método ITERATIVO para mostrar os elementos da pilha na ordem em que os mesmos foram inseridos