

# **Unidade IV:**

# **Tipos Abstratos de Dados Flexíveis - Pilha**

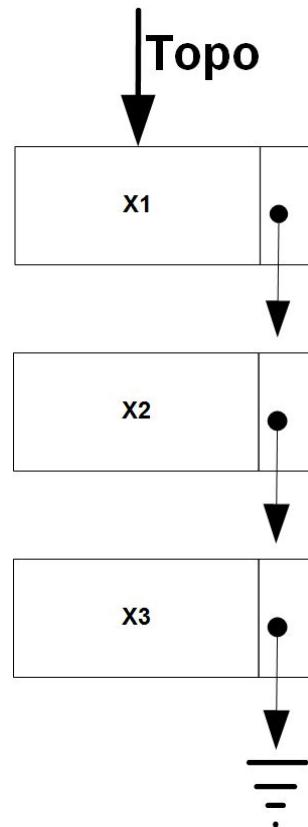


**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Código Fonte

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

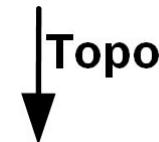


# Classe Pilha

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

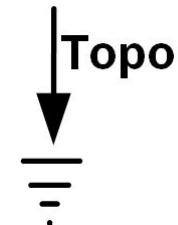
# Classe Pilha

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



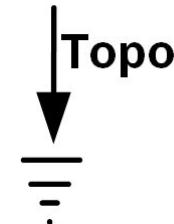
# Classe Pilha

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



# Inserir (ou Empilhar ou push)

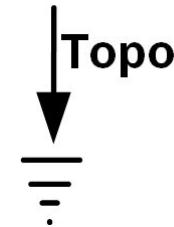
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



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

# Inserir (ou Empilhar ou push)

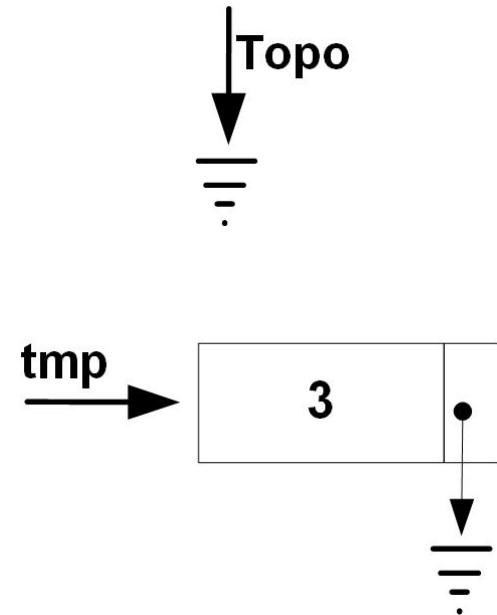
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



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

# Inserir (ou Empilhar ou push)

```
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;
```

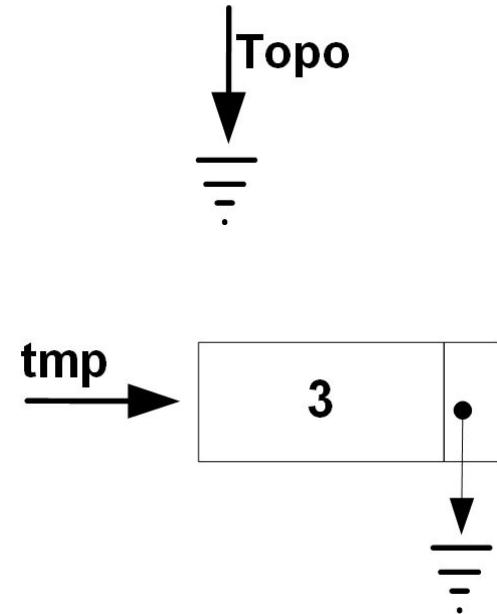
```
topo = tmp;
```

```
tmp = null;
```

```
}
```

# Inserir (ou Empilhar ou push)

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

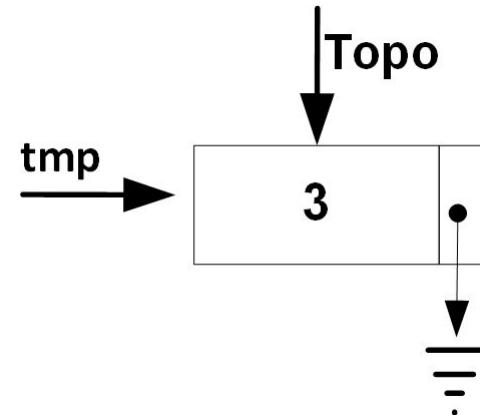


```
public void inserir(int x) { //Inserir(3)
    Celula tmp = new Celula(x);
    tmp.prox = topo;
    topo = tmp;
    tmp = null;
}
```

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

# Inserir (ou Empilhar ou push)

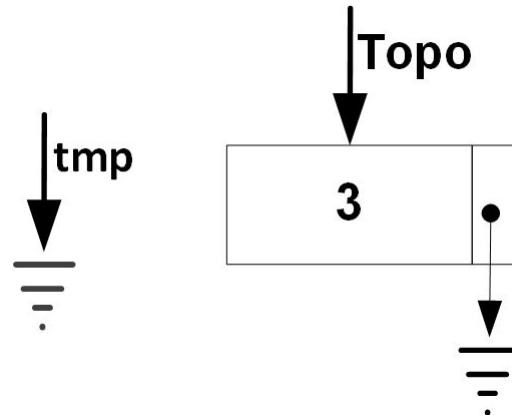
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



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

# Inserir (ou Empilhar ou push)

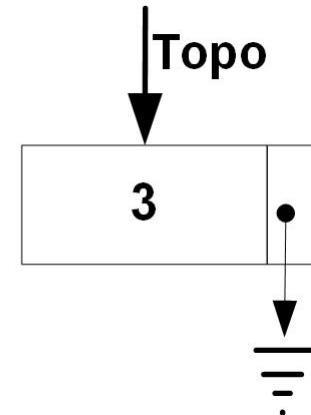
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



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

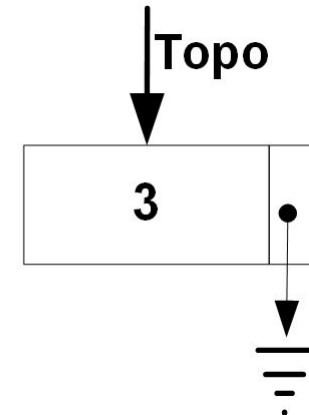
# Classe Pilha

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



# Inserir (ou Empilhar ou push)

```
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 ou push)

```
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)
```

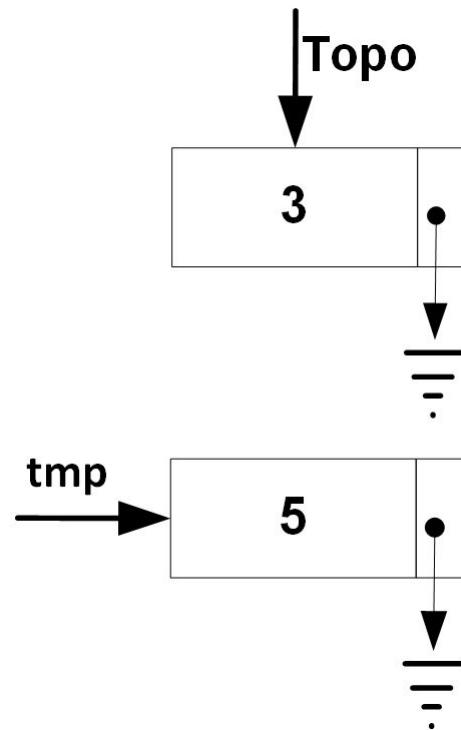
```
    Celula tmp = new Celula(x);
```

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

```
    topo = tmp;
```

```
    tmp = null;
```

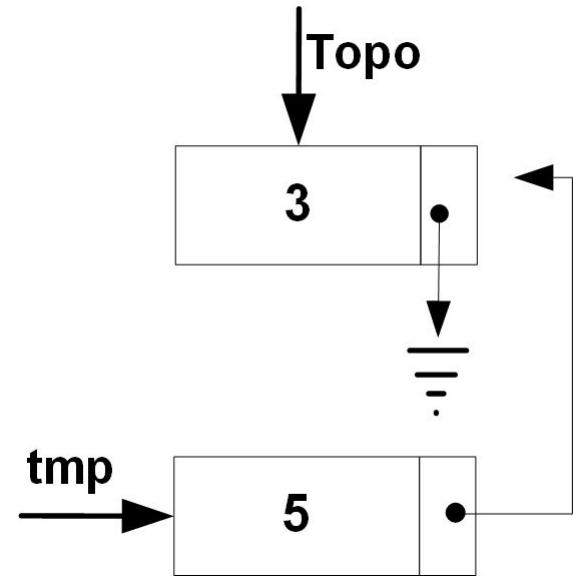
```
}
```



# Inserir (ou Empilhar ou push)

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

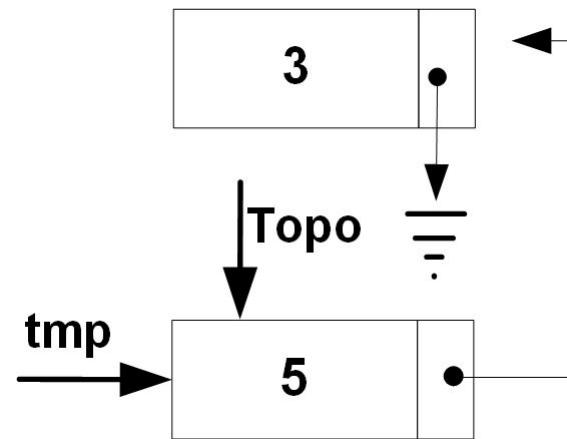
```
public void inserir(int x) { //Inserir(5)
    Celula tmp = new Celula(x);
    tmp.prox = topo;
    topo = tmp;
    tmp = null;
}
```



# Inserir (ou Empilhar ou push)

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

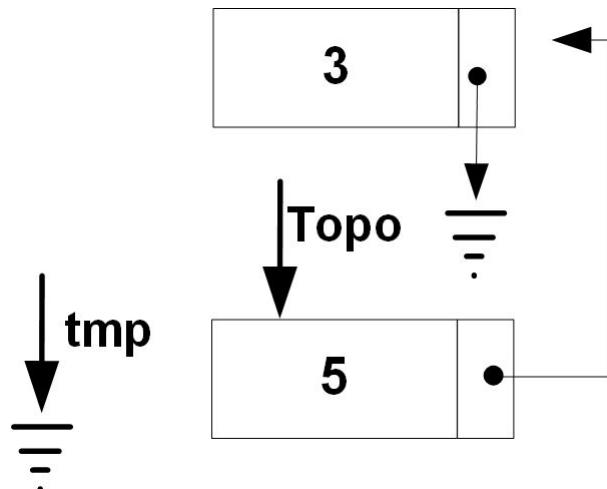
```
public void inserir(int x) { //Inserir(5)
    Celula tmp = new Celula(x);
    tmp.prox = topo;
    topo = tmp;
    tmp = null;
}
```



# Inserir (ou Empilhar ou push)

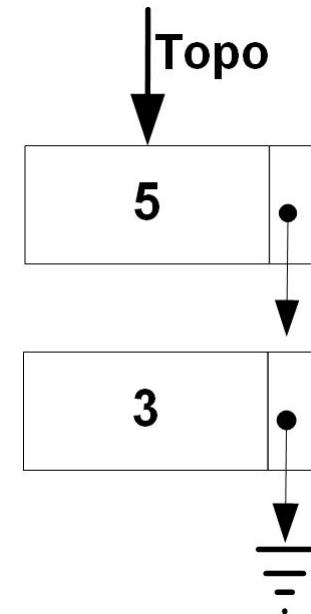
```
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}
```

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



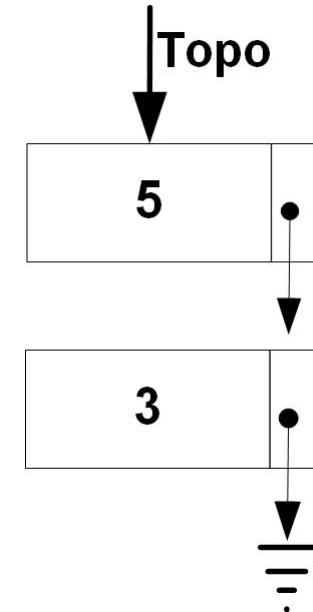
# Classe Pilha

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



# Inserir (ou Empilhar ou push)

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



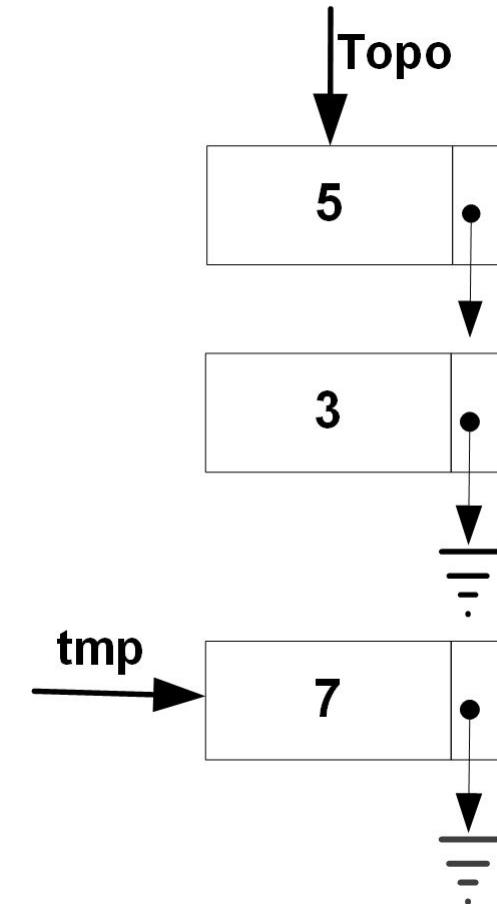
```
public void inserir(int x) { //Inserir(7)  
    Celula tmp = new Celula(x);  
    tmp.prox = topo;  
    topo = tmp;  
    tmp = null;  
}
```

# Inserir (ou Empilhar ou push)

```
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)}**

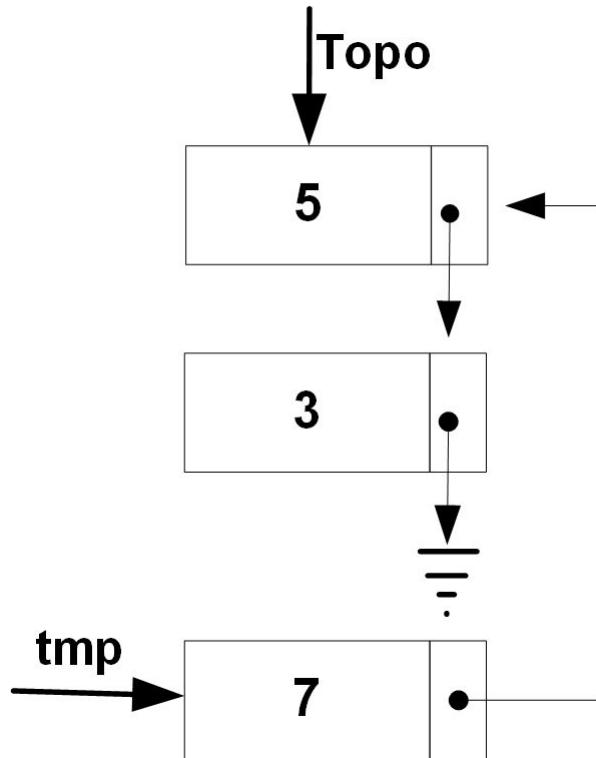
```
Celula tmp = new Celula(x);
tmp.prox = topo;
topo = tmp;
tmp = null;
```



# Inserir (ou Empilhar ou push)

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

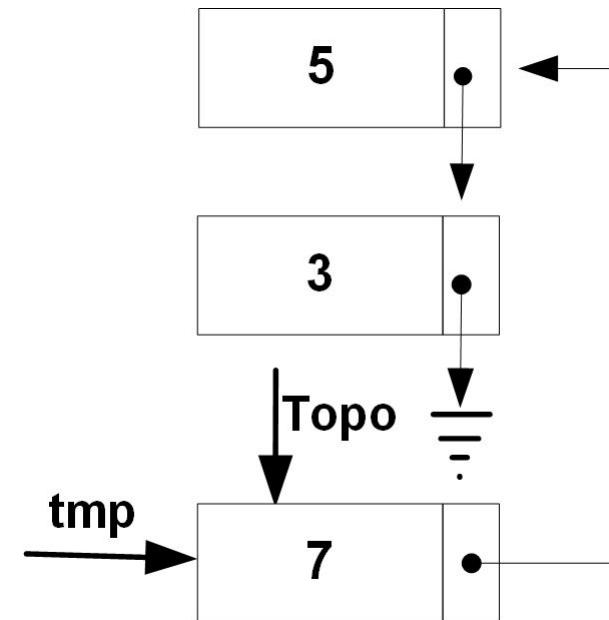
```
public void inserir(int x) { //Inserir(7)
    Celula tmp = new Celula(x);
    tmp.prox = topo;
    topo = tmp;
    tmp = null;
}
```



# Inserir (ou Empilhar ou push)

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

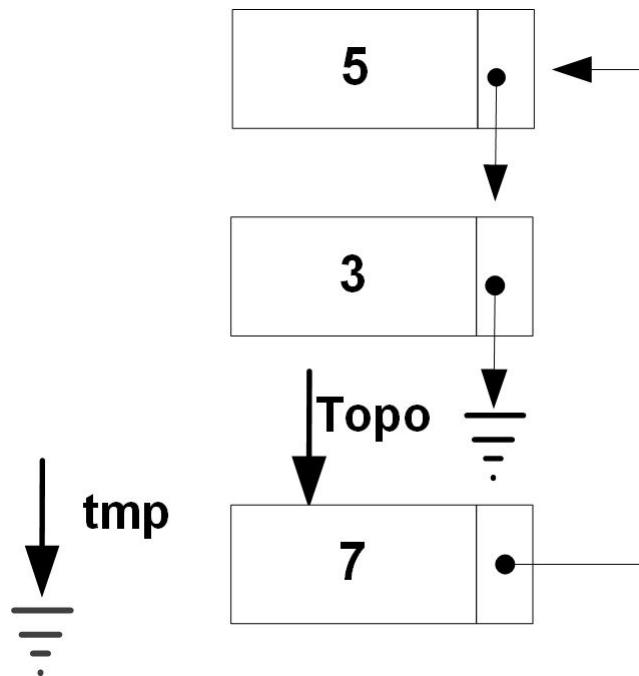
```
public void inserir(int x) { //Inserir(7)
    Celula tmp = new Celula(x);
    tmp.prox = topo;
    topo = tmp;
    tmp = null;
}
```



# Inserir (ou Empilhar ou push)

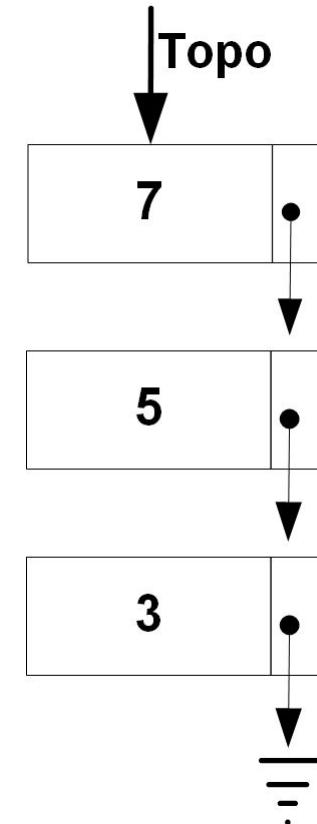
```
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}
```

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



## Classe Pilha

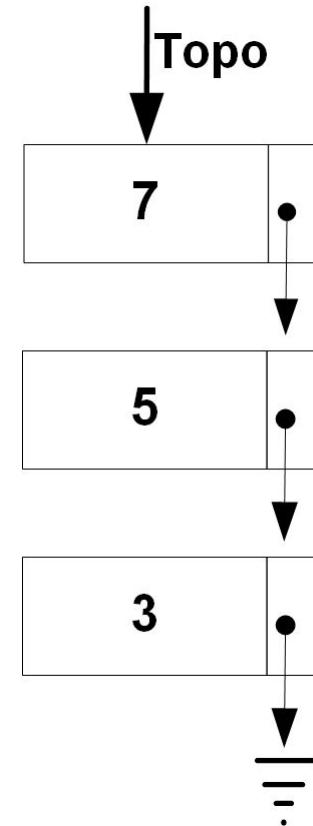
```
class Pilha {  
    private Celula topo;  
    public Pilha () {  
        topo = null;  
    }  
    public void inserir(int x) { ... }  
    public int remover() { ... }  
    public void mostrar() { ... }  
}
```



# Remover (ou Desempilhar ou pop)

```
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;
}
```

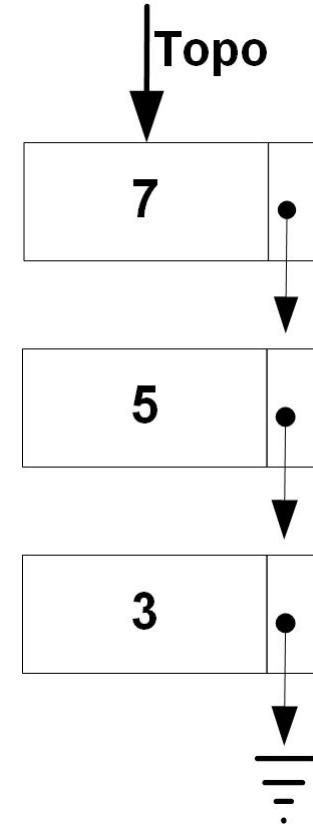


# Remover (ou Desempilhar ou pop)

```
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

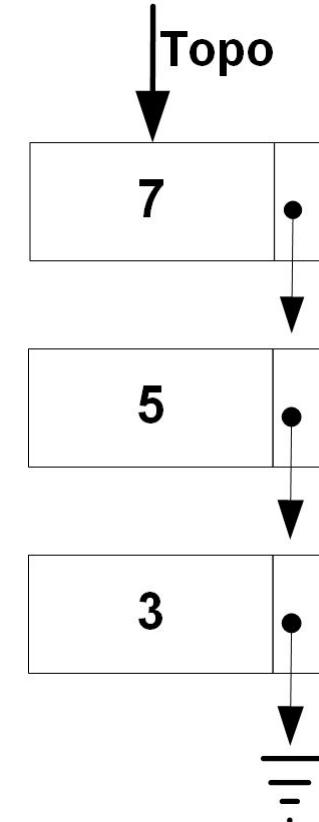


# Remover (ou Desempilhar ou pop)

```
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;
}
```

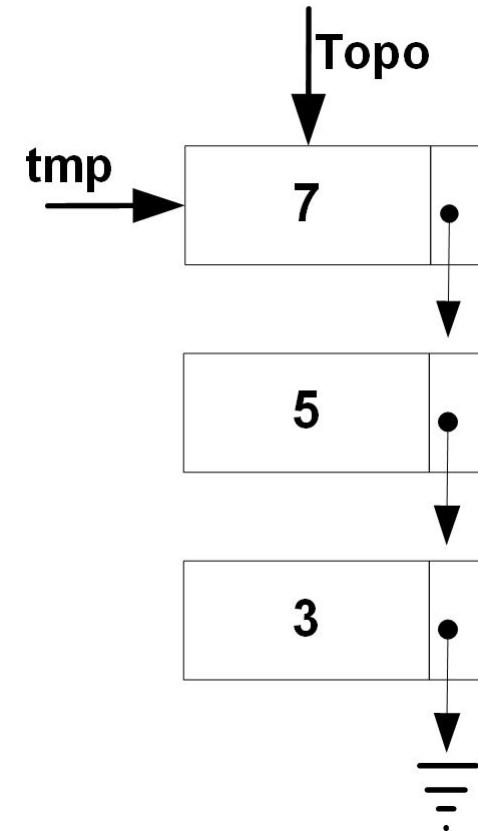


elemento 7

# Remover (ou Desempilhar ou pop)

```
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;
}
```

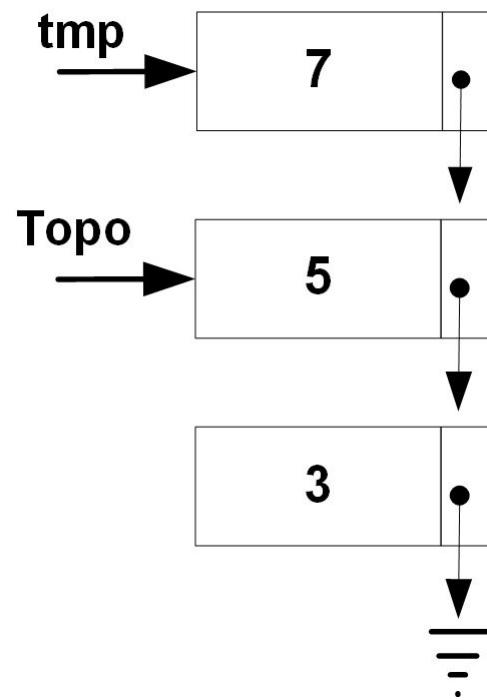


elemento 7

# Remover (ou Desempilhar ou pop)

```
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;
}
```

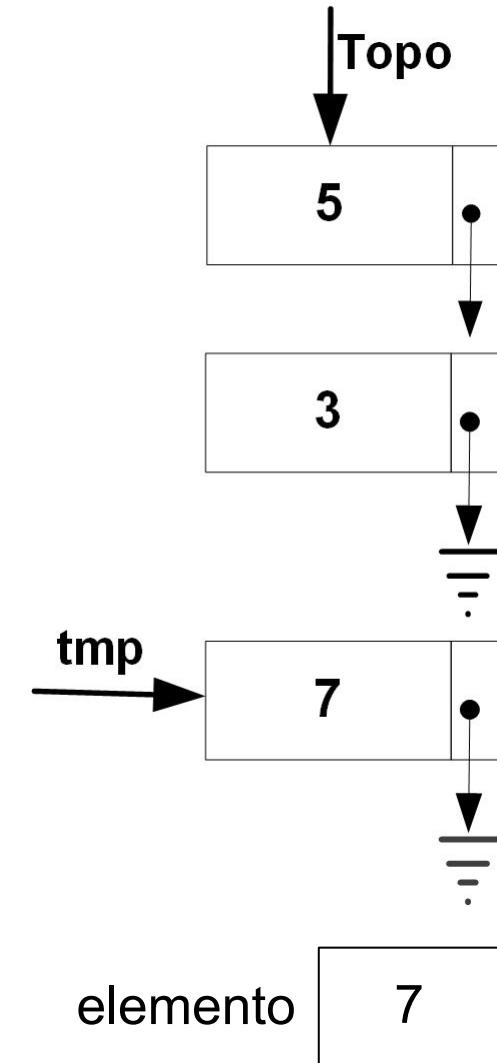


elemento 7

# Remover (ou Desempilhar ou pop)

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

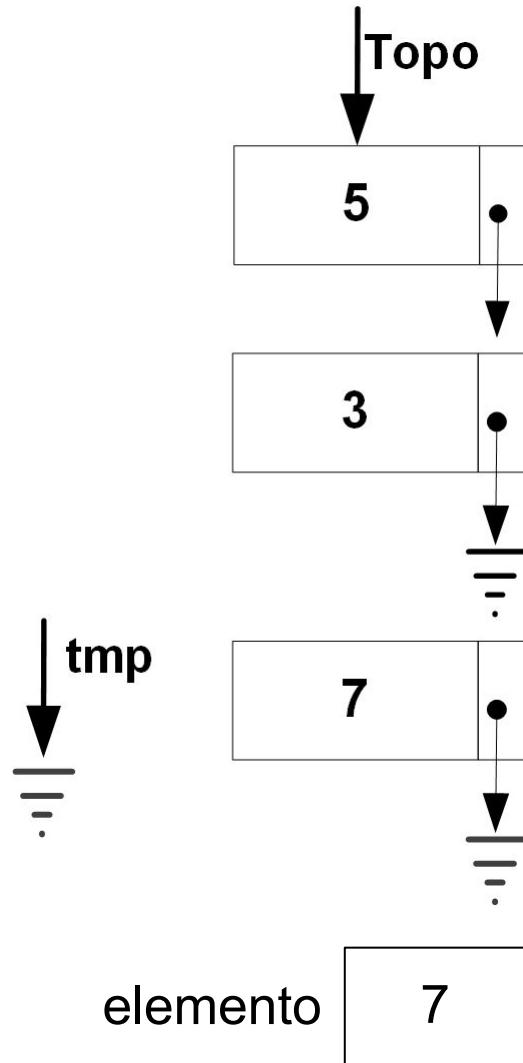
```
public int remover() throws Exception {
    if (topo == null)
        throw new Exception("Erro!");
    int elemento = topo.elemento;
    Celula tmp = topo;
    topo = topo.prox;
    tmp.prox = null;
    tmp = null;
    return elemento;
}
```



# Remover (ou Desempilhar ou pop)

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

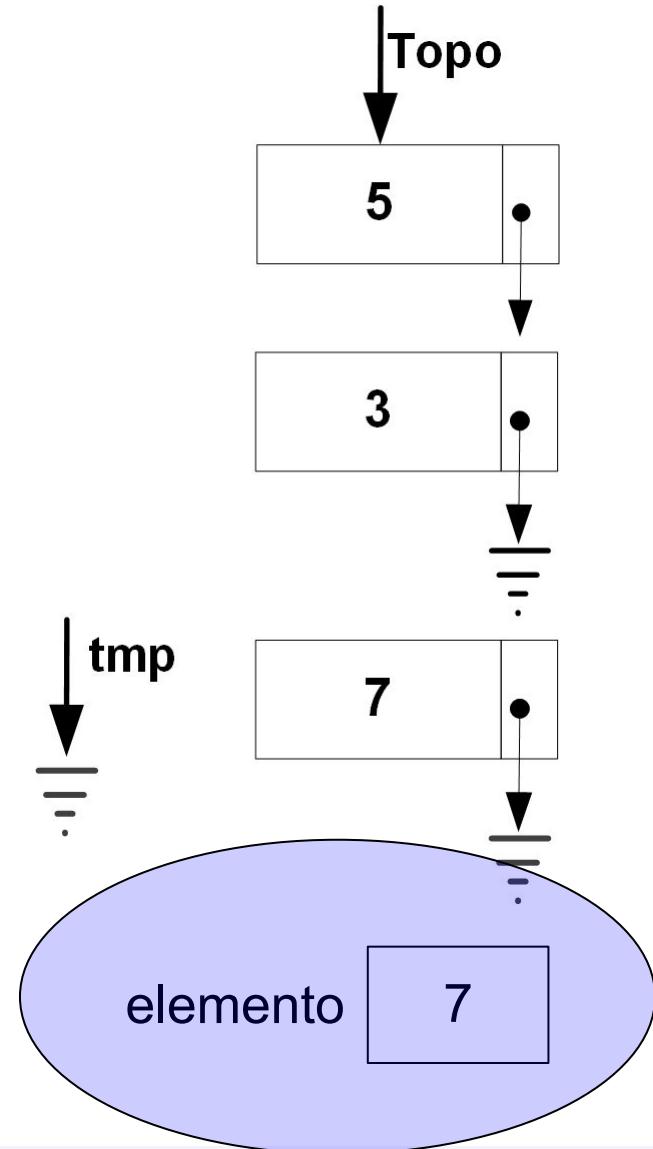
```
public int remover() throws Exception {
    if (topo == null)
        throw new Exception("Erro!");
    int elemento = topo.elemento;
    Celula tmp = topo;
    topo = topo.prox;
    tmp.prox = null;
    tmp = null;
    return elemento;
}
```



# Remover (ou Desempilhar ou pop)

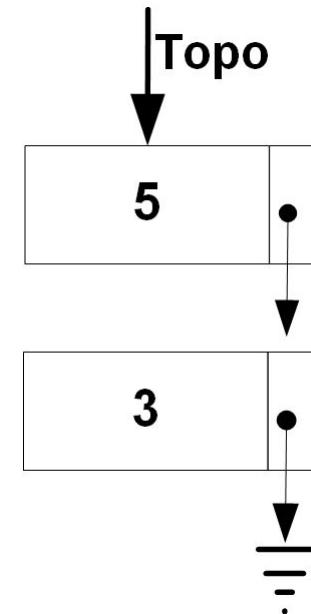
```
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}
```

```
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;
}
```



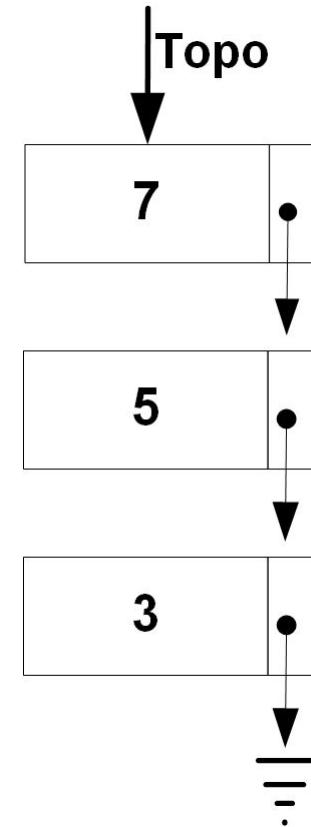
# Classe Pilha

```
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() { ... }  
}
```



```

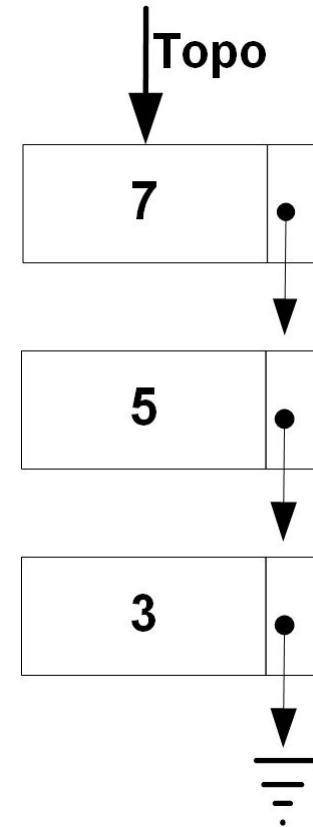
class Pilha {
    private Celula topo;
    public Pilha () {
        topo = null;
    }
    public void inserir(int x) { ... }
    public int remover() { ... }
    public void mostrar() { ... }
}

```

```

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

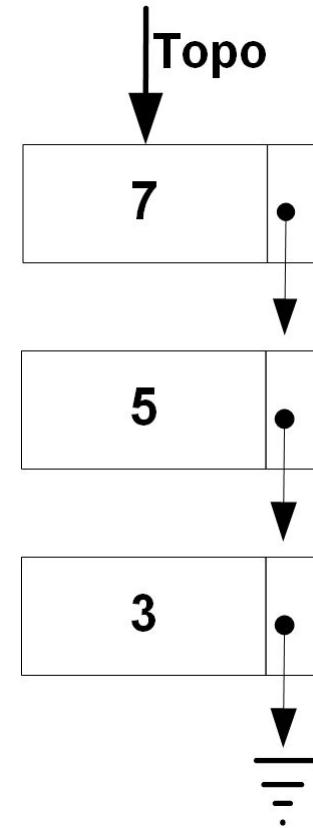
```



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("]");
}
```



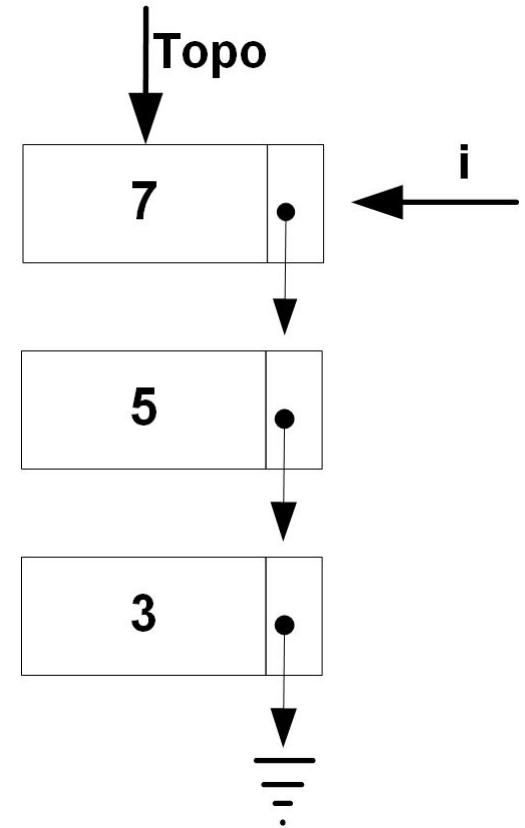
Saída  
na tela

[

## 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("]");
}
```



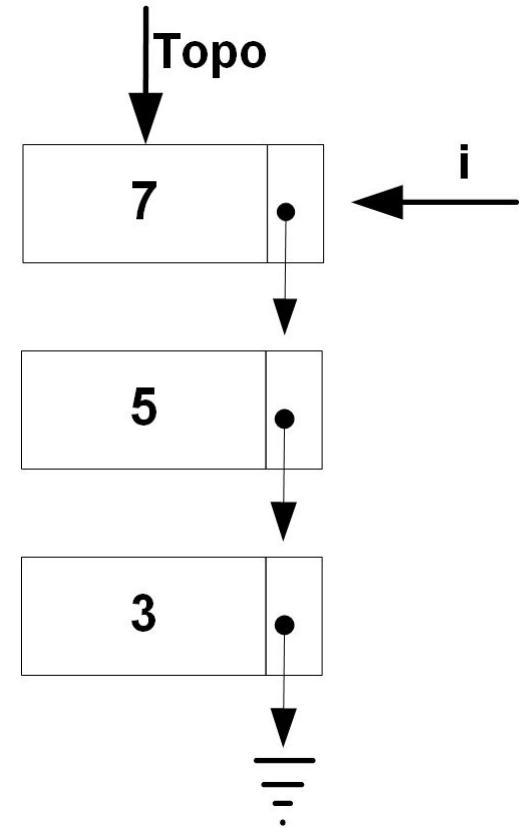
Saída  
na tela

[

## 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("]");
}
```



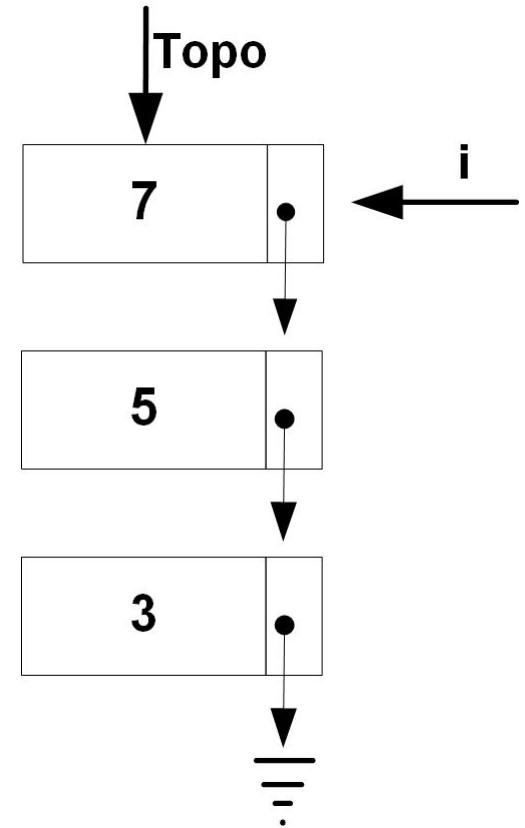
Saída  
na tela

[

## 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("]");
}
```



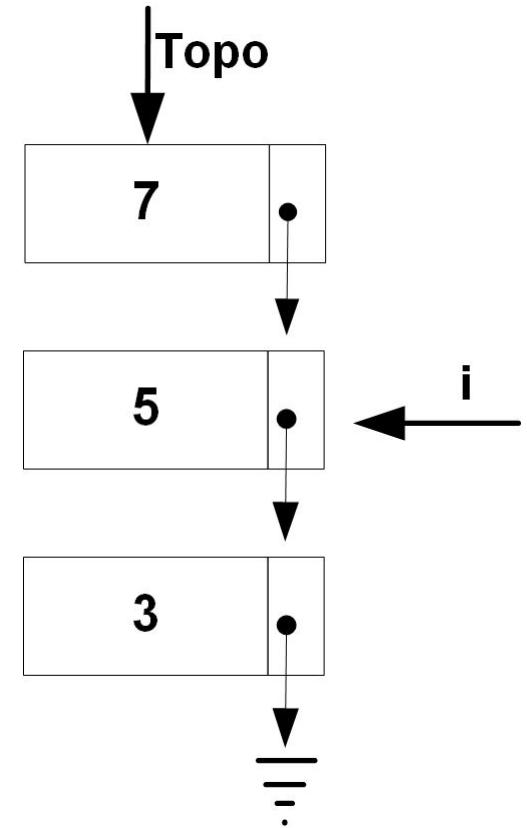
Saída  
na tela

[ 7

## 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("]");
}
```



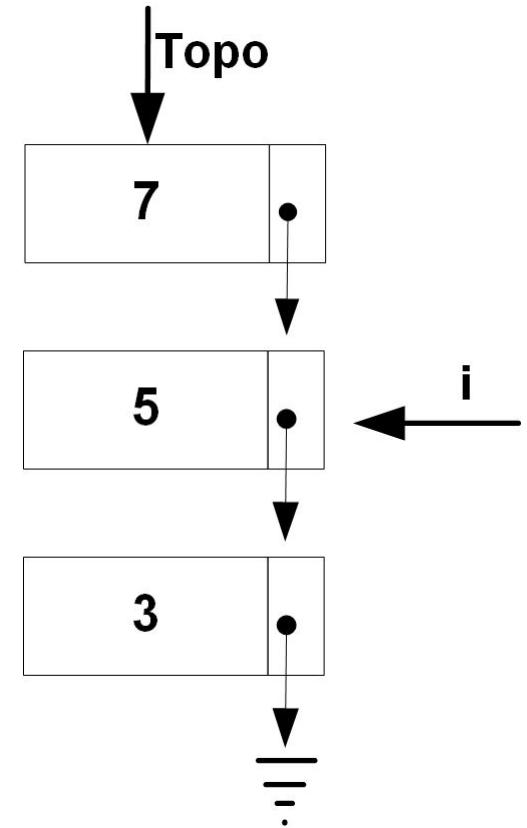
Saída  
na tela

[ 7

## 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("]");
}
```



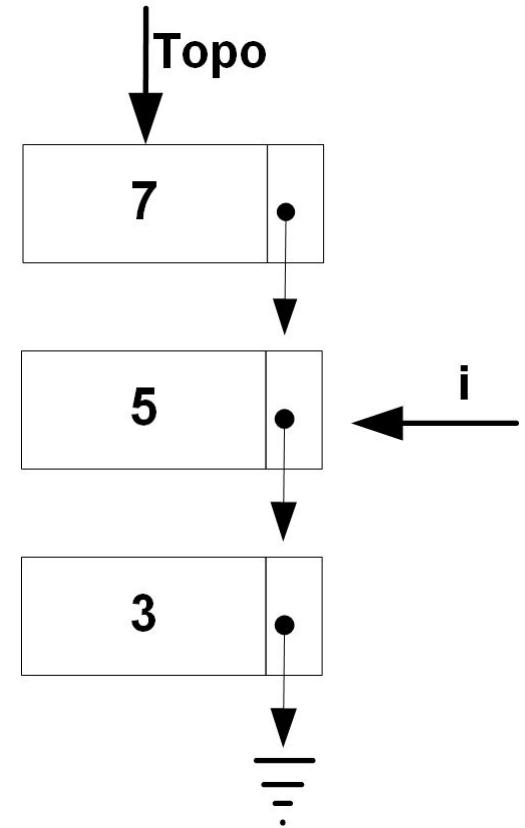
Saída  
na tela

[ 7

## 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("]");
}
```



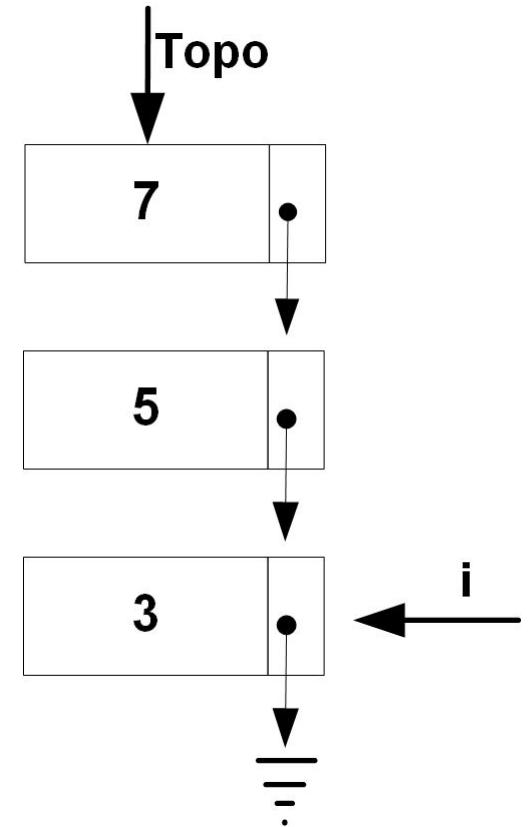
Saída  
na tela

[ 7 5

## 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("]");
}
```



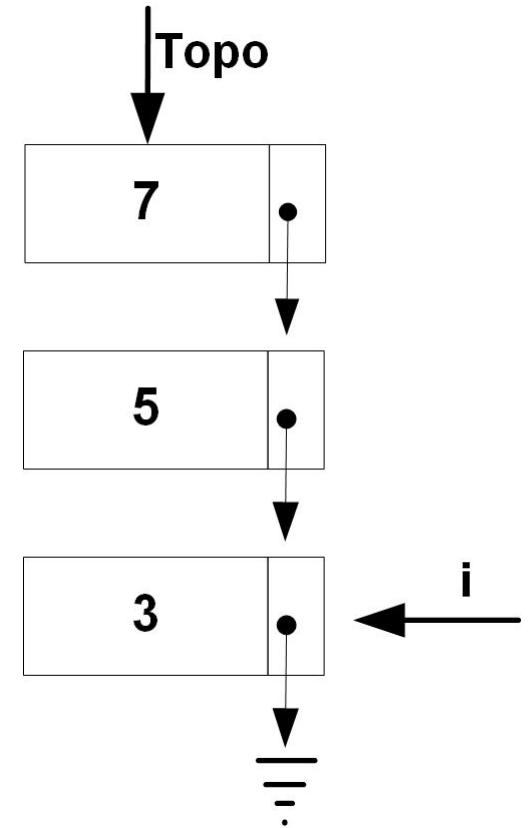
Saída  
na tela

[ 7 5 ]

## 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("]");
}
```



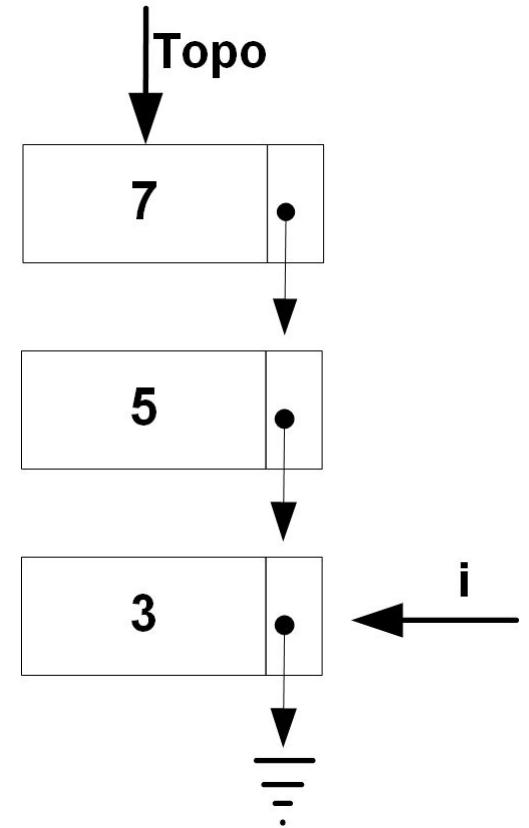
Saída  
na tela

[ 7 5

## 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("]");
}
```



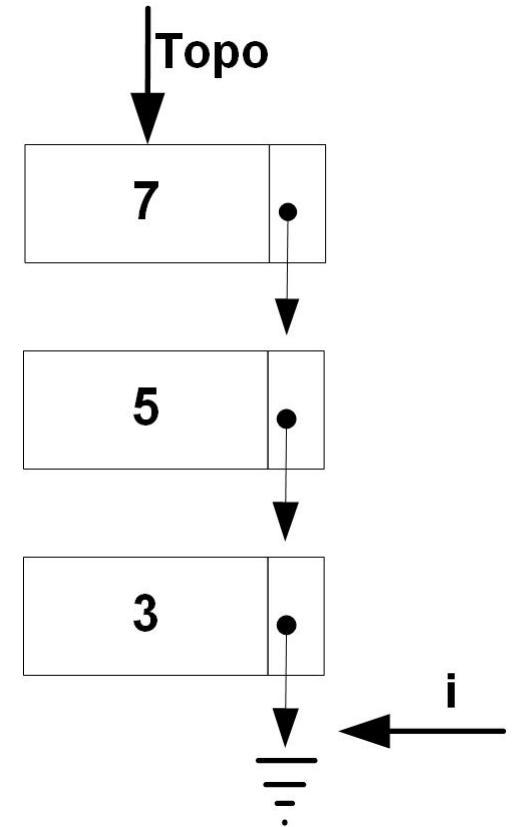
Saída  
na tela

[ 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("]");
}
```



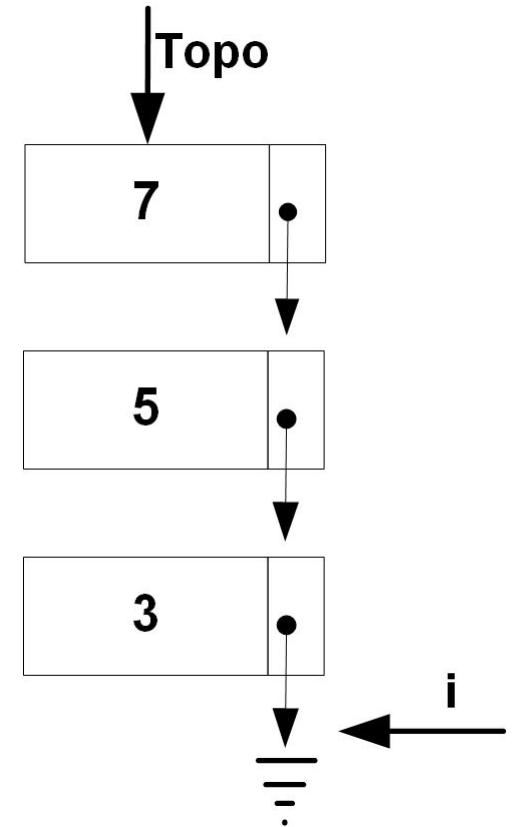
Saída  
na tela

[ 7 5 3

## 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("]");
}
```

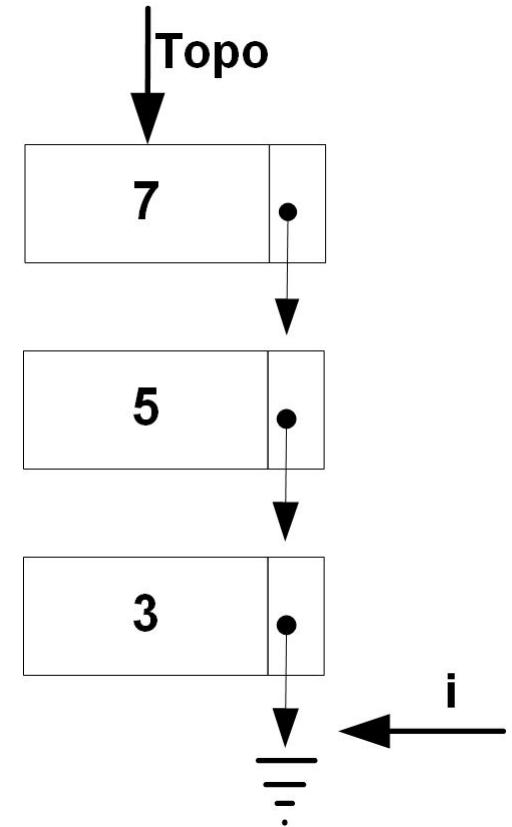
Saída  
na tela

[ 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 ]

# Exercício Resolvido (1)

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

# Exercício Resolvido (1)

- Seja nossa Pilha, faça um método que retorna soma 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 (1)

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

## Exercício (2)

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

## Exercício (3)

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

## Exercício (4)

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

## Exercício (5)

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

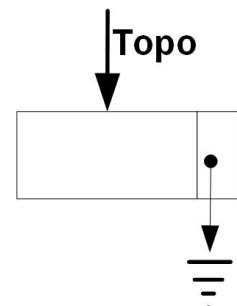
## Exercício (6)

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

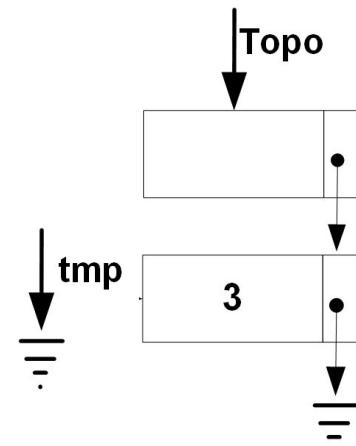
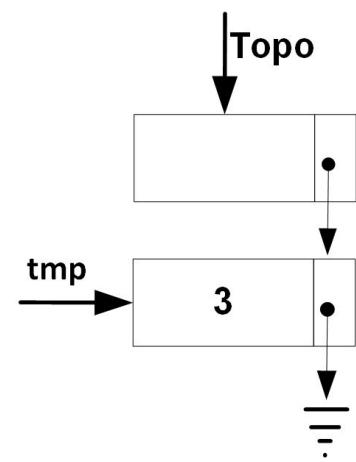
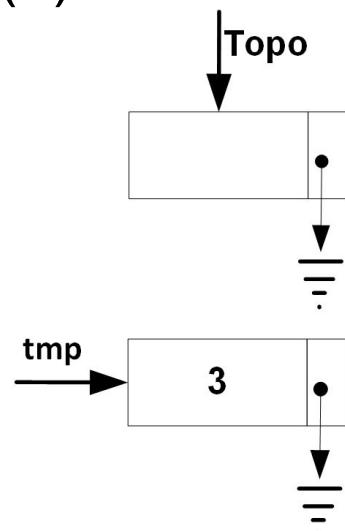
## Exercício (7)

- As ilustrações abaixo mostram a execução dos métodos construtor e do inserir de uma pilha, apresente o código dessa classe e desses métodos

## (1) Construtor



## (2) Inserir 3



## Exercício (7)

- As ilustrações abaixo mostram a execução dos métodos construtor e do inserir de uma pilha, apresente o código dessa classe e desses métodos
- (3) Inserir 5

