

# **Unidade VI:**

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

### **Lista Simples**



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

- As listas flexíveis simples também são chamadas de:
  - Listas simplesmente encadeadas
  - Listas simplesmente ligadas
  - Listas ligadas
  - ...

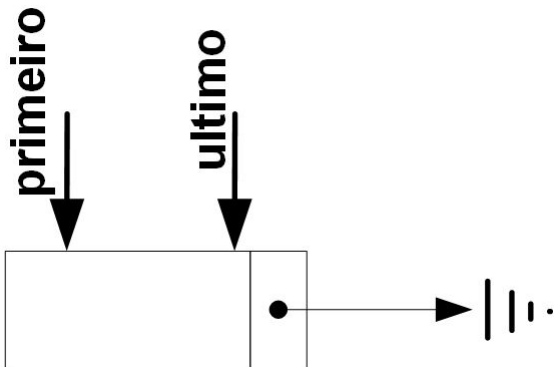
- [PrincipalLista.java](#), igual ao da estrutura sequencial
- [Lista.java](#), tem os atributos primeiro e início e os métodos abaixo:
  - Inserir no início
  - Inserir no fim
  - Inserir
  - Remover no início
  - Remover no fim
  - Remover

## Classe Lista Simples

```
class Lista {  
    private Celula primeiro, ultimo;  
    public Lista () {  
        primeiro = new Celula();  
        ultimo = primeiro;  
    }  
    public void inserirInicio(int x) { ... }  
    public void inserirFim(int x) { ... }  
    public int removerInicio() { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
    public void mostrar() { ... }  
}
```

# Classe Lista Simples

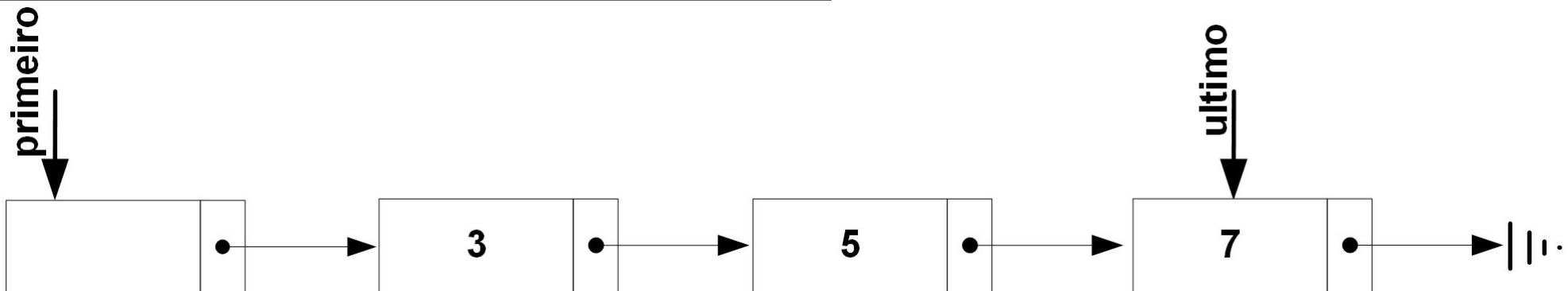
```
class Lista {  
    private Celula primeiro, ultimo;  
    public Lista () {  
        primeiro = new Celula();  
        ultimo = primeiro;  
    }  
    public void inserirInicio(int x) { ... }  
    public void inserirFim(int x) { ... }  
    public int removerInicio() { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
    public void mostrar() { ... }  
}
```



## Classe Lista Simples

```
class Lista {  
    private Celula primeiro, ultimo;  
    public Lista () {  
        primeiro = new Celula();  
        ultimo = primeiro;  
    }  
    public void inserirInicio(int x) { ... }  
    public void inserirFim(int x) { ... }  
    public int removerInicio() { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
    public void mostrar() { ... }  
}
```

Iguais aos métodos da fila



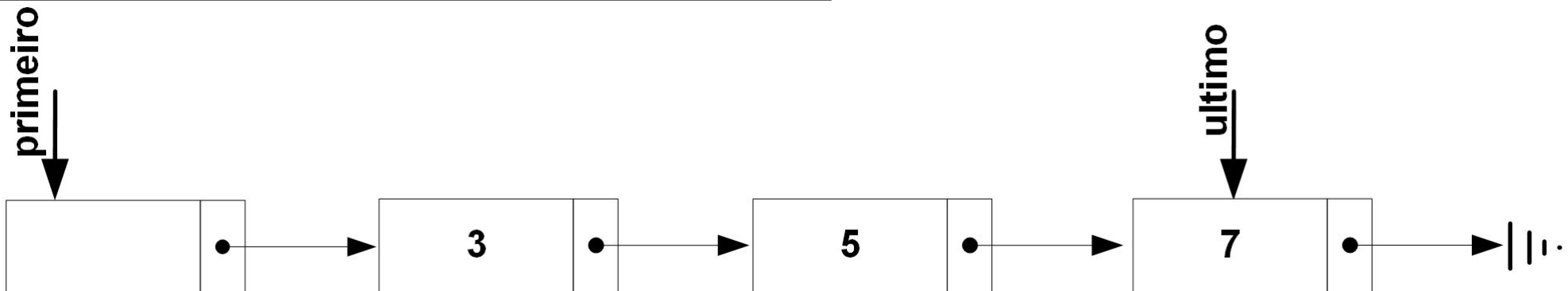
## Classe Lista Simples

```

class Lista {
    private Celula primeiro, ultimo;
    public Lista () {
        primeiro = new Celula();
        ultimo = primeiro;
    }
    public void inserirInicio(int x) { ... }
    public void inserirFim(int x) { ... }
    public int removerInicio() { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
    public void mostrar() { ... }
}

```

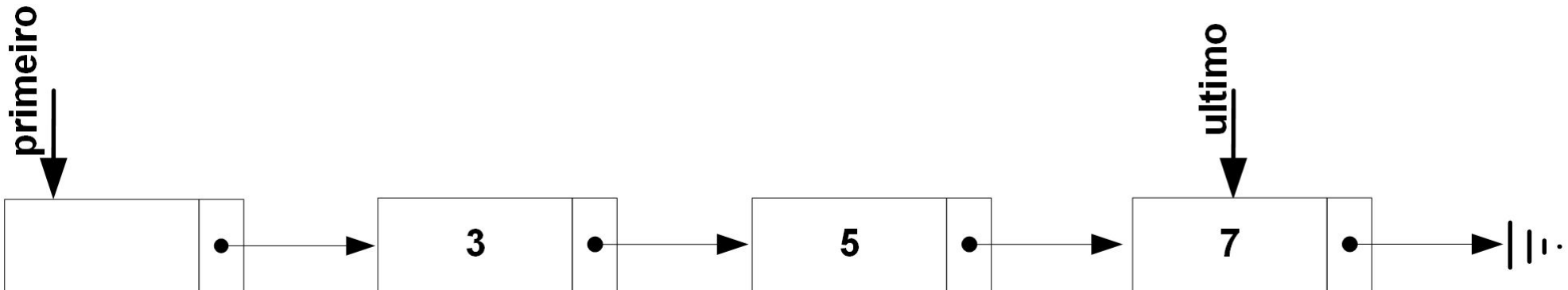
Igual aos da fila/pilha



## Classe Lista Simples

```
class Lista {  
    private Celula primeiro, ultimo;  
    public Lista () {  
        primeiro = new Celula();  
        ultimo = primeiro;  
    }  
    public void inserirInicio(int x) { ... }  
    public void inserirFim(int x) { ... }  
    public int removerInicio() { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
    public void mostrar() { ... }  
}
```

Assim, ...





# Classe Lista Simples

```
class Lista {
```

```
    public void inserirInicio(int x) { ... }
```

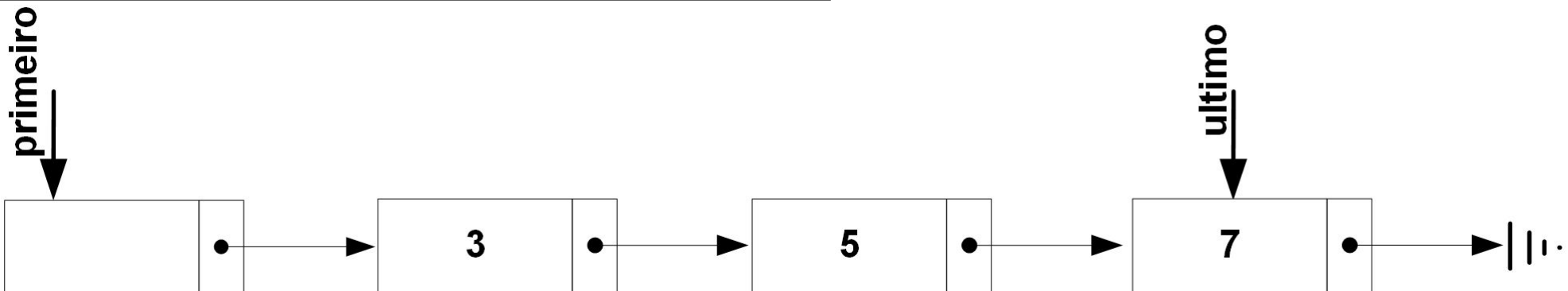
```
    public int removerFim() { ... }
```

```
    public void inserir(int x, int pos) { ... }
```

```
    public int remover(int pos) { ... }
```

```
}
```

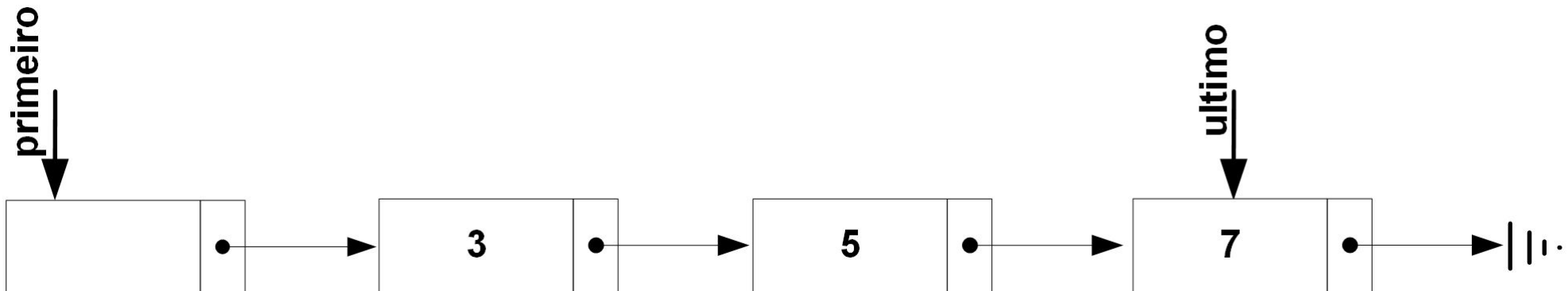
Assim, ...



# Classe Lista Simples

```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```

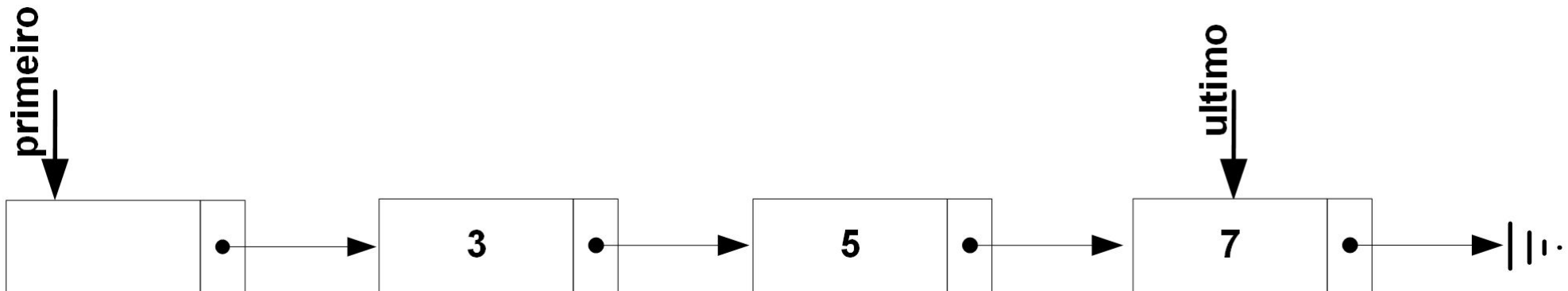
Assim, ...



## Inserir no Início

```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```

```
//inserirInicio(1)  
public void inserirInicio(int x) {  
    Celula tmp = new Celula(x);  
    tmp.prox = primeiro.prox;  
    primeiro.prox = tmp;  
    if (primeiro == ultimo) {  
        ultimo = tmp;  
    }  
    tmp = null;  
}
```



## Inserir no Início

```

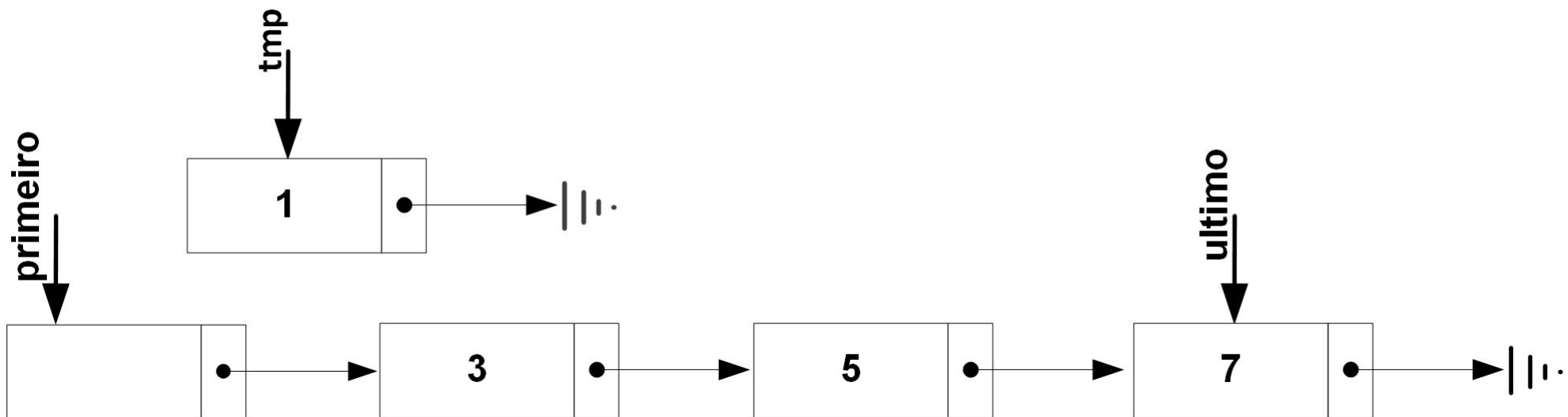
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

//inserirInicio(1)
public void inserirInicio(int x) {
    Celula tmp = new Celula(x);
    tmp.prox = primeiro.prox;
    primeiro.prox = tmp;
    if (primeiro == ultimo) {
        ultimo = tmp;
    }
    tmp = null;
}

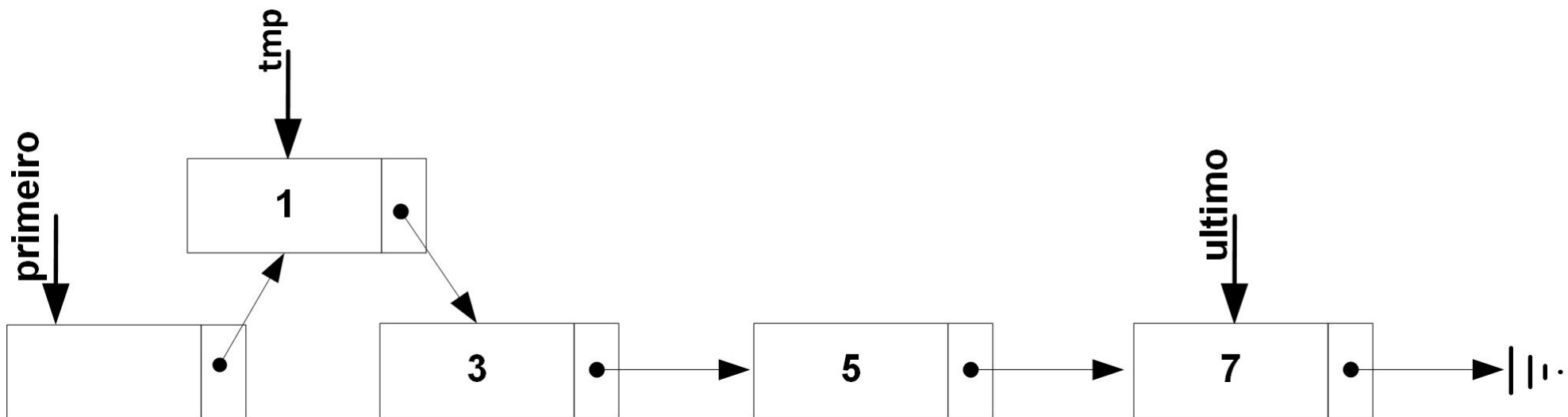
```



# Inserir no Início

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

```
//inserirInicio(1)
public void inserirInicio(int x) {
    Celula tmp = new Celula(x);
    tmp.prox = primeiro.prox;
    primeiro.prox = tmp;
    if (primeiro == ultimo) {
        ultimo = tmp;
    }
    tmp = null;
}
```

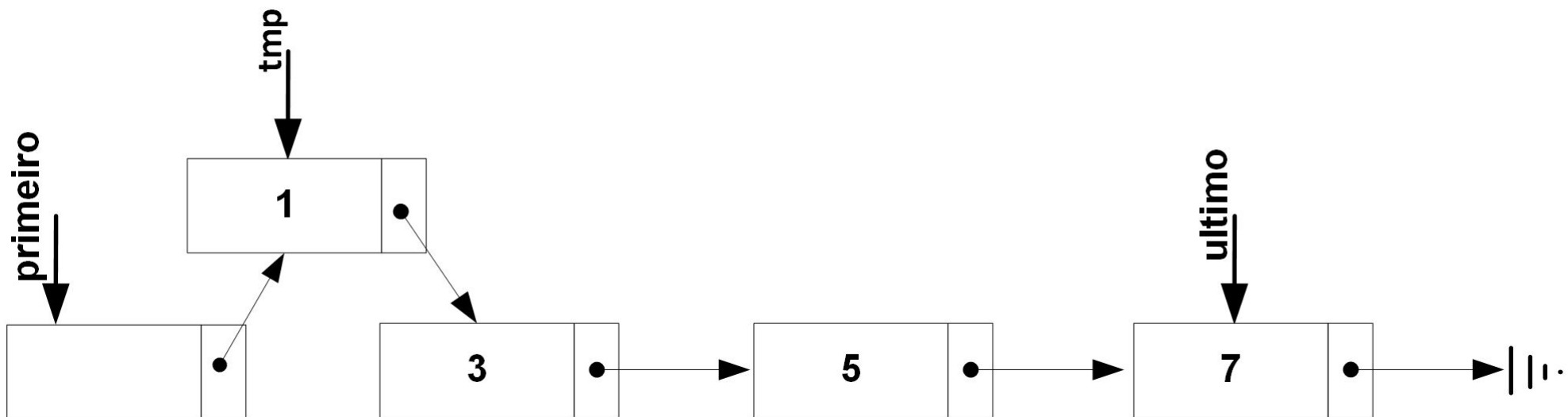


# Inserir no Início

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

//inserirInicio(1)

```
public void inserirInicio(int x) {
    Celula tmp = new Celula(x);
    tmp.prox = primeiro.prox;
    primeiro.prox = tmp;
    if (primeiro == ultimo) {
        ultimo = tmp;
    }
    tmp = null;
}
```



## Inserir no Início

```

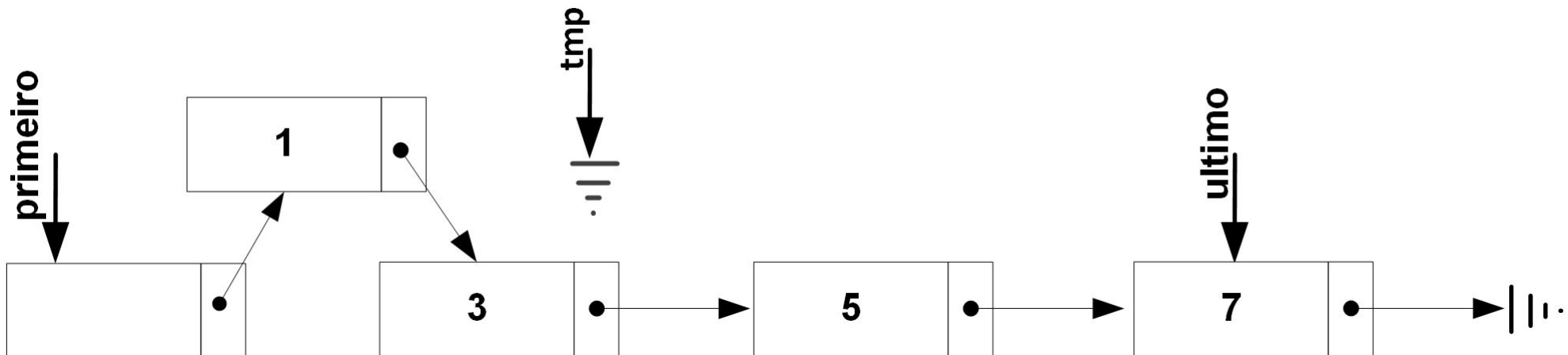
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

//inserirInicio(1)
public void inserirInicio(int x) {
    Celula tmp = new Celula(x);
    tmp.prox = primeiro.prox;
    primeiro.prox = tmp;
    if (primeiro == ultimo) {
        ultimo = tmp;
    }
    tmp = null;
}

```

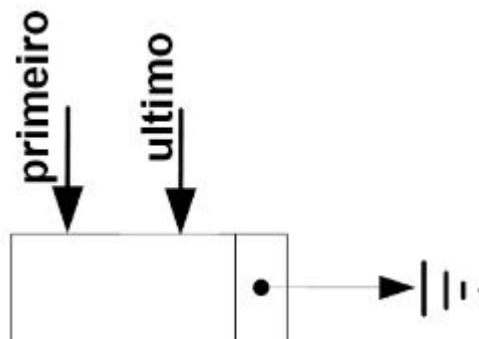


## Exercício

```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```

```
//inserirInicio(1)  
public void inserirInicio(int x) {  
    Celula tmp = new Celula(x);  
    tmp.prox = primeiro.prox;  
    primeiro.prox = tmp;  
    if (primeiro == ultimo) {  
        ultimo = tmp;  
    }  
    tmp = null;  
}
```

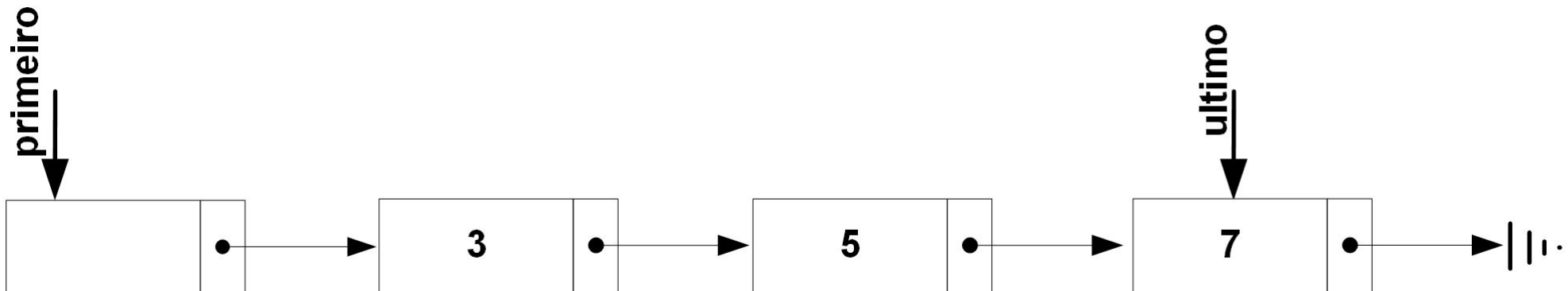
Execute o método `inserirInicio` na figura abaixo!!!





## Remover no Fim

```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```



## Remover no Fim

```

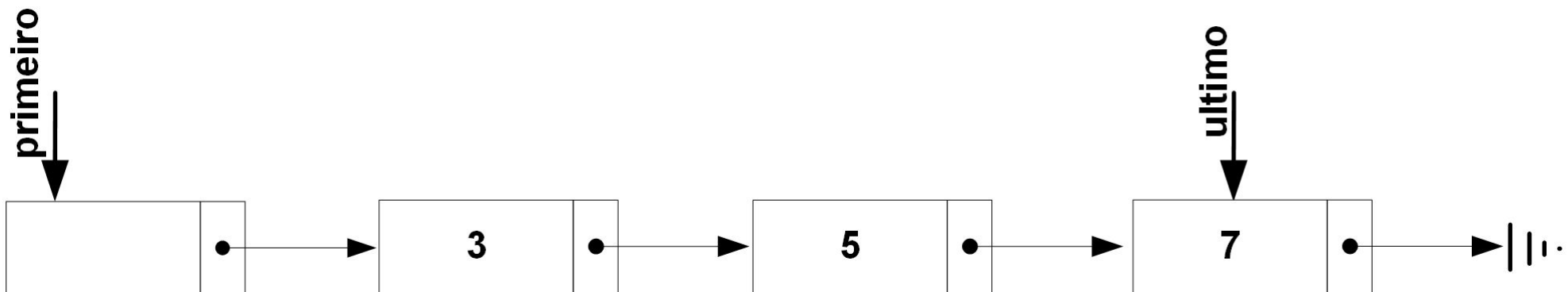
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

```



## Remover no Fim

```

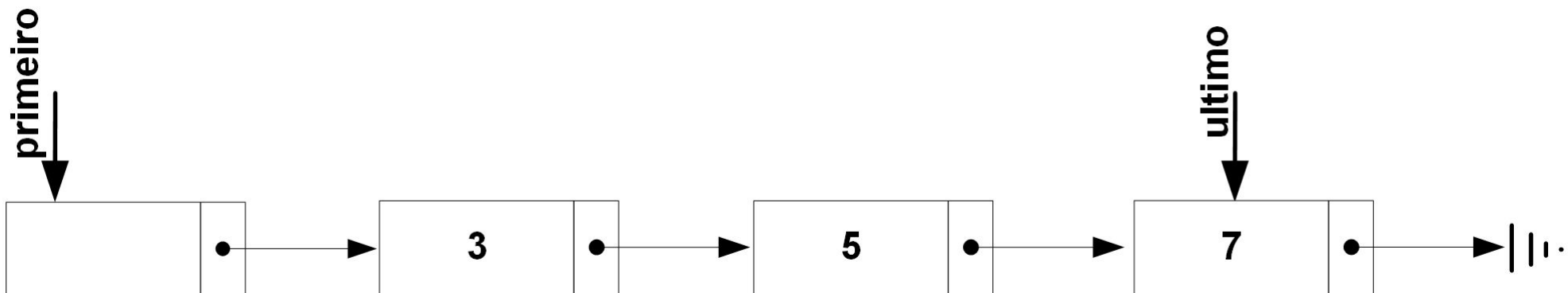
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

```



## Remover no Fim

```

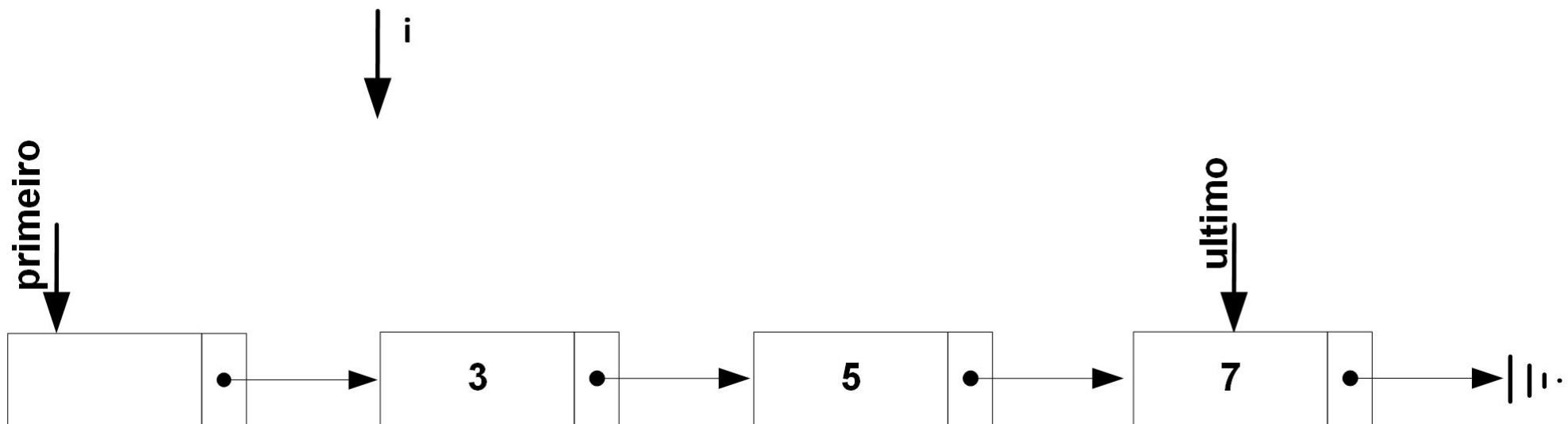
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

```



## Remover no Fim

```

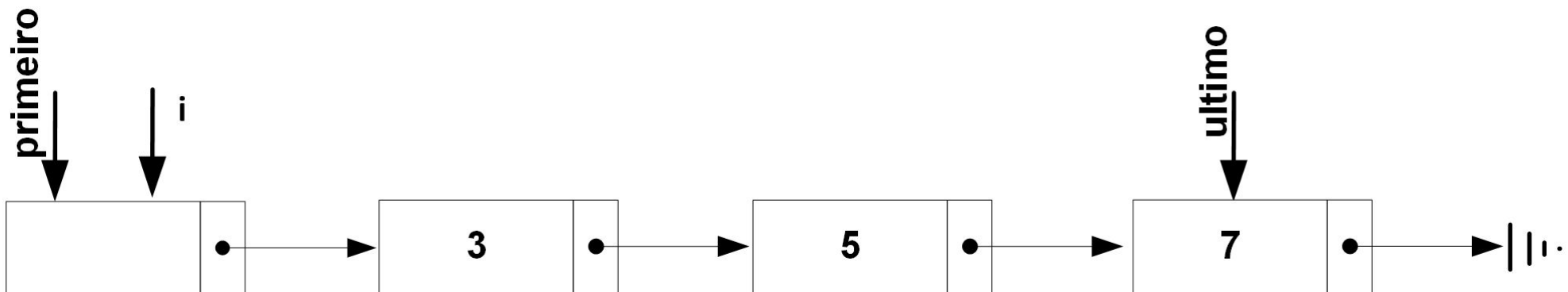
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for (i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

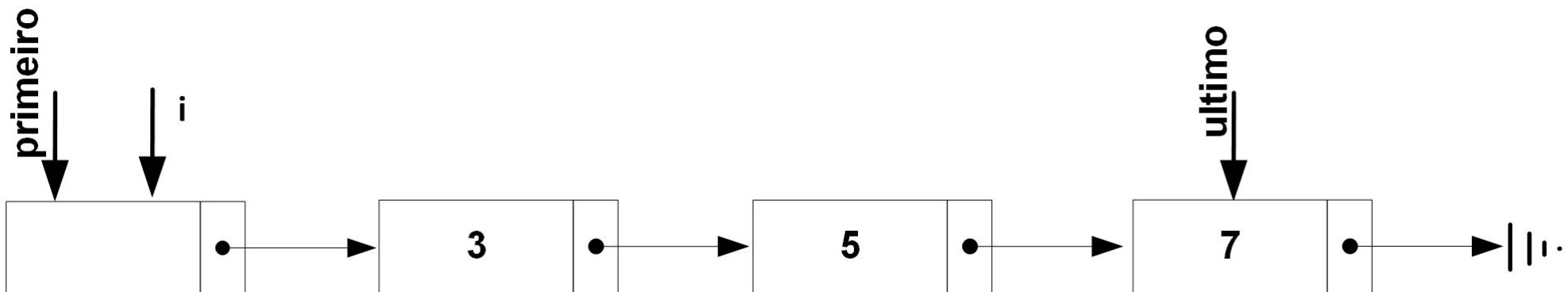
```



## Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

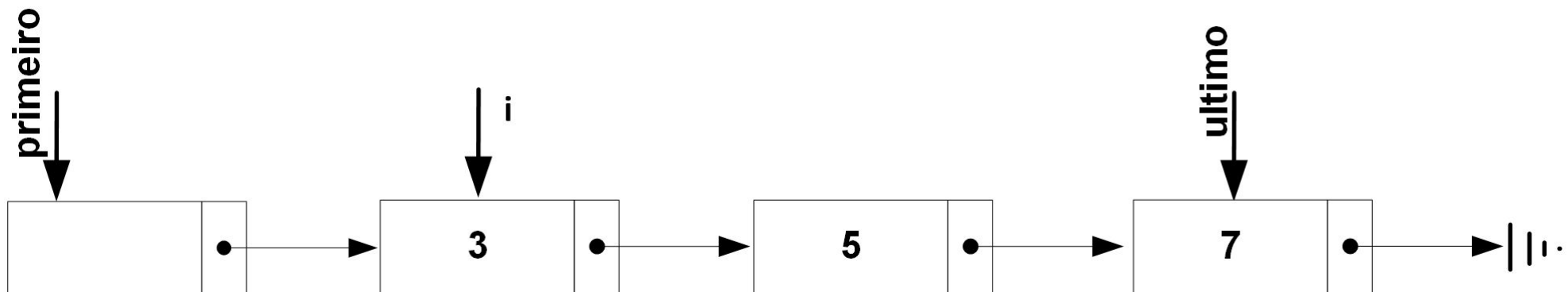
```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```



# Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

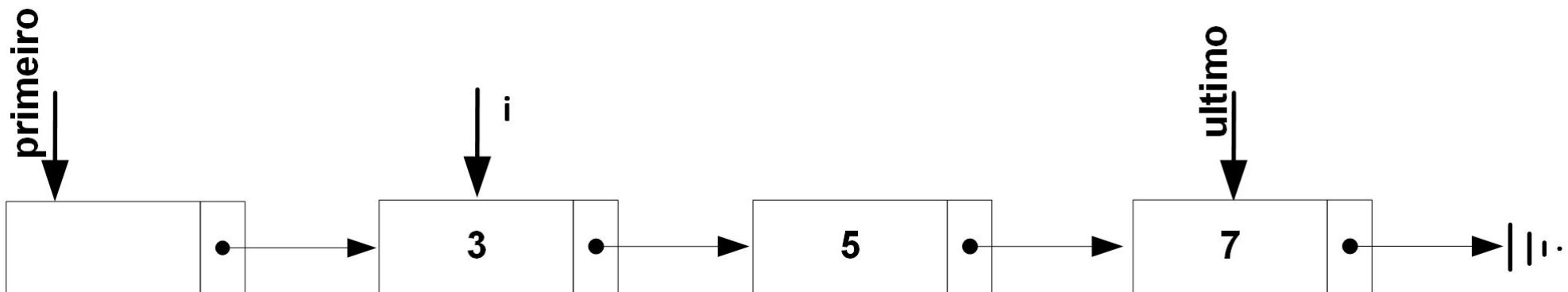
```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```



# Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```

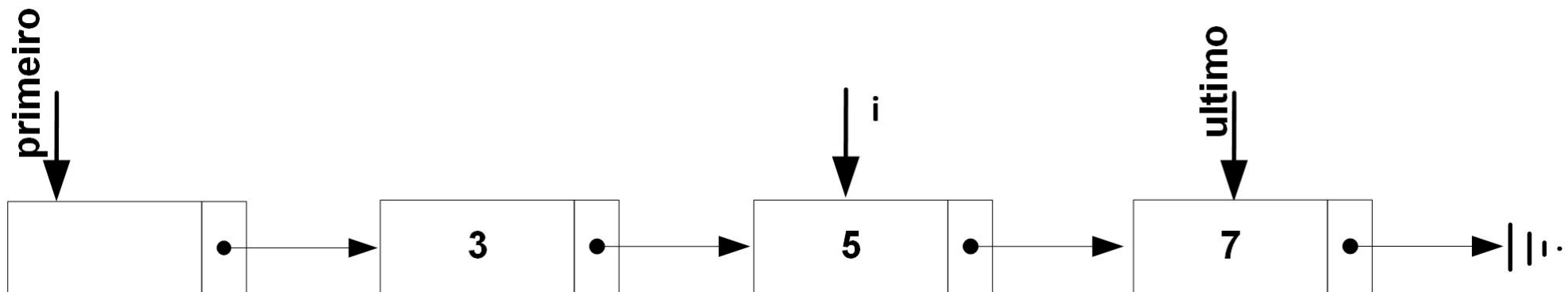




# Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```



## Remover no Fim

```

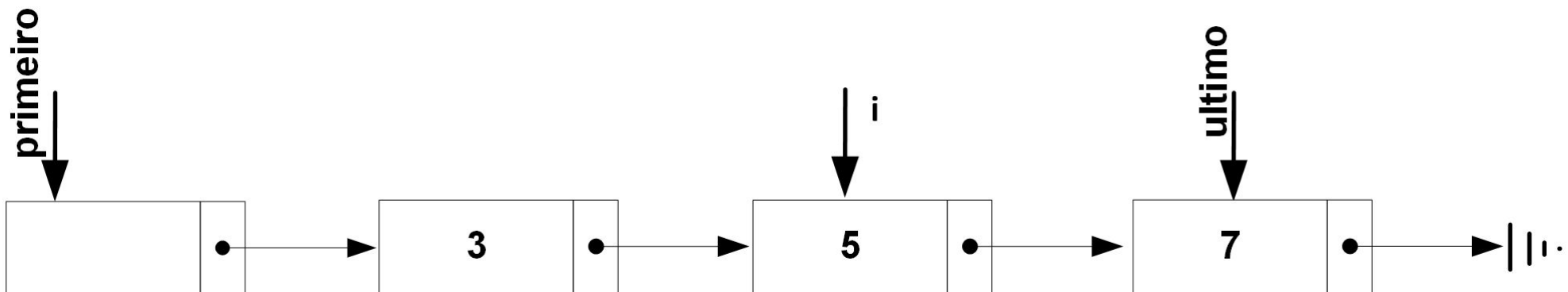
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

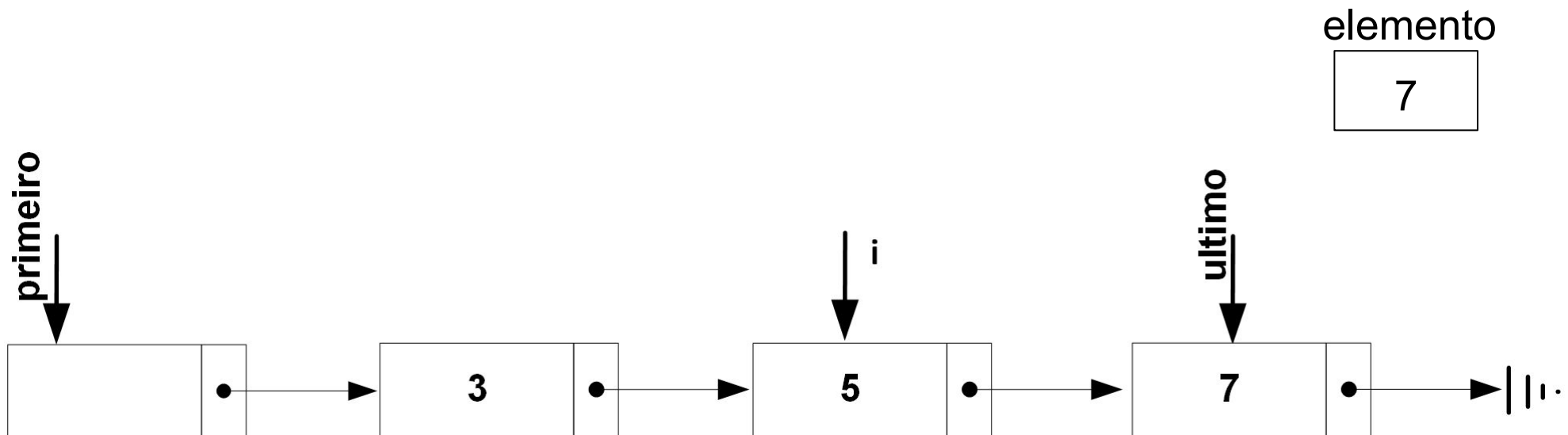
```



# Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```



## Remover no Fim

```

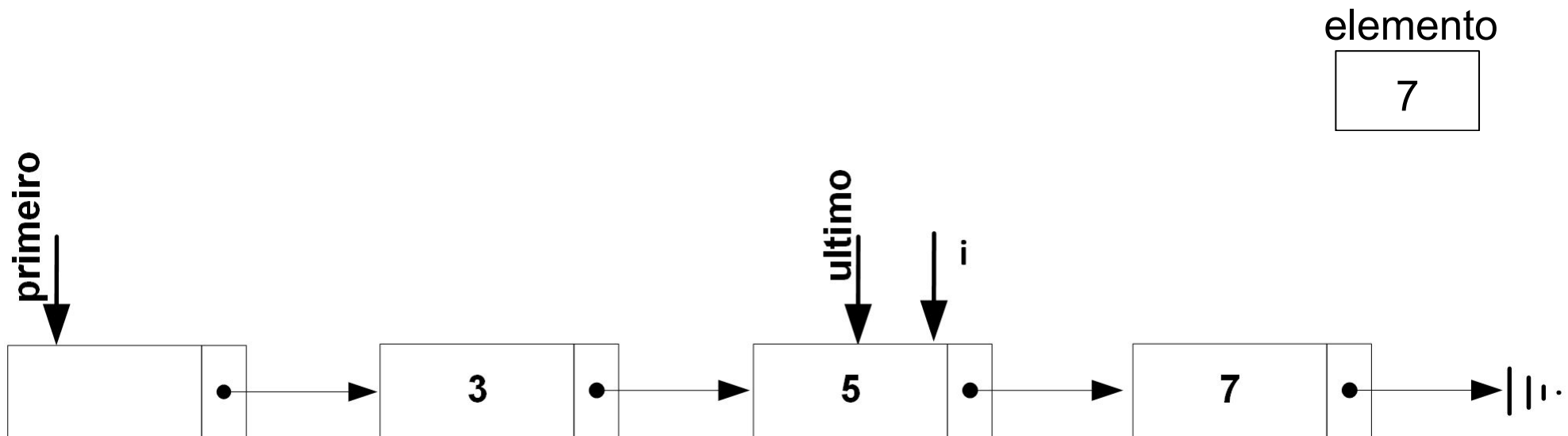
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

```



## Remover no Fim

```

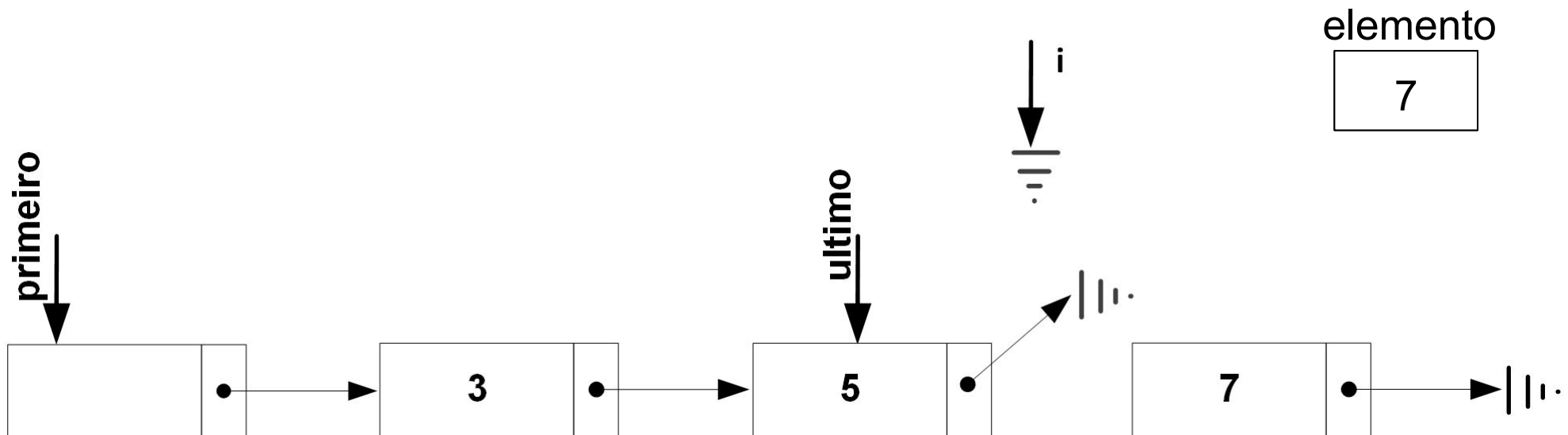
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}

```

```

public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}

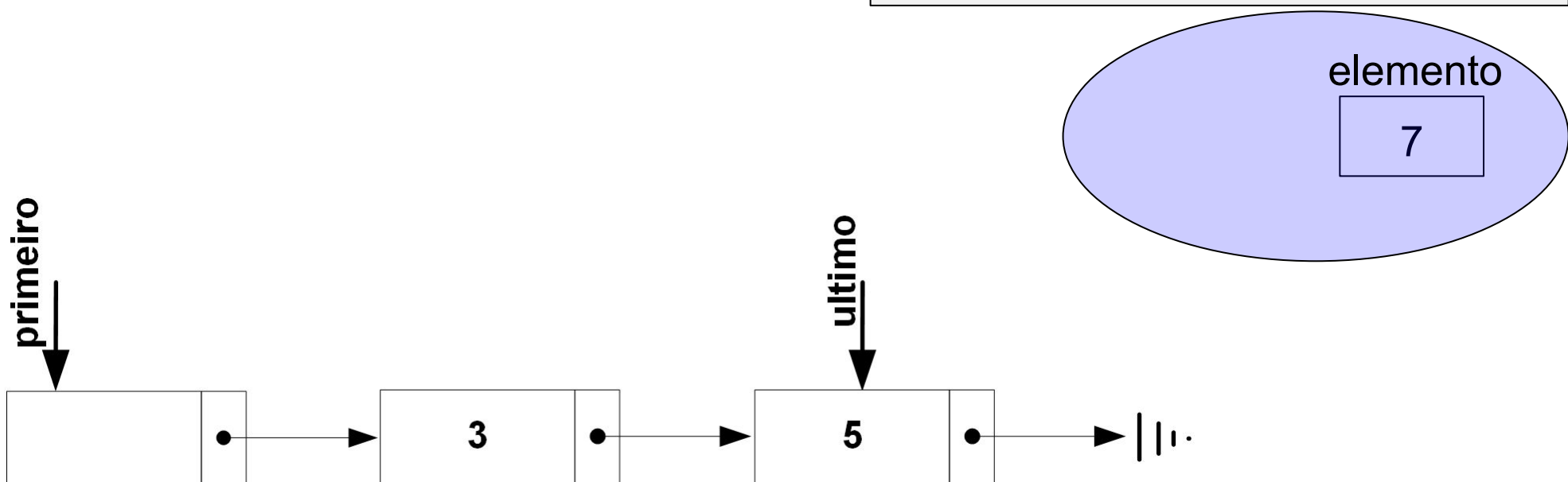
```



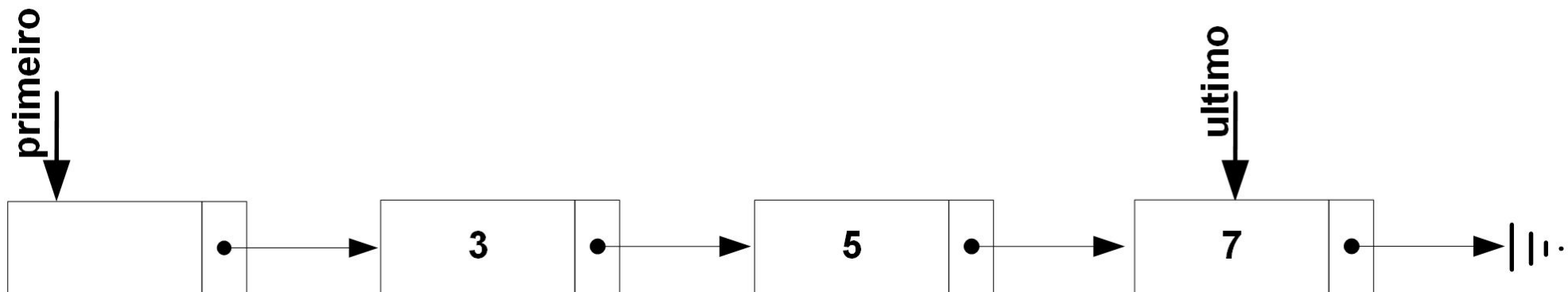
# Remover no Fim

```
class Lista {
    ...
    public void inserirInicio(int x) { ... }
    public int removerFim() { ... }
    public void inserir(int x, int pos) { ... }
    public int remover(int pos) { ... }
}
```

```
public int removerFim() throws Exception {
    if (primeiro == ultimo)
        throw new Exception("Erro!");
    Celula i;
    for(i = primeiro; i.prox != ultimo; i = i.prox);
    int elemento = ultimo.elemento;
    ultimo = i;
    i = ultimo.prox = null;
    return elemento;
}
```



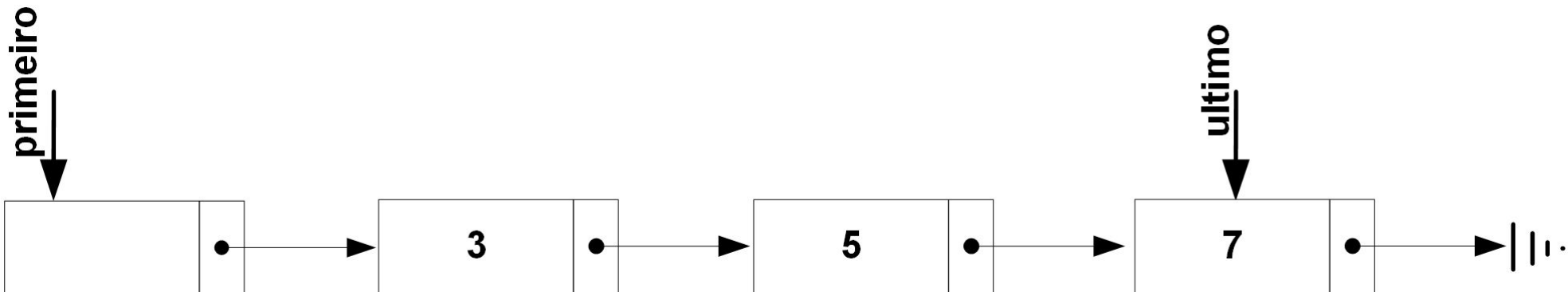
```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```



```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){      inserirInicio(x);
    } else if (pos == tamanho){  inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

```

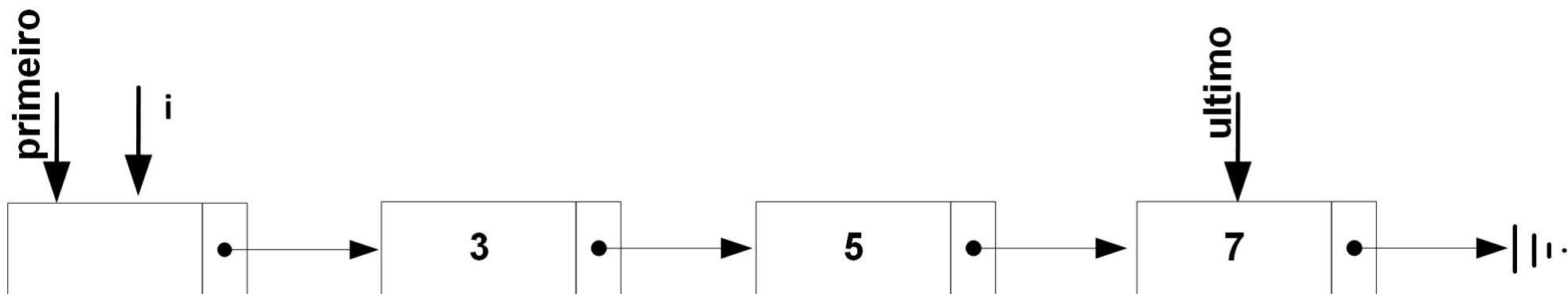




```

public void inserir(int x, int pos) throws Exception {    //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

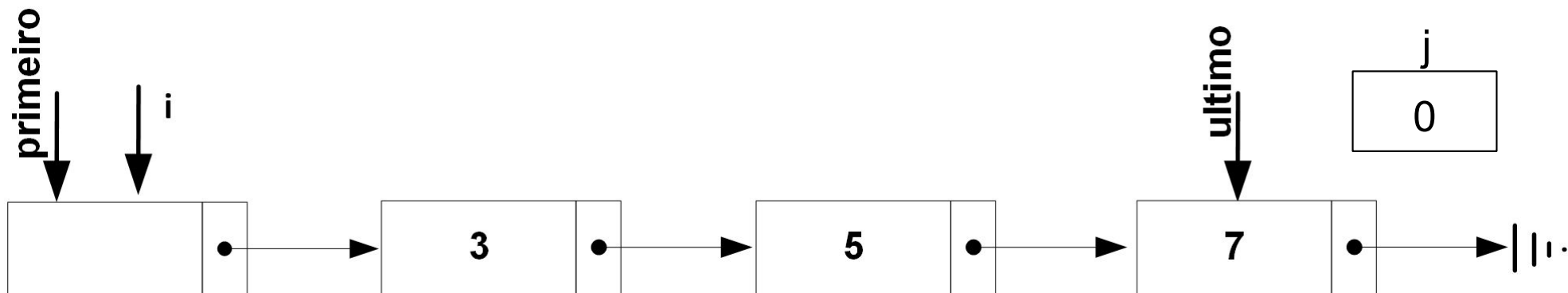
```



```

public void inserir(int x, int pos) throws Exception {    //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

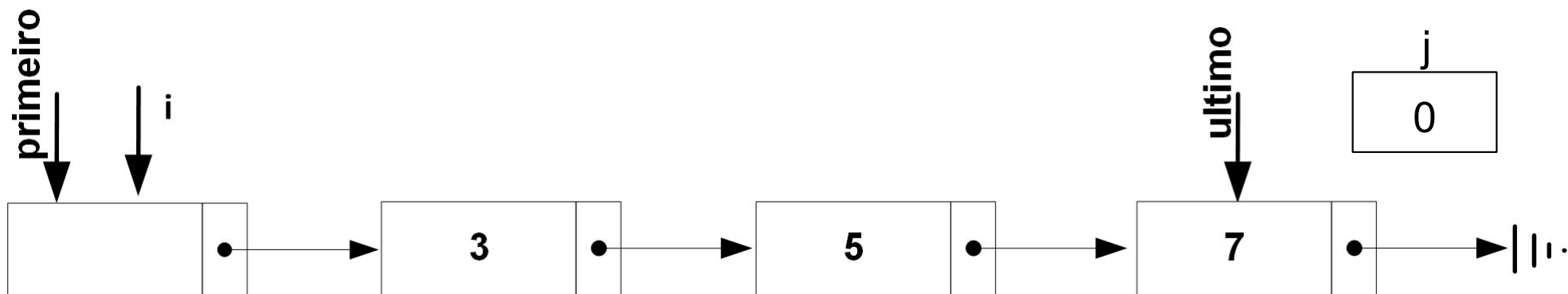


```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

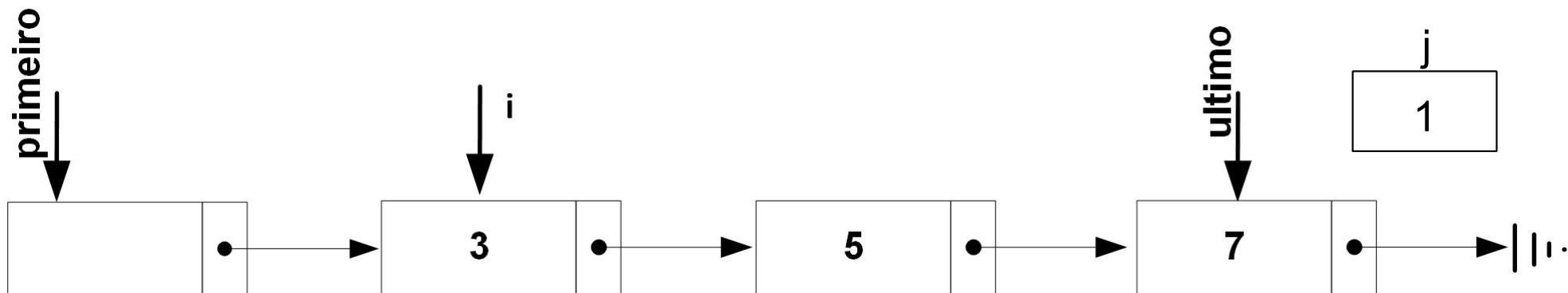
true



```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){ inserirInicio(x);
    } else if (pos == tamanho){ inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

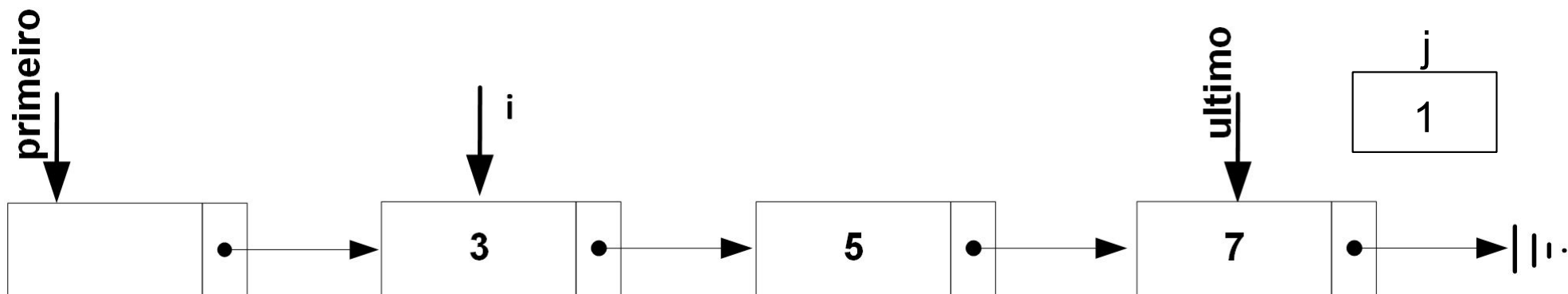


```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){   inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

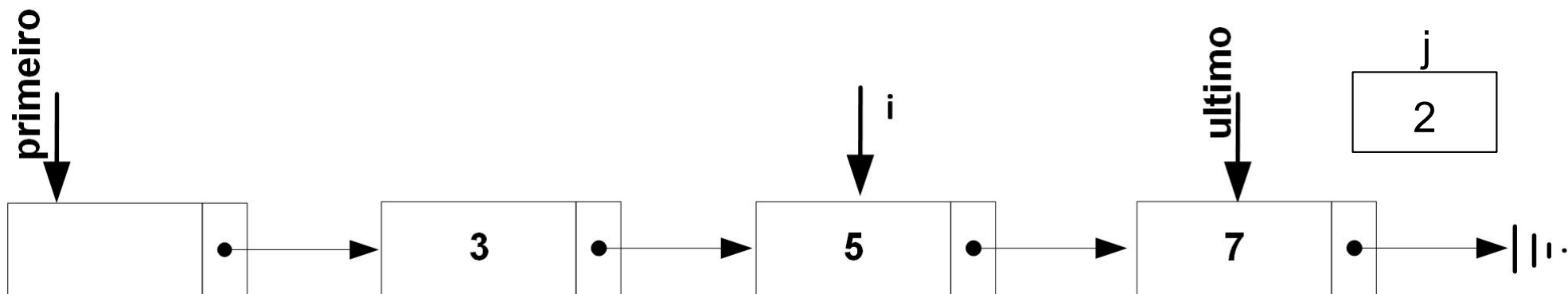
true



```

public void inserir(int x, int pos) throws Exception {    //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

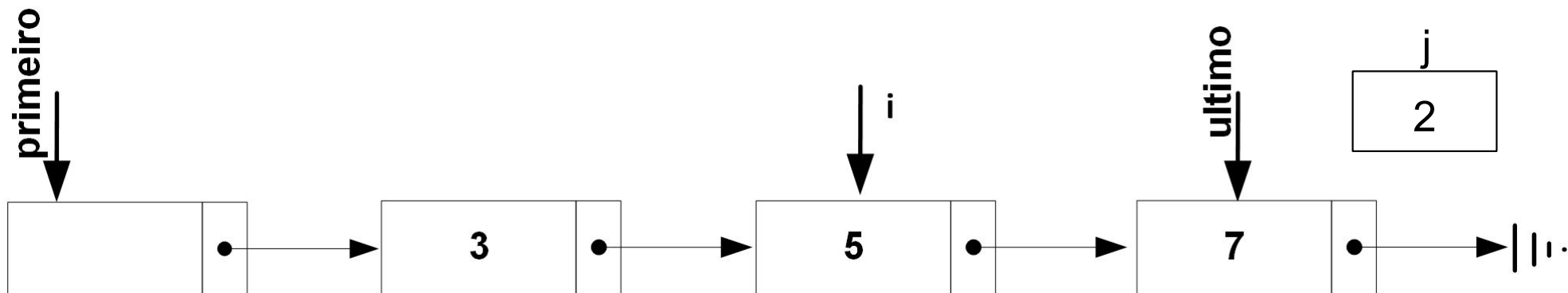


```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){      inserirInicio(x);
    } else if (pos == tamanho){  inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    }
}

```

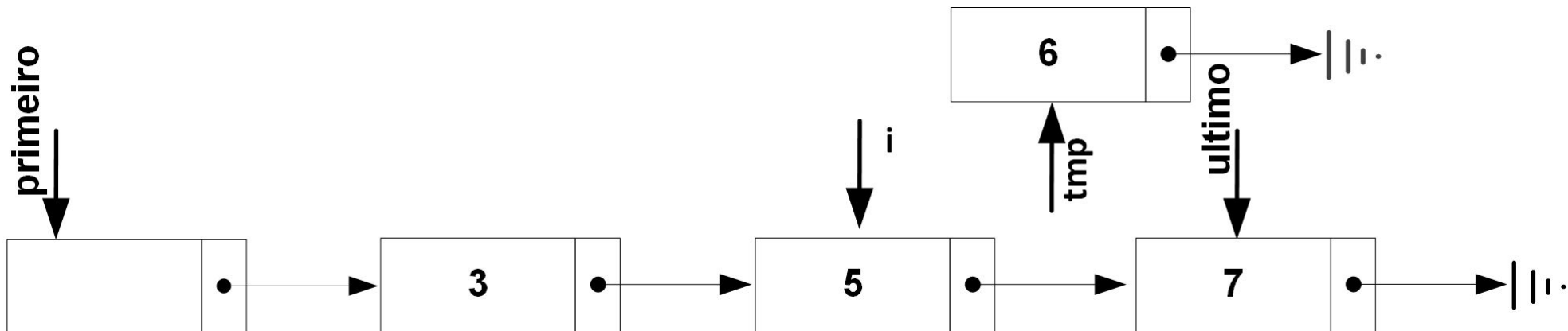
false



```

public void inserir(int x, int pos) throws Exception {    //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

```

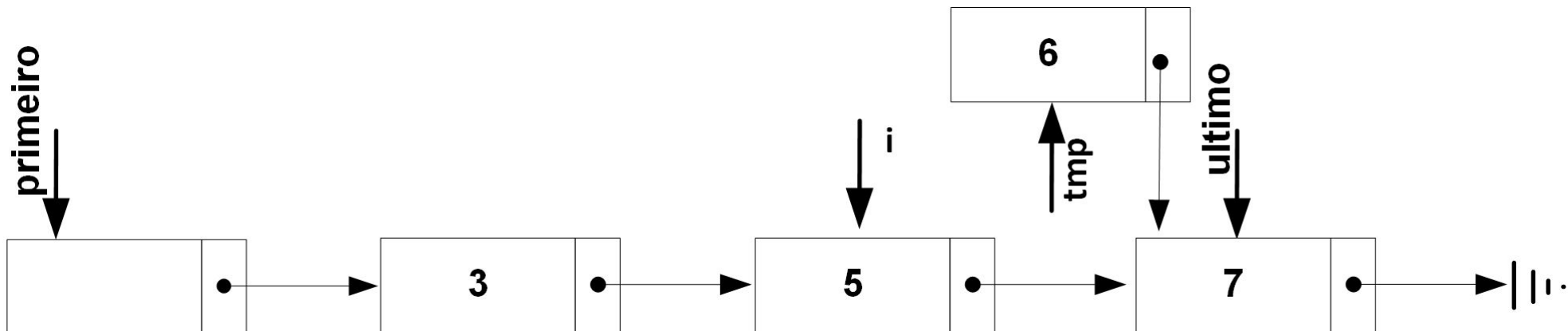




```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){      inserirInicio(x);
    } else if (pos == tamanho){  inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

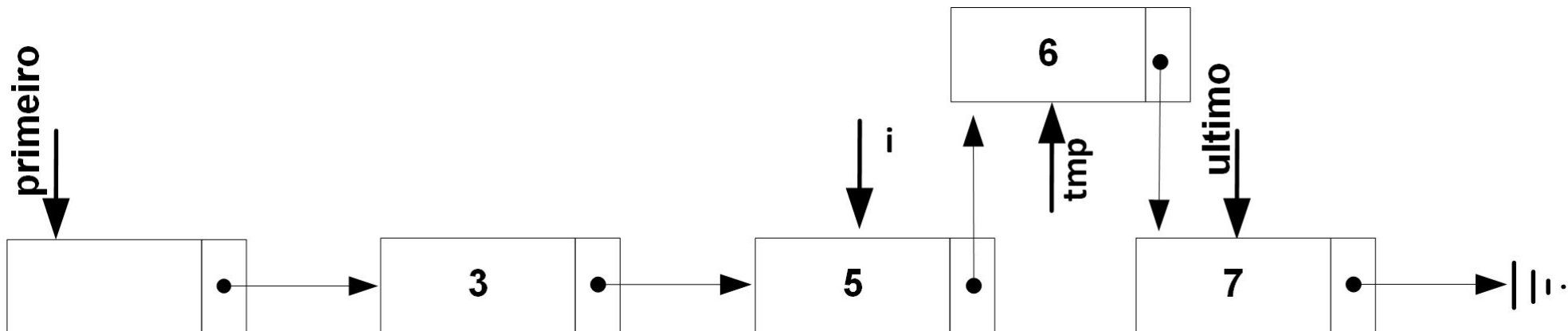
```



```

public void inserir(int x, int pos) throws Exception {    //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){          inserirInicio(x);
    } else if (pos == tamanho){    inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

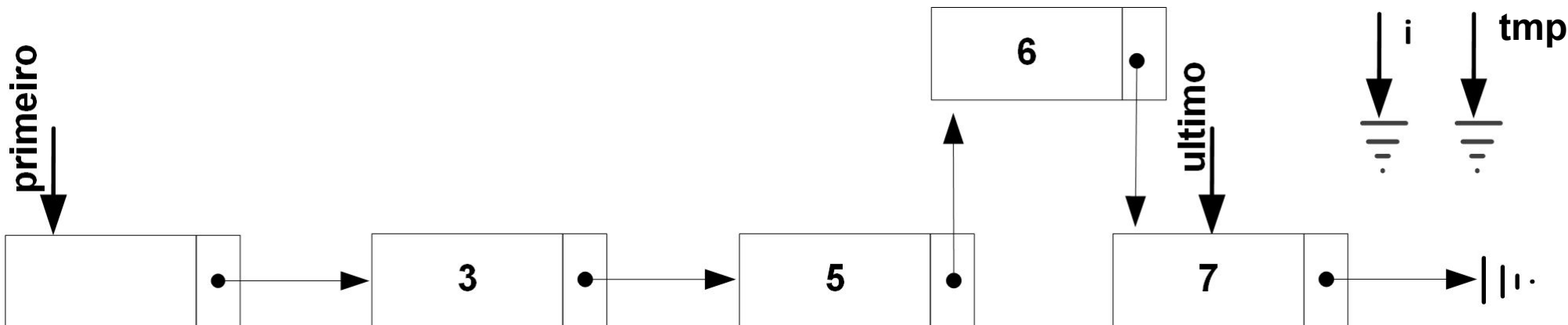
```



```

public void inserir(int x, int pos) throws Exception {           //Inserir(6, 2)
    int tamanho = tamanho();
    if (pos < 0 || pos > tamanho){ throw new Exception("Erro!");
    } else if (pos == 0){      inserirInicio(x);
    } else if (pos == tamanho){  inserirFim(x);
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = new Celula(x);
        tmp.prox = i.prox;
        i.prox = tmp;
        tmp = i = null;
    } }

```



# Remover

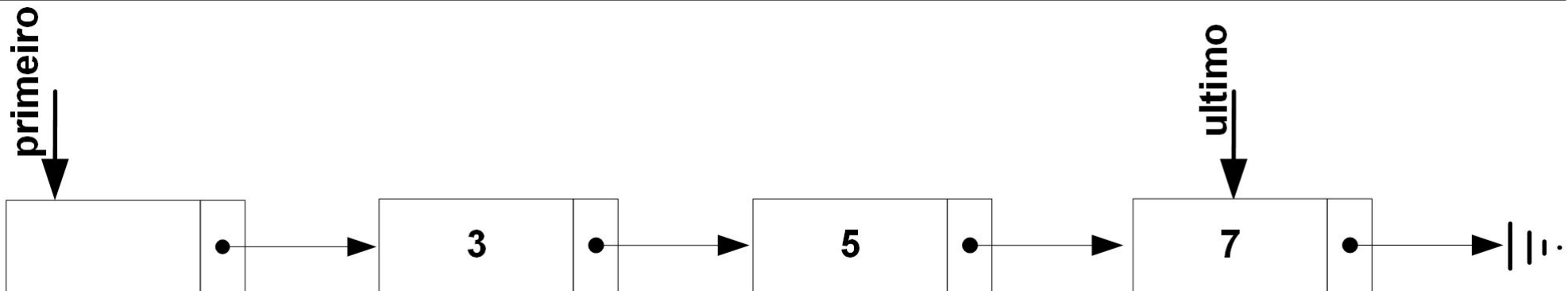
```
class Lista {  
    ...  
    public void inserirInicio(int x) { ... }  
    public int removerFim() { ... }  
    public void inserir(int x, int pos) { ... }  
    public int remover(int pos) { ... }  
}
```

## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;    i.prox = tmp.prox;
        tmp.prox = null;           i = tmp = null;
    }
    return elemento;
}

```



## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {

```

```

    Celula i = primeiro;

```

```

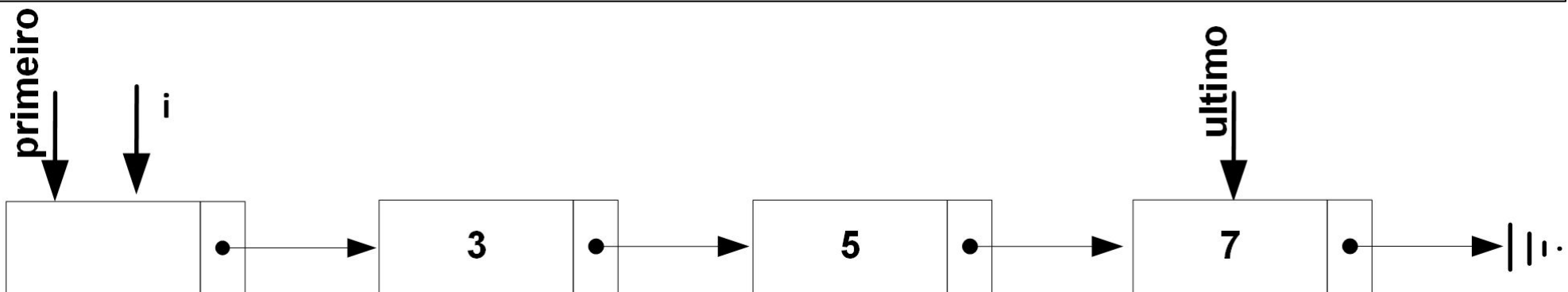
    for(int j = 0; j < pos; j++, i = i.prox);
    Celula tmp = i.prox;
    elemento = tmp.elemento;    i.prox = tmp.prox;
    tmp.prox = null;           i = tmp = null;
    }
    return elemento;

```

```

}

```

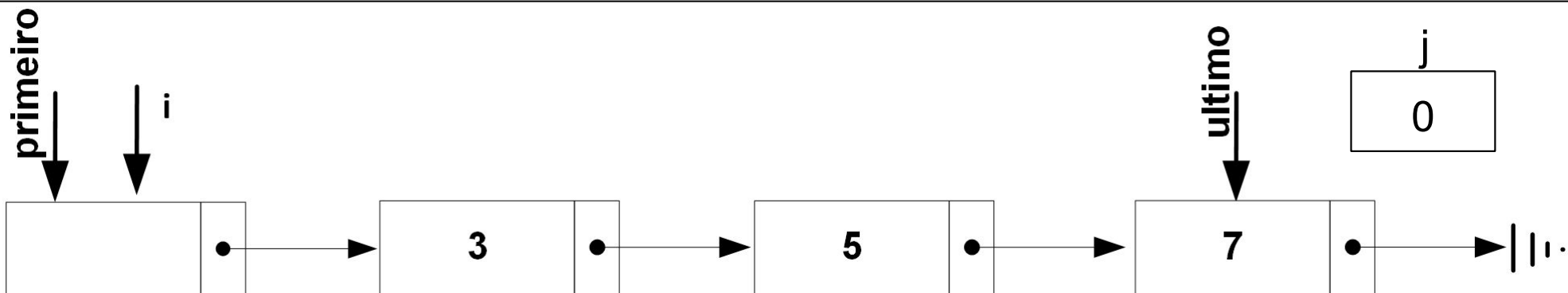


## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;      i.prox = tmp.prox;
        tmp.prox = null;              i = tmp = null;
    }
    return elemento;
}

```

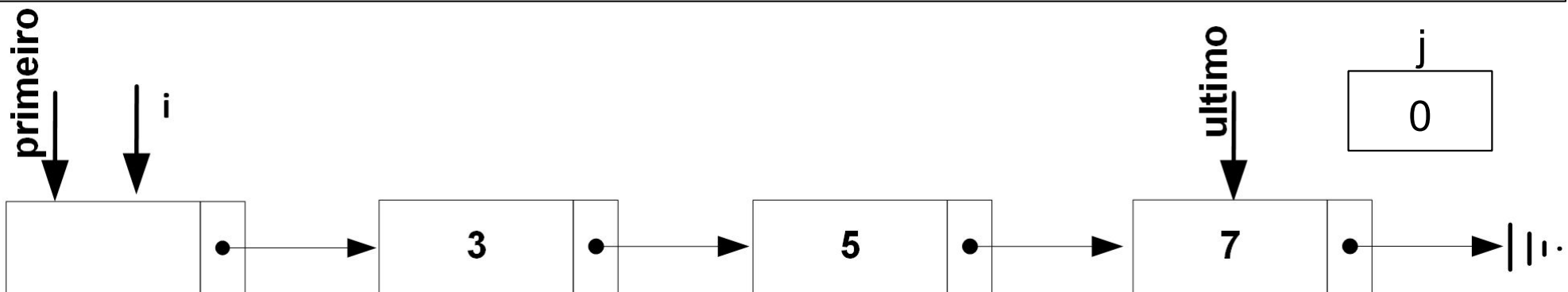


## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;      i.prox = tmp.prox;
        tmp.prox = null;              i = tmp = null;
    }
    return elemento;
}

```



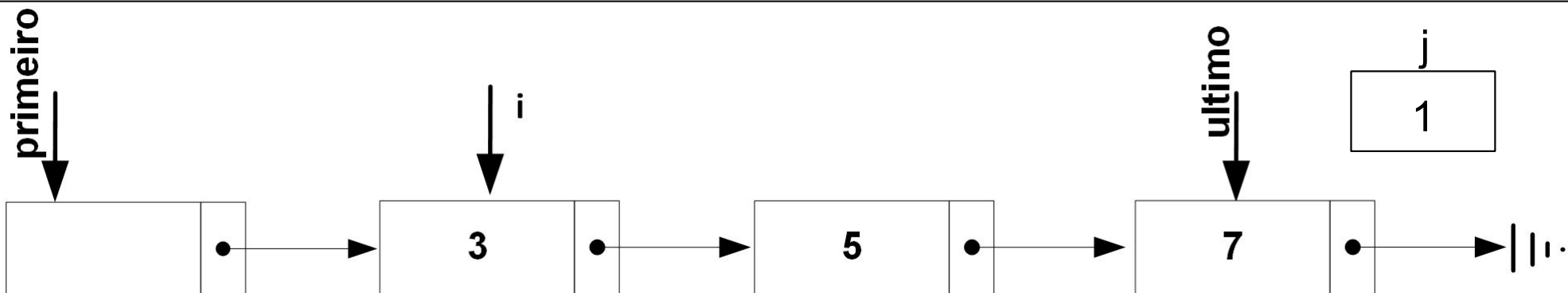


## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;    i.prox = tmp.prox;
        tmp.prox = null;           i = tmp = null;
    }
    return elemento;
}

```



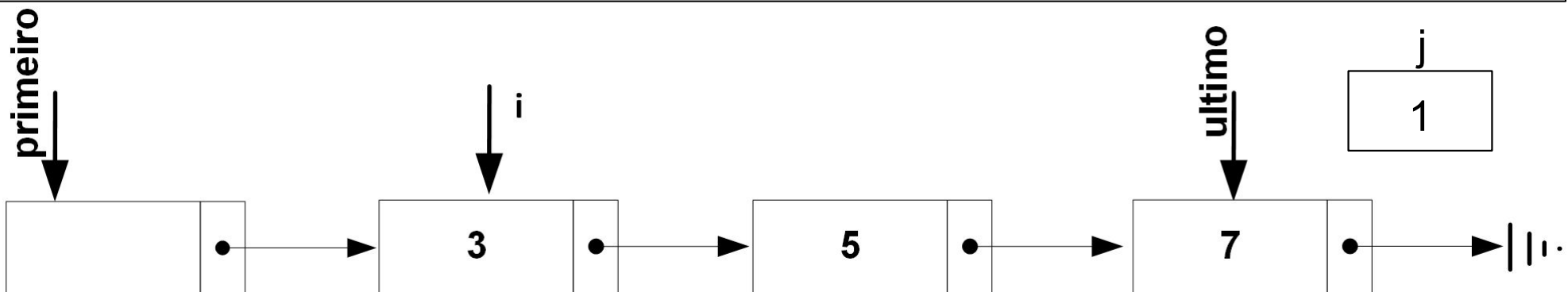
## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;    i.prox = tmp.prox;
        tmp.prox = null;            i = tmp = null;
    }
    return elemento;
}

```

false

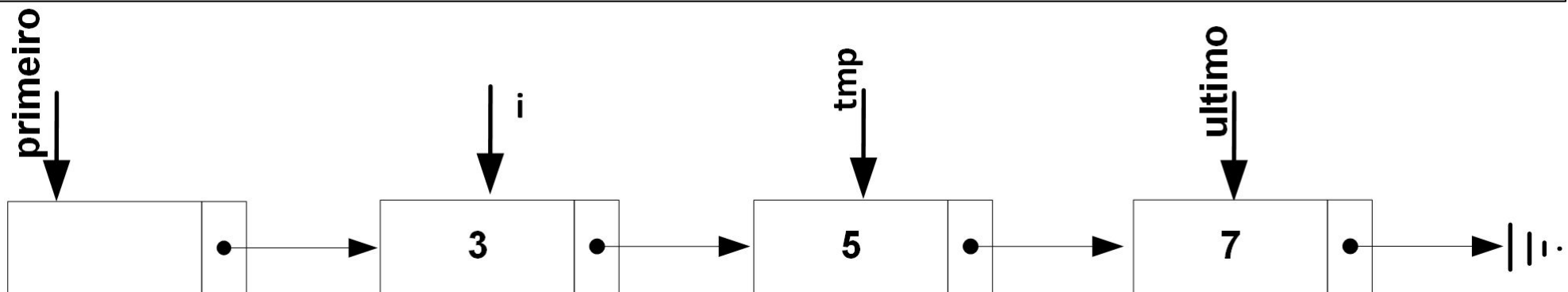


## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;    i.prox = tmp.prox;
        tmp.prox = null;           i = tmp = null;
    }
    return elemento;
}

```



## Remover

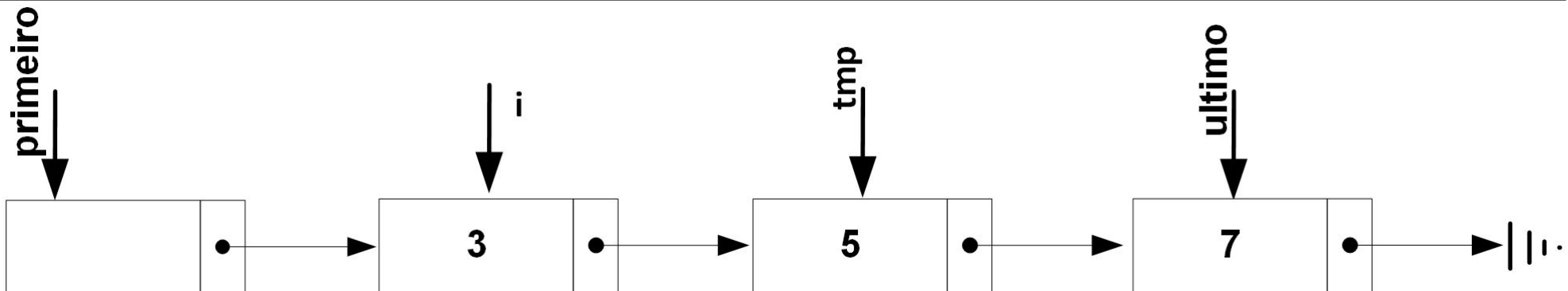
```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;      i.prox = tmp.prox;
        tmp.prox = null;              i = tmp = null;
    }
    return elemento;
}

```

elemento

5



## Remover

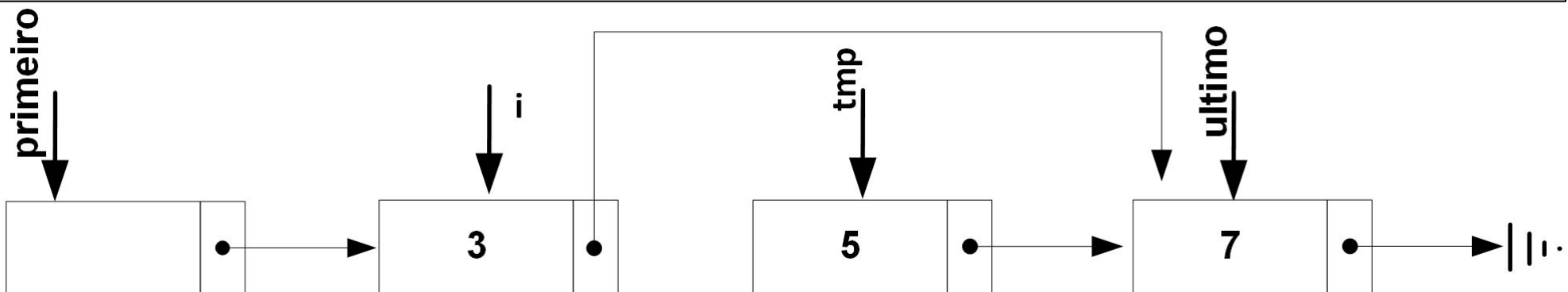
```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento; i.prox = tmp.prox;
        tmp.prox = null; i = tmp = null;
    }
    return elemento;
}

```

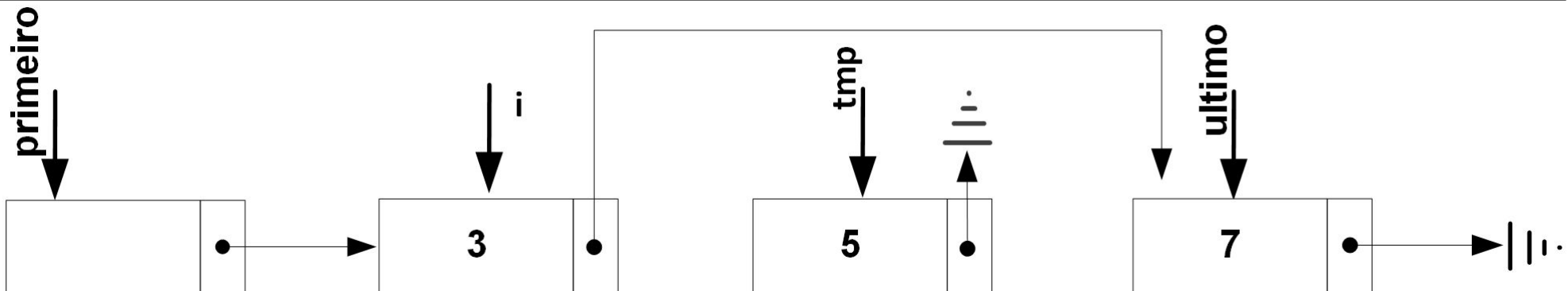
elemento

5
---



elemento

5



## Remover

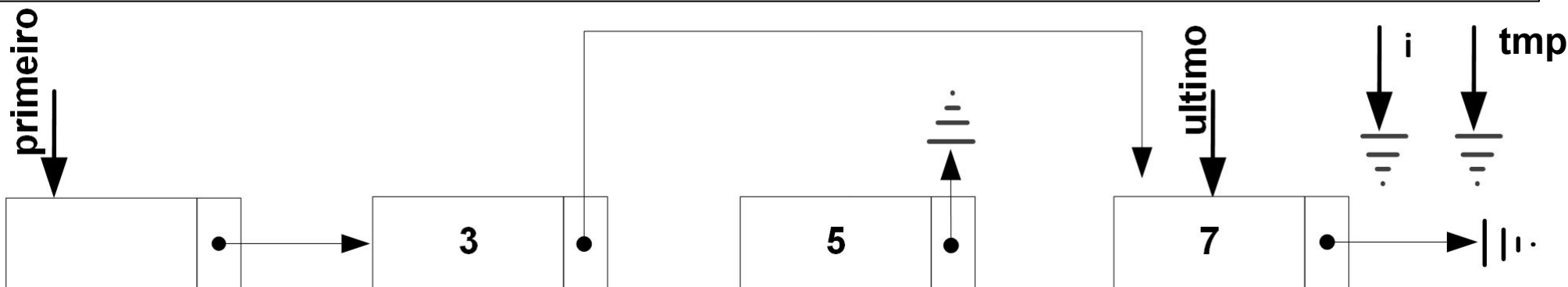
```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;    i.prox = tmp.prox;
        tmp.prox = null;            i = tmp = null;
    }
    return elemento;
}

```

elemento

5

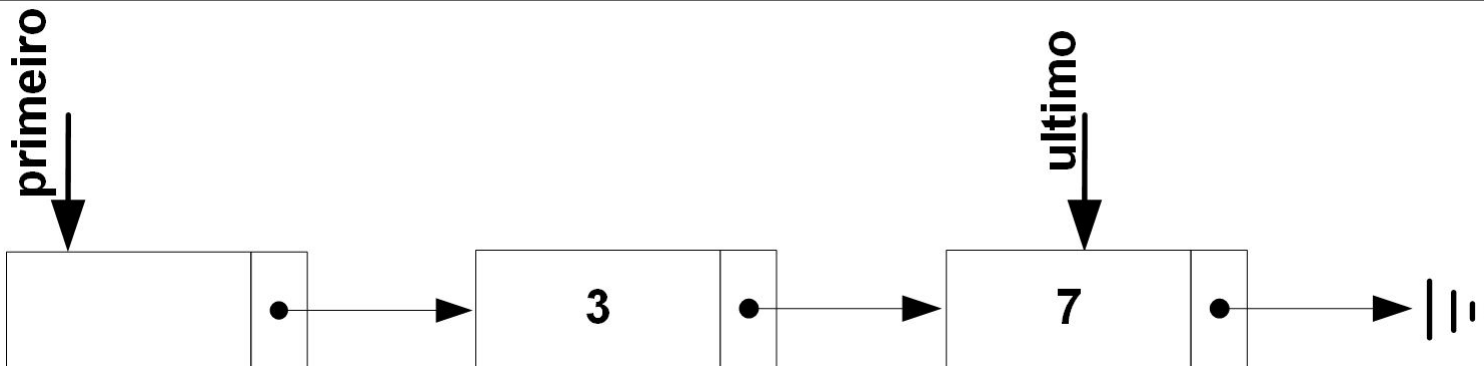
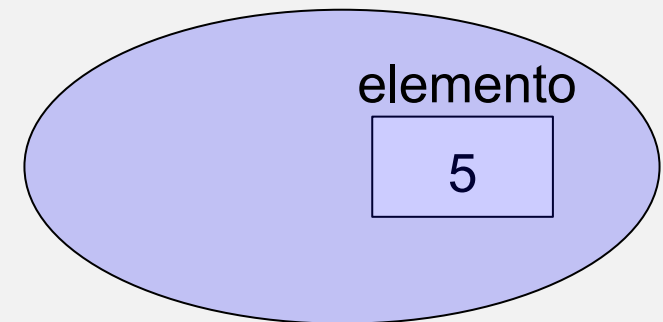


## Remover

```

public int remover(int pos ) throws Exception {           //remover(1)
    int elemento, tamanho = tamanho();
    if (primeiro == ultimo || pos < 0 || pos >= tamanho){ throw new Exception("Erro!");
    } else if (pos == 0) {                                elemento = removerInicio();
    } else if (pos == tamanho - 1){                       elemento = removerFim();
    } else {
        Celula i = primeiro;
        for(int j = 0; j < pos; j++, i = i.prox);
        Celula tmp = i.prox;
        elemento = tmp.elemento;      i.prox = tmp.prox;
        tmp.prox = null;              i = tmp = null;
    }
    return elemento;
}

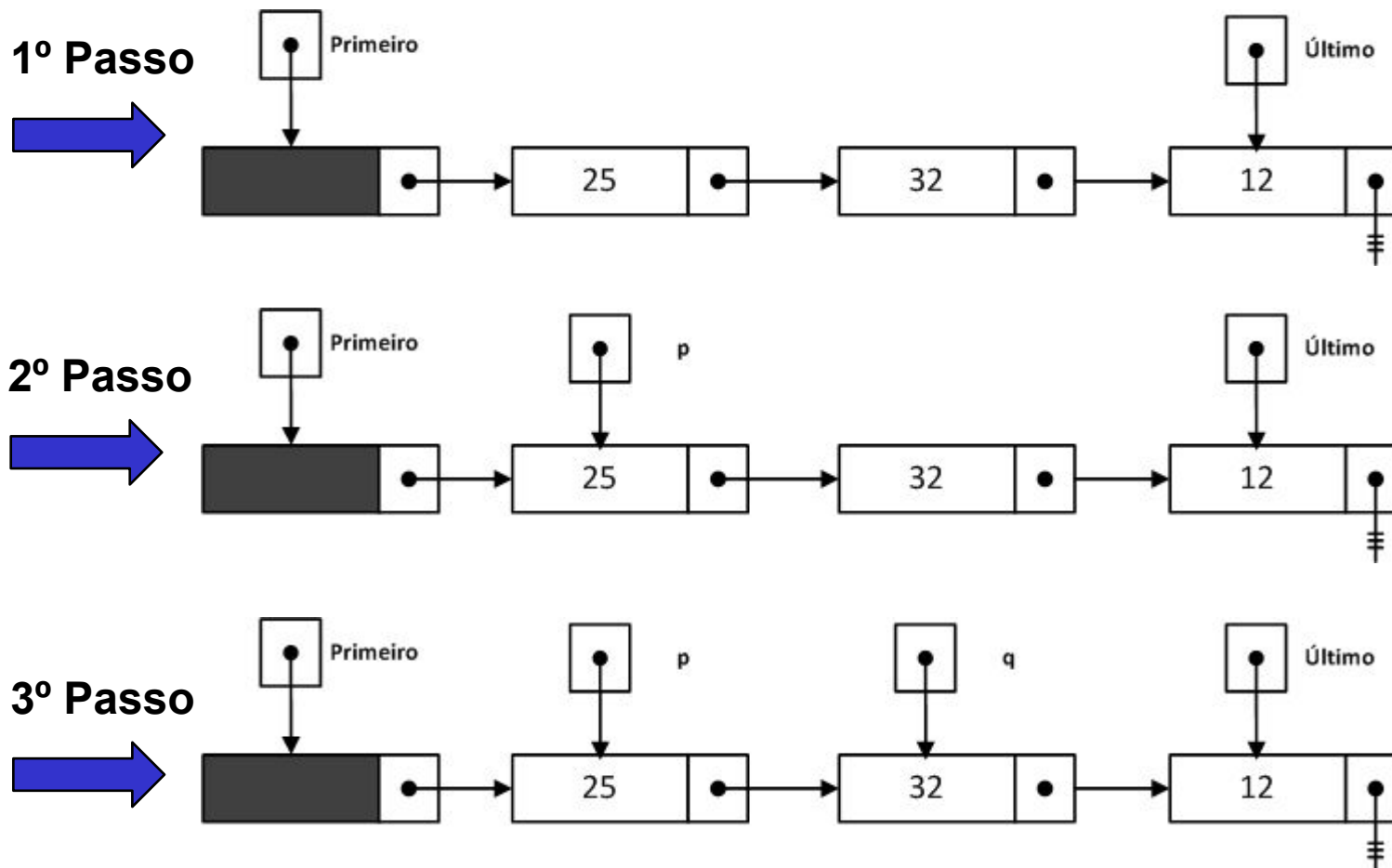
```





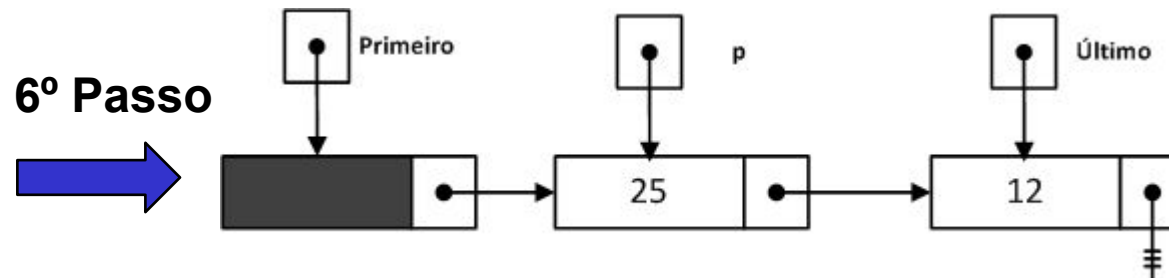
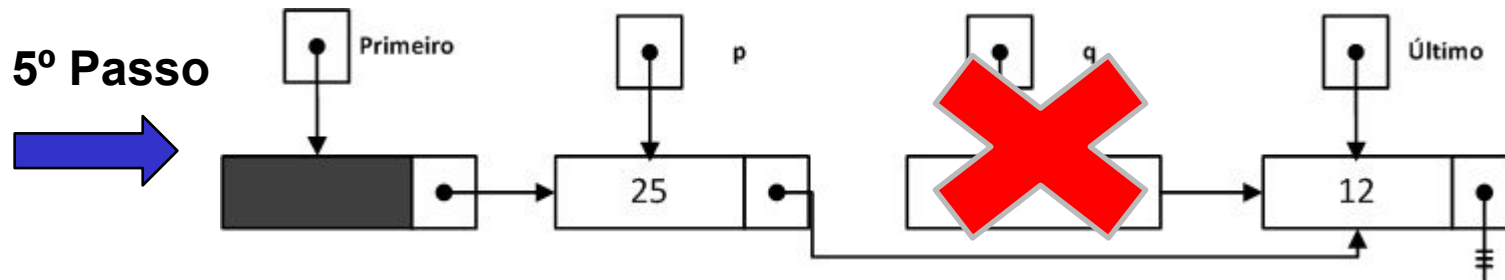
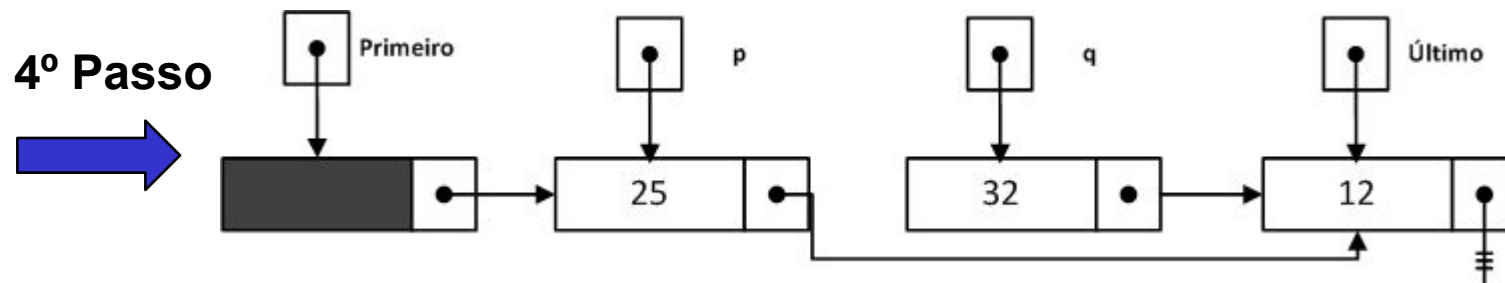
## Exercício

- Seja nossa classe Lista, implemente um método que remove a segunda posição válida. Siga os passos da figura abaixo



## Exercício

- Seja nossa classe Lista, implemente um método que remove a segunda posição válida. Siga os passos da figura abaixo



## Exercício

- Crie uma classe `ListaSimplesEncadeadaOrdenada`, garantindo que os elementos sempre fiquem ordenados.

# Exercício

- Na lista simples, crie um contador para a quantidade de elementos.

DICA: Nas inserções aumente o contador em uma unidade e nas remoções, decemente uma unidade.