

# Introdução a Estruturas de Dados: Lista

## AULA 10

Ricardo Massa F. Lima  
[rmfl@cin.ufpe.br](mailto:rmfl@cin.ufpe.br)

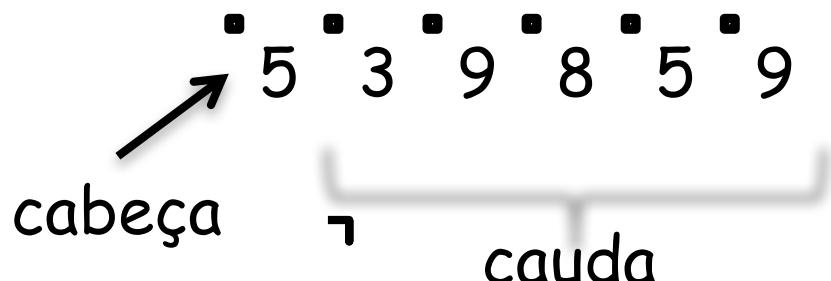
Sérgio C. B. Soares  
[scbs@cin.ufpe.br](mailto:scbs@cin.ufpe.br)



[CIn.ufpe.br](http://CIn.ufpe.br)

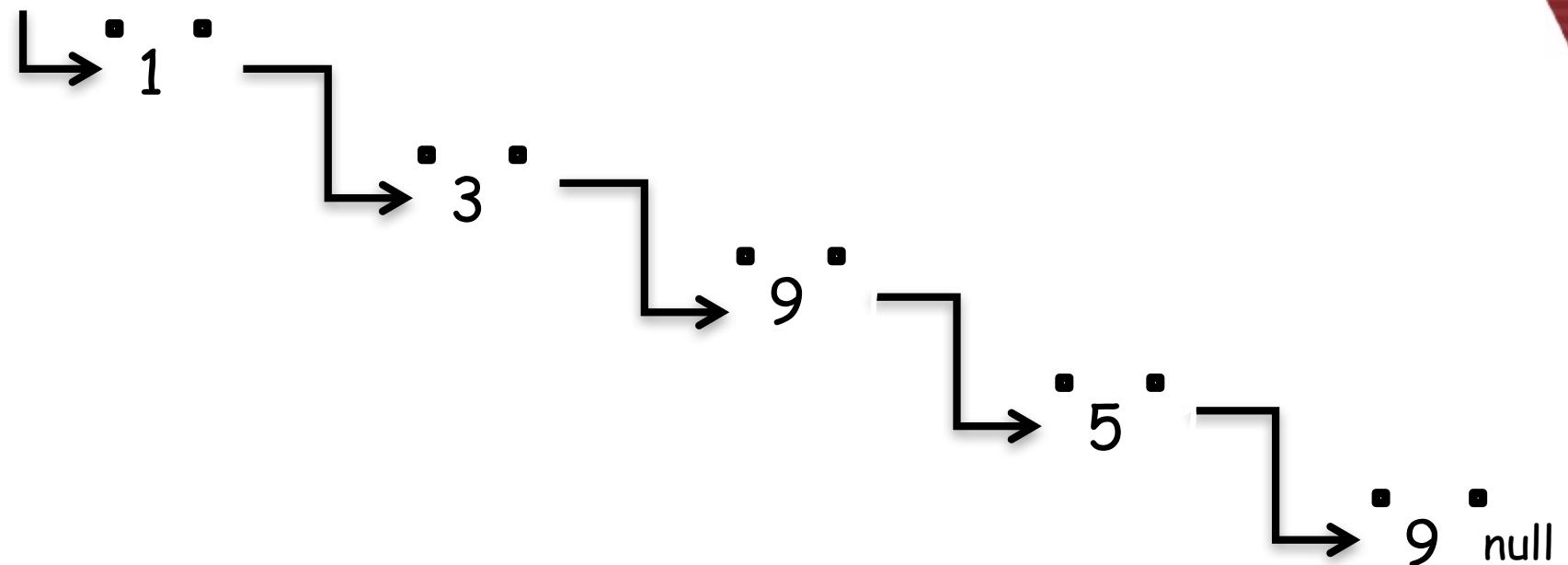
# Listas

- Uma lista (ou sequência) é uma estrutura na qual os elementos
  - seguem uma ordem
    - 1º elemento, 2º elemento, ... N-ésimo elemento
  - podem se repetir
- A lista cresce/diminui dinamicamente
- Ex.: Uma lista de números inteiros



# Listas de inteiros em Java

cabeça

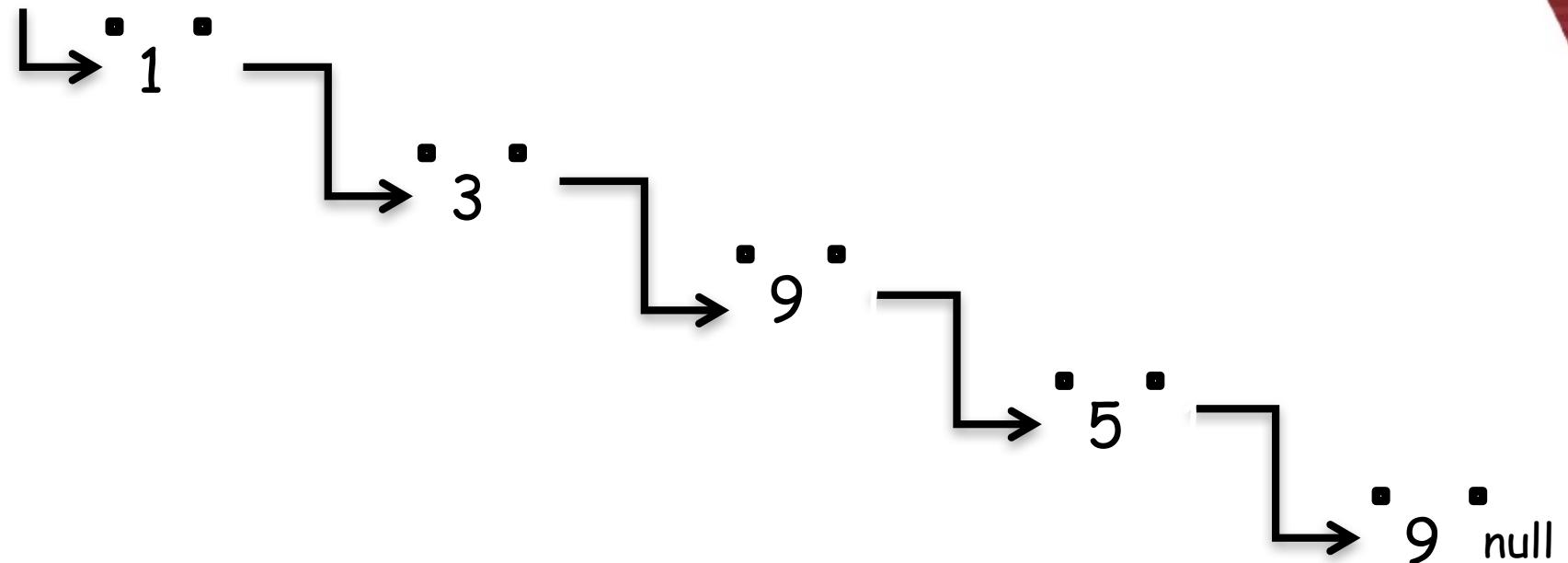


# Como implementar a lista de inteiros em Java?

- Criar uma classe . . .
- Ideias???

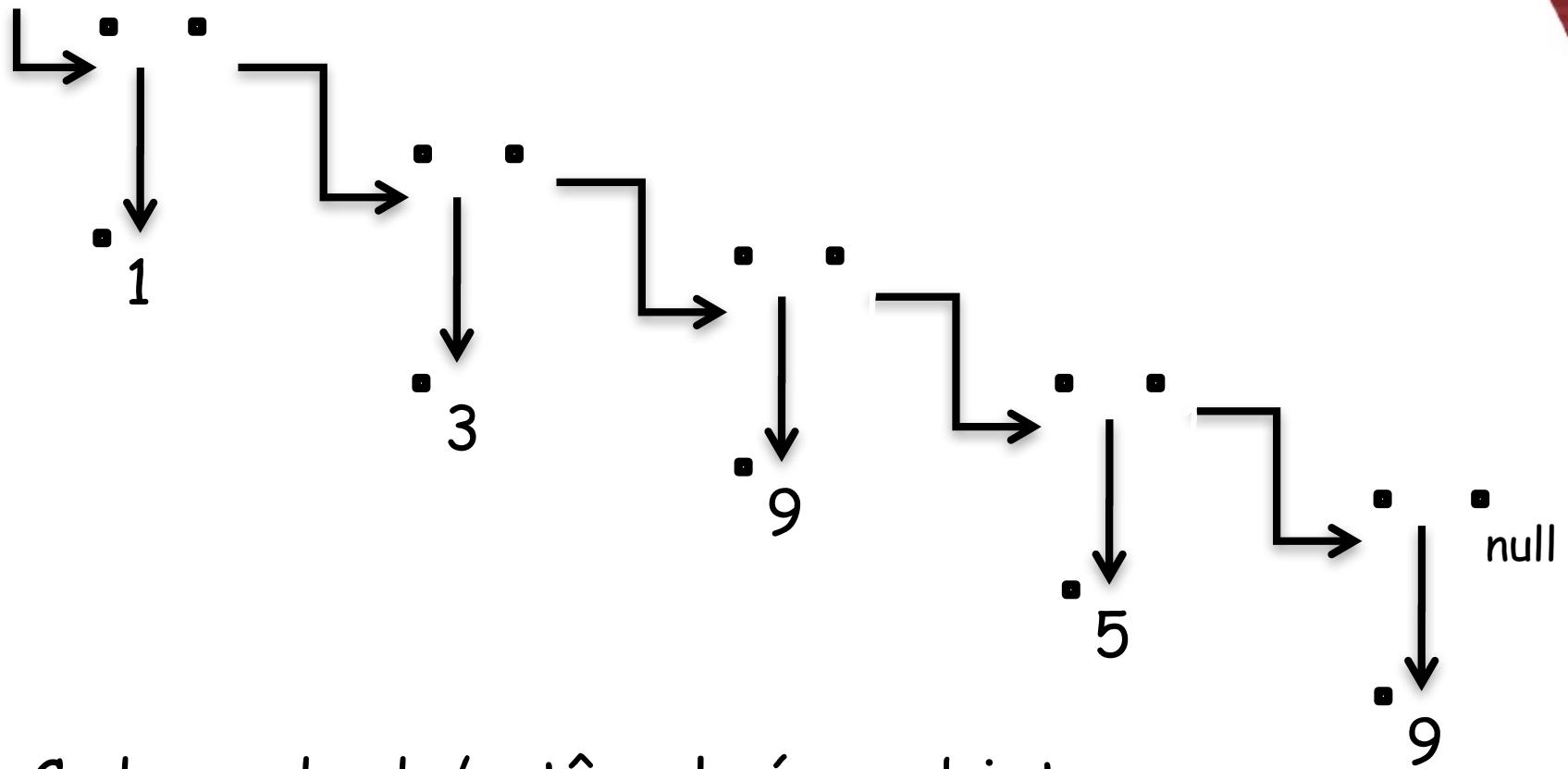
# Vamos fazer uma pequena mudança nesta representação!

cabeça



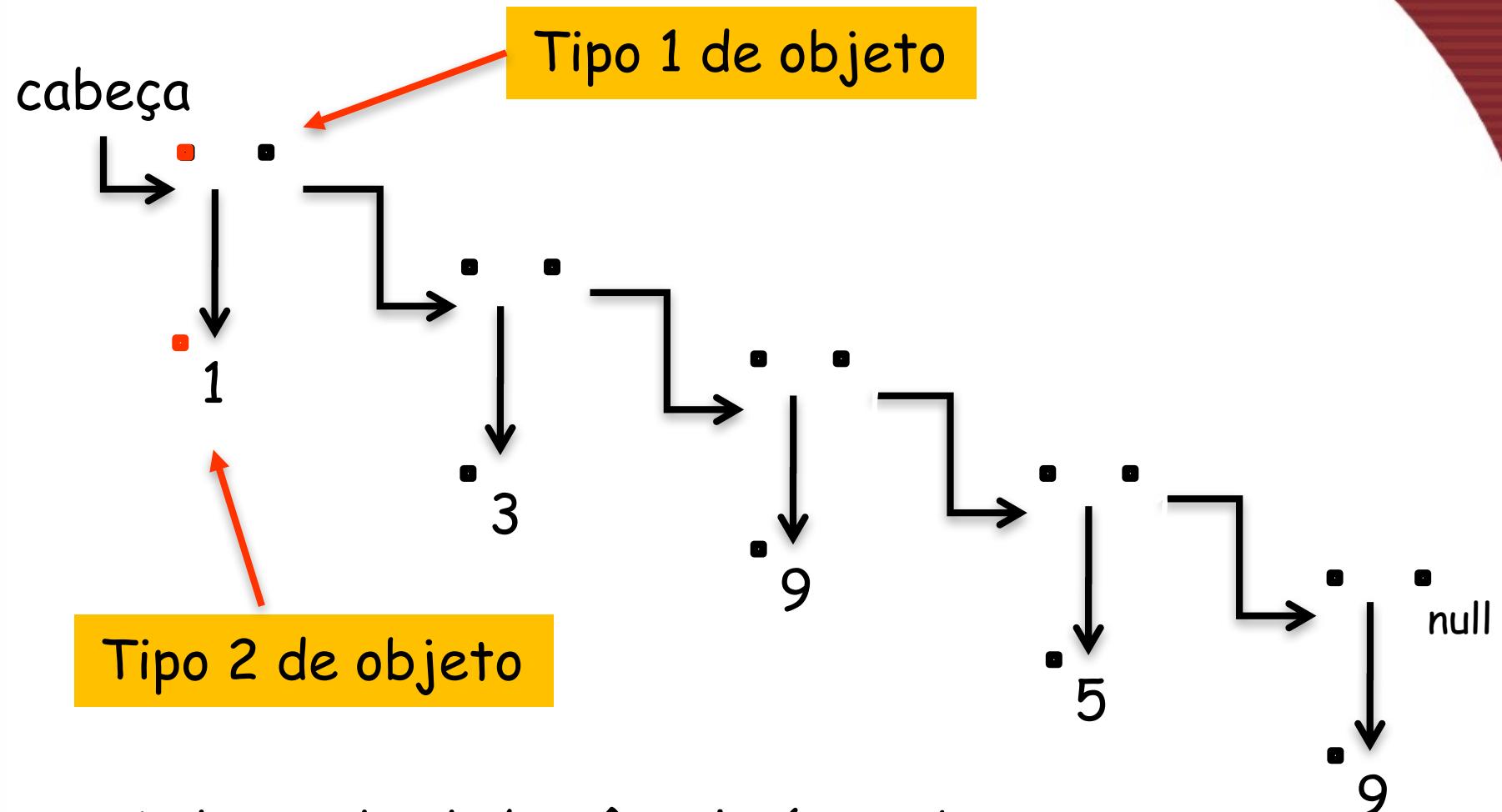
# Listas de inteiros em Java

cabeça



- Cada quadrado/retângulo é um objeto
- Para cada objeto deve haver uma classe

# Listas de inteiros em Java



- Cada quadrado/retângulo é um objeto
- Para cada objeto deve haver uma classe

# Listas de int (1/5)

```
public class Valor {  
    private int valor;  
  
    public Valor (int valor) {  
        this.valor = valor;  
    }  
  
    public int getValor() {  
        return this.valor;  
    }  
  
    public boolean equals(Valor valor) {  
        return valor.getValor() == this.valor;  
    }  
}
```

# Listas de int (2/5)

```
public class Lista {  
    private Valor valor;  
    private Lista proximo;  
    public Lista () {  
        . . .  
        . . .  
    }  
    public void inserir(Valor valor) { . . . }  
    public void remover(Valor valor) { . . . }  
}
```

# Listas de int (3/5)

```
public class Lista {  
    private Valor valor;  
    private Lista proximo;  
    public Lista () {  
        this.valor = null;  
        this.proximo = null;  
    }  
    public void inserir(Valor valor) { . . . }  
    public void remover(Valor valor) { . . . }  
}
```

# Listas de int (4/5)

```
public void inserir(Valor valor) {  
    if (this.valor == null) {  
        this.valor = valor;  
        this.proximo = new Lista();  
    } else {  
        this.proximo.inserir(valor);  
    }  
}
```

# Listas de int (5/5)

```
public void remover(Valor valor) {  
    if (this.valor != null) {  
        if (this.valor.equals(valor)) {  
            this.valor = this.proximo.valor;  
            this.proximo = this.proximo.proximo;  
        } else {  
            this.proximo.remover(valor);  
        }  
    } else {  
        throw new RuntimeException("Erro");  
    }  
}
```

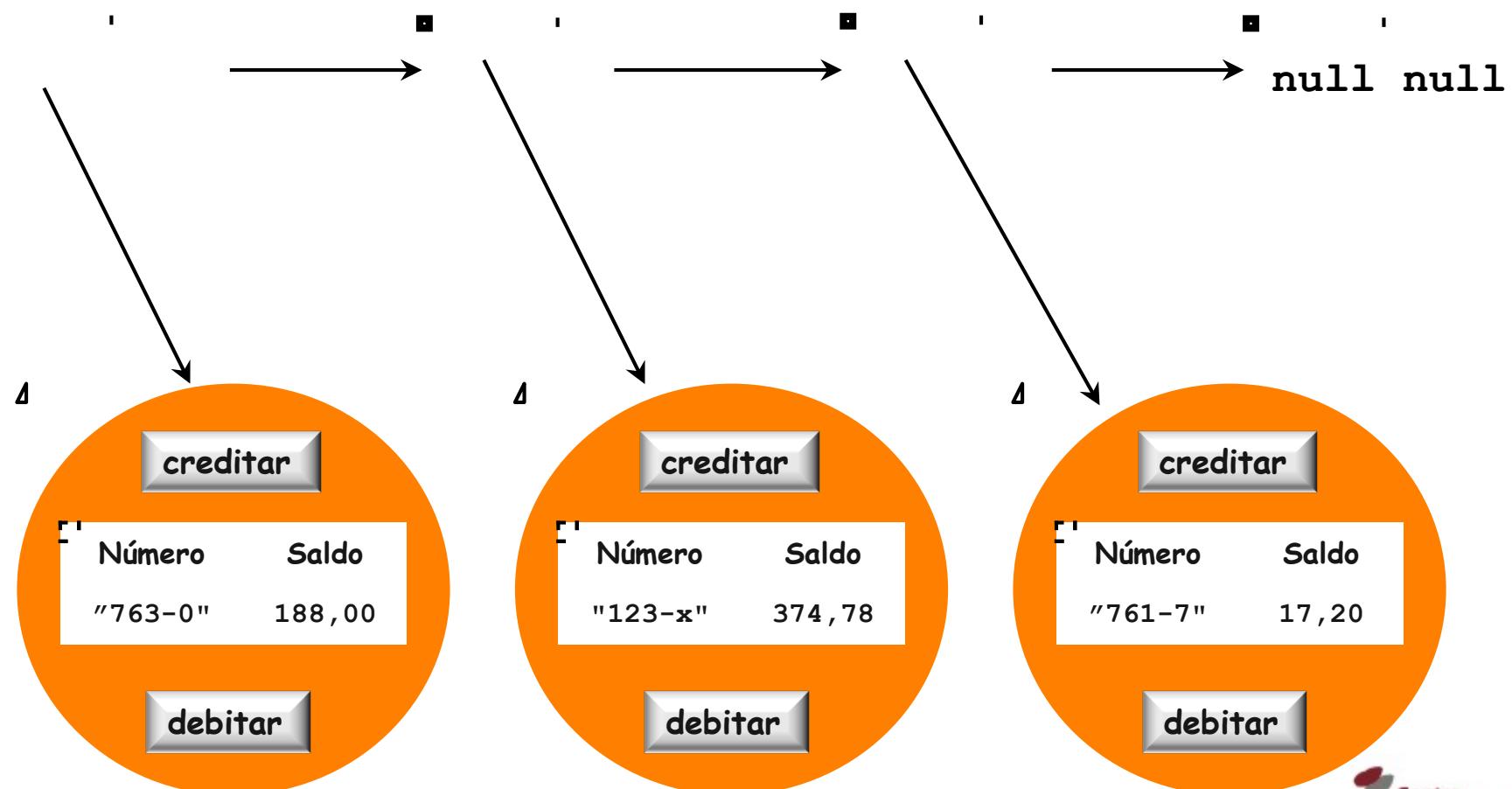


Atenção: Por enquanto, vamos indicar erros assim. Mas isso vai mudar

# **Exercício (1)**

- Crie a classe `ListaInt` no pacote  
`aulaEstruturaDados.br.ufpe.cin.listaint`  
conforme apresentado em sala
- Adicionalmente crie um método `toString` que  
retorna os elementos da lista em um `String`.
- Crie uma classe `Programa` no mesmo pacote para  
testar a classe criada e imprimir a mesma usando o  
método `toString`
- Teste o programa

# Listas de Contas



# Listas de Contas: Descrição

```
public class ListaContas {  
    private Conta conta;  
    private ListaContas prox;  
  
    public ListaContas() {  
        this.conta = null;  
        this.prox = null;  
    }  
}
```

# Listas de Contas: Descrição

```
public void inserir (Conta conta) {  
    if (this.conta == null) {  
        this.conta = conta;  
        this.prox = new ListaContas();  
    } else {  
        this.prox.inserir(conta);  
    }  
}
```

# Listas de Contas: Descrição

```
public void remover(Conta c) {  
    if (this.conta != null) {  
        if (this.conta.equals(c)) {  
            this.conta = this.prox.conta;  
            this.prox = this.prox.prox;  
        } else {  
            this.prox.remover(c);  
        }  
    } else {  
        throw new RuntimeException("Erro");  
    }  
}
```

# Listas de Contas: Descrição

```
public Conta procurar (String numero) {  
    Conta resposta = null;  
    if (this.conta != null) {  
        if (this.conta.getNumero () .equals (numero)) {  
            resposta = this.conta;  
        } else {  
            resposta = this.prox.procurar (numero);  
        }  
    } else {  
        throw new RuntimeException ("Erro");  
    }  
    return resposta;  
}
```

## Exercício (2)

- Utilize a solução dos exercícios da última aula  
[www.cin.ufpe.br/~if669/material/solucoes/aula15.zip](http://www.cin.ufpe.br/~if669/material/solucoes/aula15.zip)
- Crie a classe `RepositorioContasLista` que implementa a interface `RepositorioContas` utilizando uma lista de contas conforme apresentado em sala
- Altere a classe `Programa` para utilizar a classe `RepositorioContasLista` no teste
- Teste o programa