

CMP1054 – ED I

Pilhas encadeadas

Pilhas usando estruturas auto referenciadas

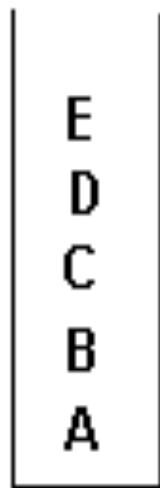
Java

Prof. Dr. José Olímpio Ferreira

Pilha

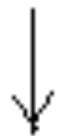
- É uma lista linear em que todas as inserções e retiradas (todos os acessos) são feitos em apenas um extremo da lista.
- Os itens são colocados um sobre o outro. O item inserido mais recentemente está no topo e o inserido menos recentemente no fundo.
- O modelo intuitivo é o de um monte de pratos em uma prateleira, sendo conveniente retirar ou adicionar pratos na parte superior.
- Esta imagem está freqüentemente associada com a teoria de autômato, na qual o topo de uma pilha é considerado como o receptáculo de uma cabeça de leitura/gravação que pode empilhar e desempilhar itens da pilha.

Pilha



Stack

Front

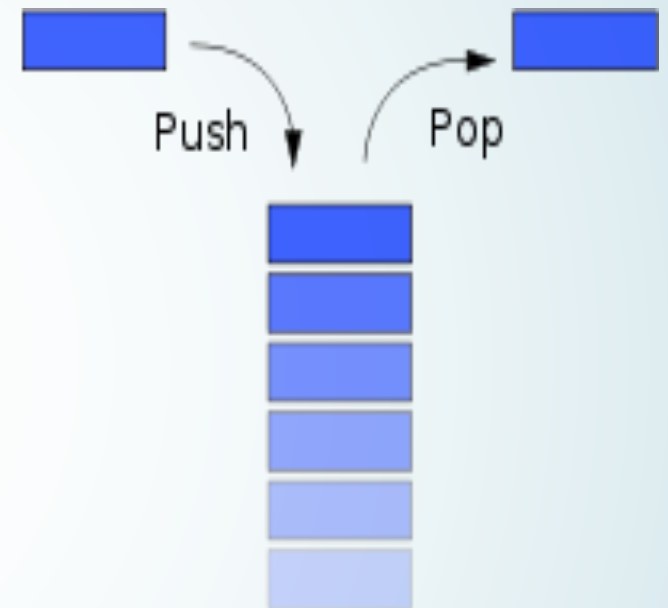


Rear



A B C D E

Queue



Propriedade e Aplicações das Pilhas

- Propriedade: o último item inserido é o primeiro item que pode ser retirado da pilha.
- São chamadas listas com disciplina lifo (“last-in, first-out”).
- Existe uma ordem linear para as pilhas, do “mais recente para o menos recente”.
- Pode-se pensar uma pilha com uma seqüência de obrigações adiadas.
- Onde a ordem de remoção garante que as estruturas mais internas serão processadas antes das mais externas.

Propriedade e Aplicações das Pilhas

- É ideal para processamento de estruturas aninhadas de profundidade imprevisível.
- Aplicações em estruturas aninhadas:
 - Quando é necessário caminhar em um conjunto de dados e guardar uma lista de coisas a fazer posteriormente.
 - O controle de seqüências de chamadas de sub-programas.
 - A sintaxe de expressões aritméticas.
- As pilhas ocorrem em estruturas de natureza recursiva (como árvores). Elas são utilizadas para implementar a recursividade.

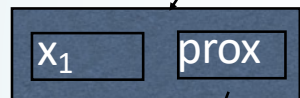
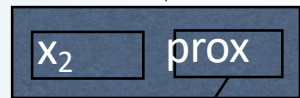
TAD Pilhas

- Conjunto de operações:
 - Cria uma pilha Vazia.
 - Verifica se a pilha está vazia.
 - Verifica se a pilha está cheia.
 - Empilhar o item x no topo da pilha.
 - Desempilhar o item x do topo da pilha, retirando-o da pilha.
- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas.
- As duas representações mais utilizadas são as implementações por meio de arranjos e de estruturas auto-referenciadas (listas ligadas).

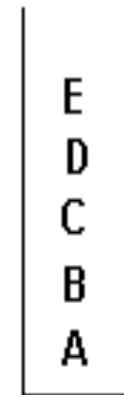
Pilhas usando Estruturas Auto-Referenciadas

- Para desempilhar um item, basta desligar a célula que contém x_n e a célula que contém x_{n-1} passa a ser a célula de topo.
- Para empilhar um novo item, basta fazer a operação contrária, criando uma nova célula para receber o novo item, colocando-a no topo.

topo



null



Stack

← Top

Front



Rear



A B C D E

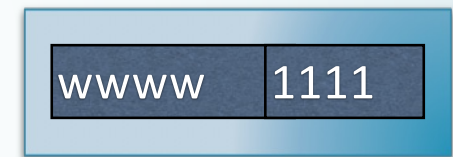
Queue

Pilhas usando Estruturas Auto-Referenciadas

- ➡ Pode existir um atributo tamanho para registrar o número de células (nós).
- ➡ Cada célula (nó) de uma pilha contém um item da pilha e uma referência para outra célula (nó).
- ➡ A classe Pilha contém uma referência para a célula (nó) no topo da pilha.

Objetos do tipo Item

- Atributos:
 - string nome
 - int numero
- Métodos:
 - construtor
 - getNome, getNumero
 - setNome, setNumero



Objetos do tipo No (Célula)

➤ Atributos:

➤ Item it

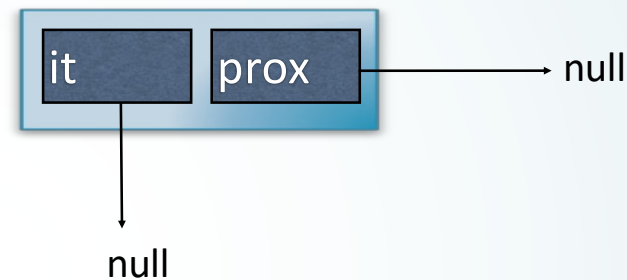
➤ No prox

➤ Métodos

➤ construtor

➤ setItem e setProx

➤ getItem e getProx



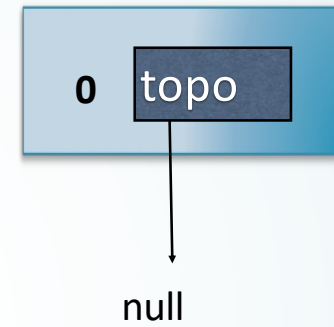
Objetos do tipo Pilha

► Atributos:

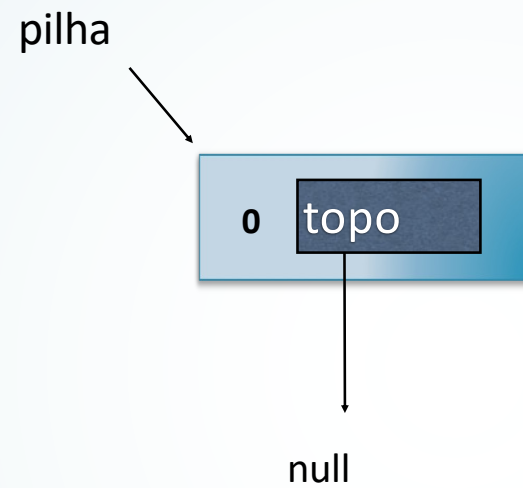
- No topo;
- Int qde;

► Métodos:

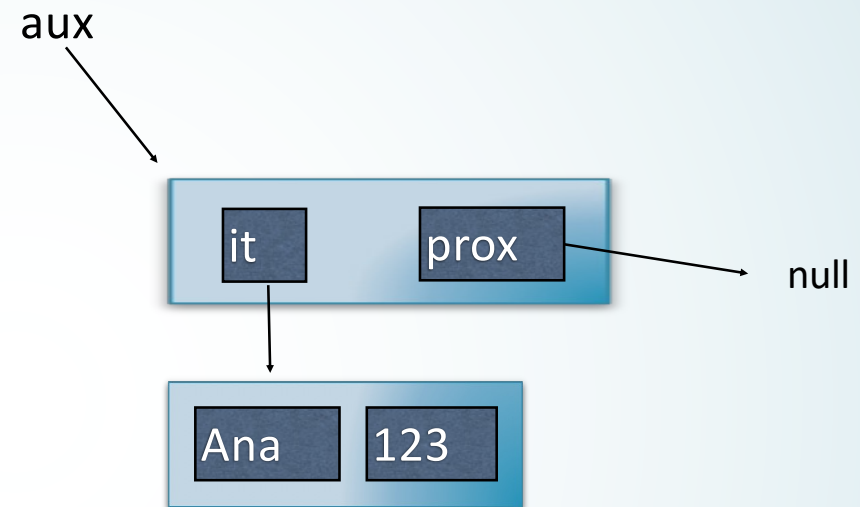
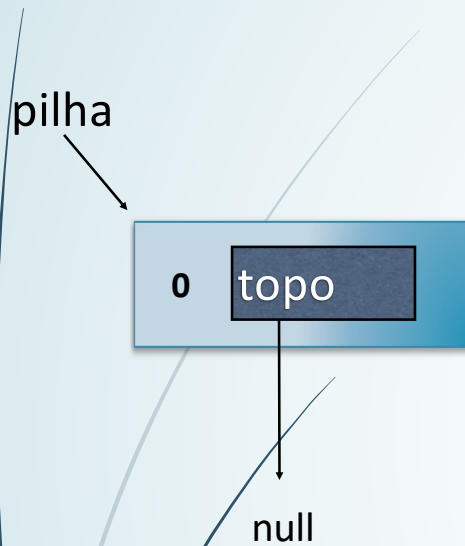
- construtor
- getQde()
- empilhar (push)
- desempilhar (pop)
- pilhaVazia()



Pilha(): Constrói pilha vazia

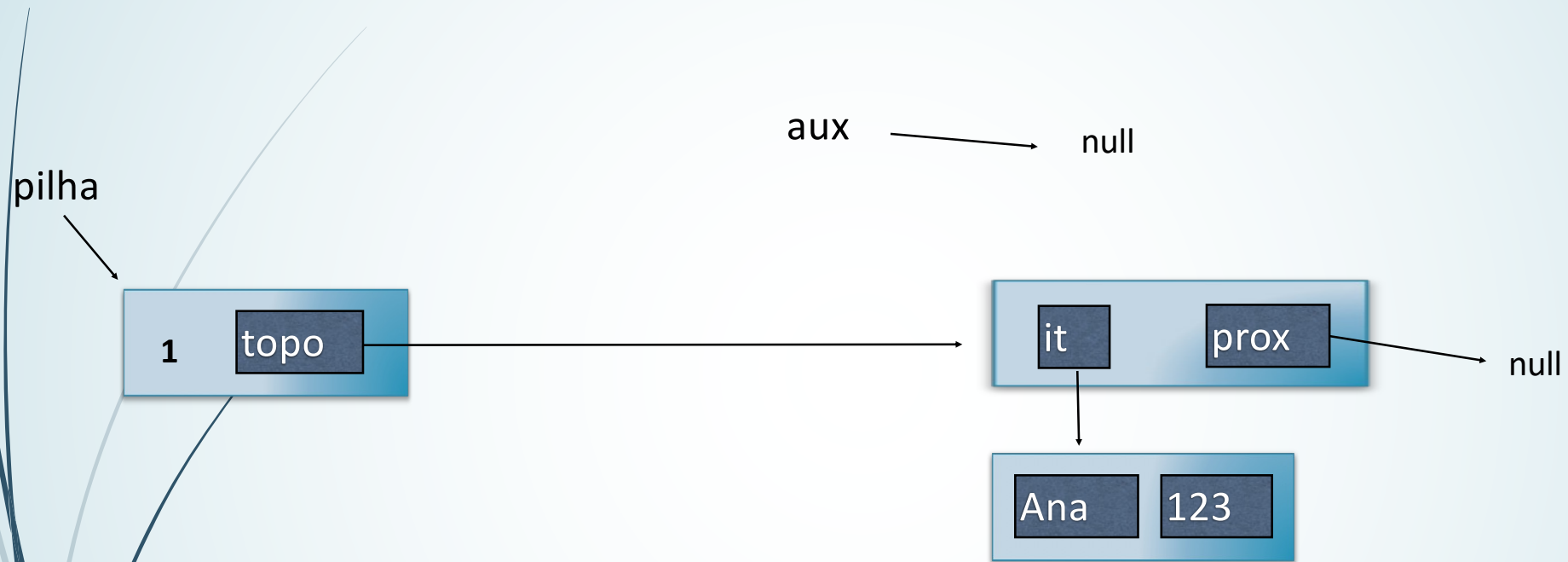


push(Item x)



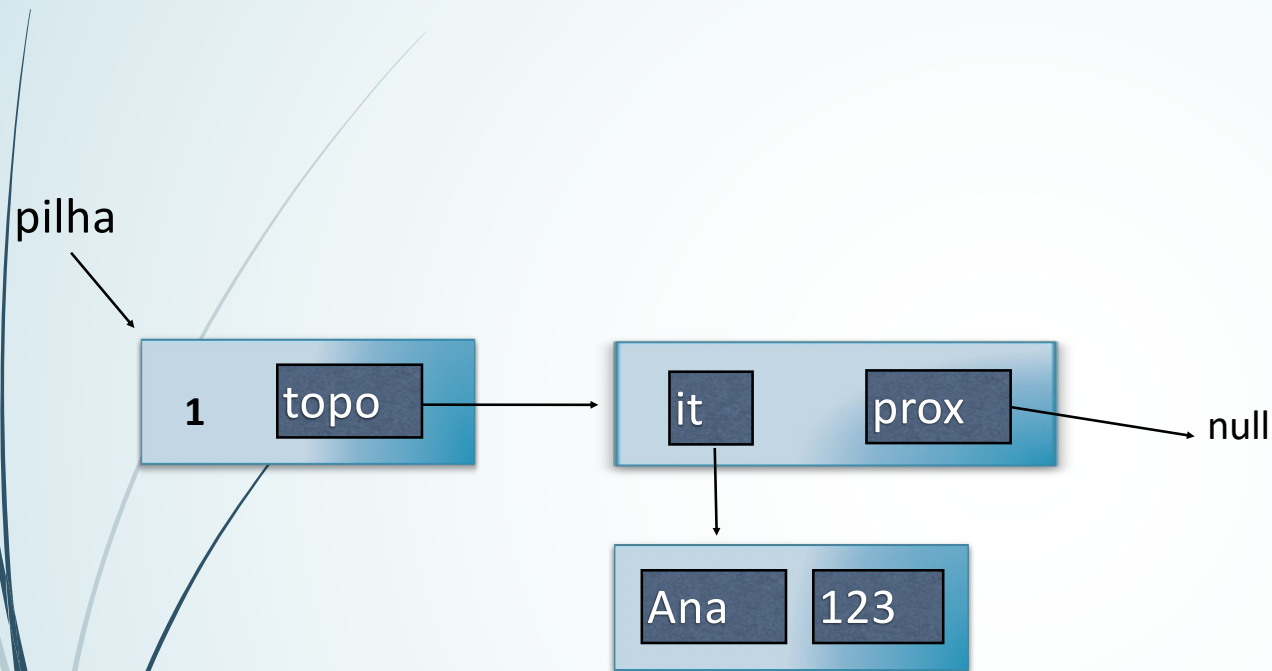
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



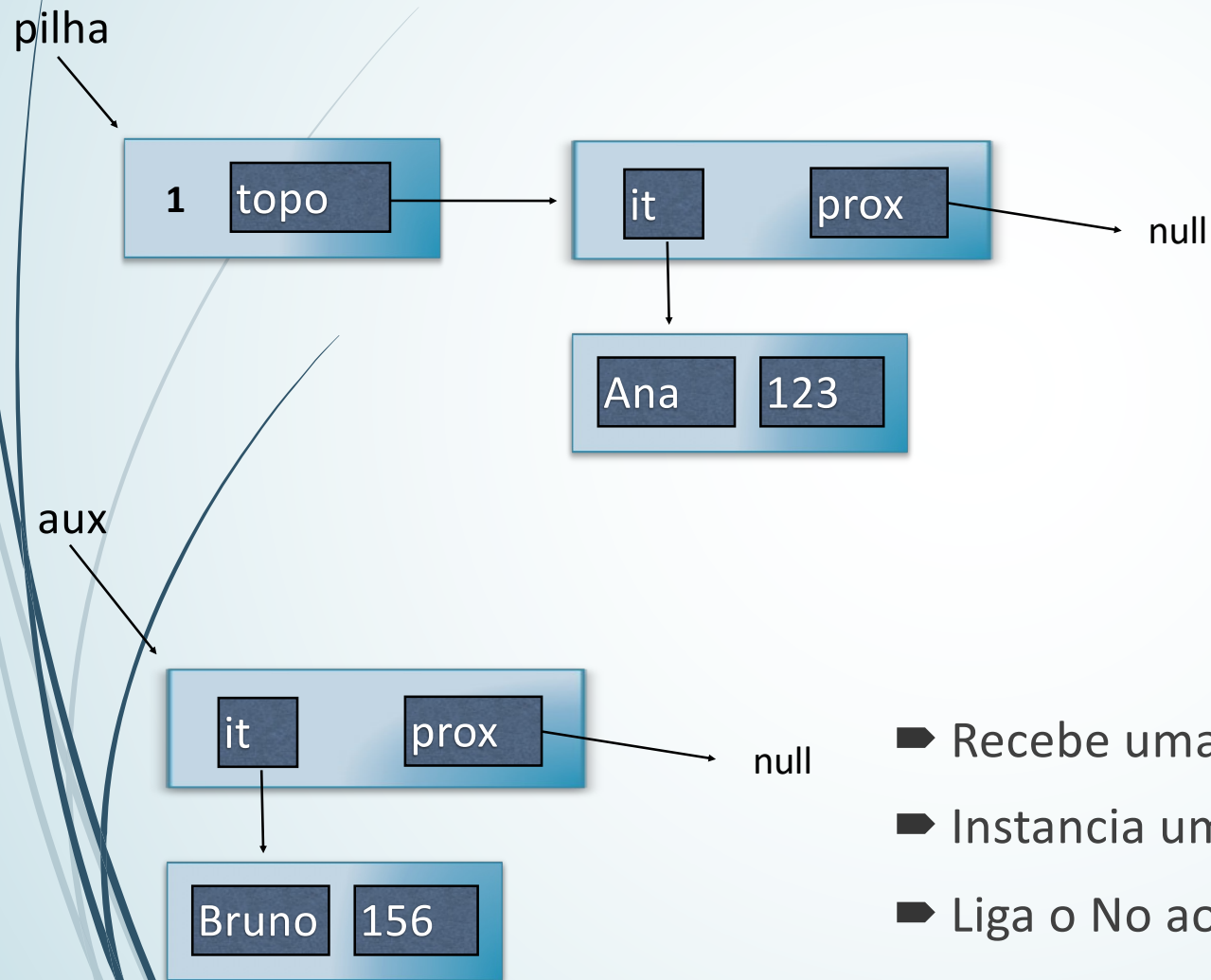
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



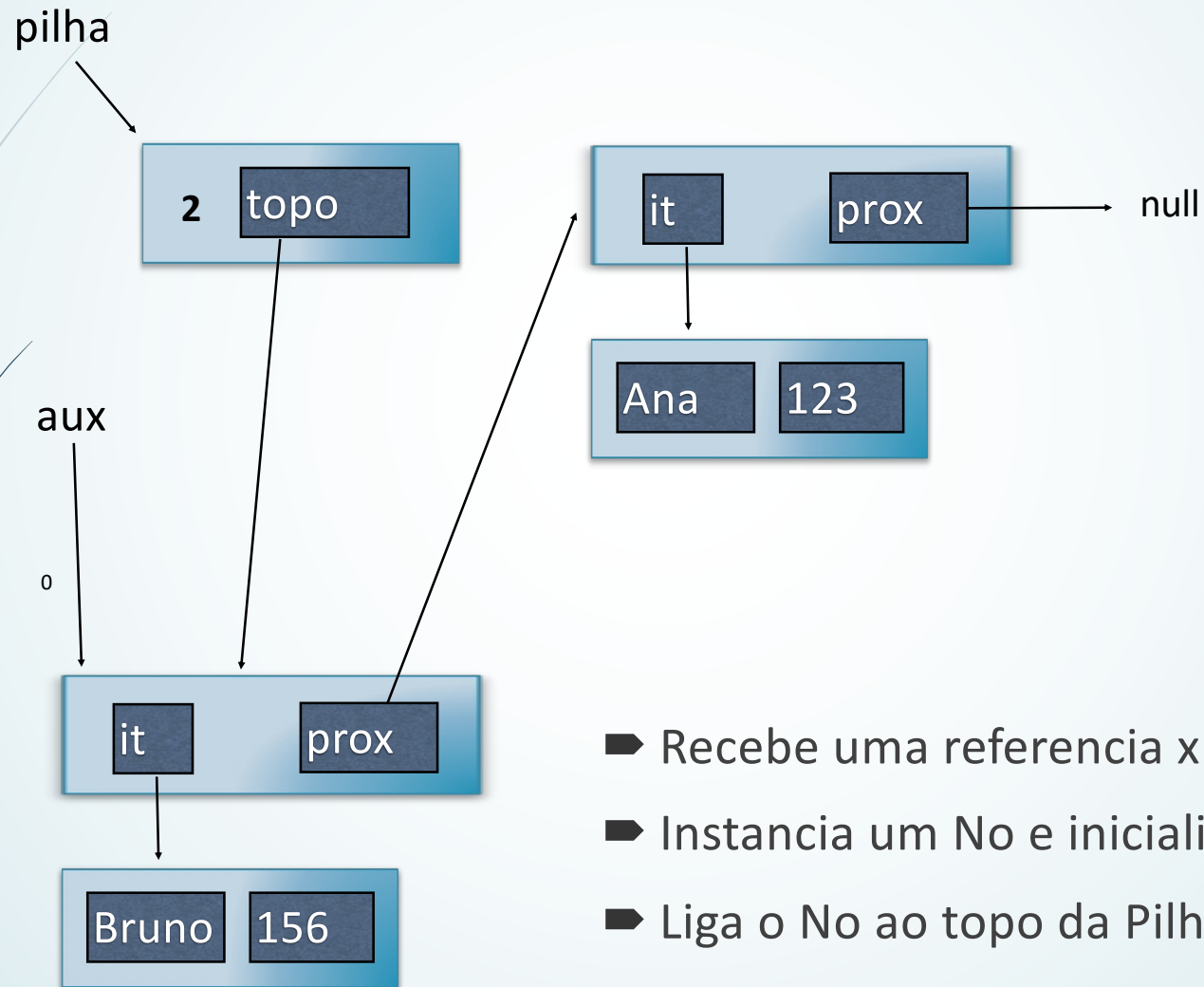
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



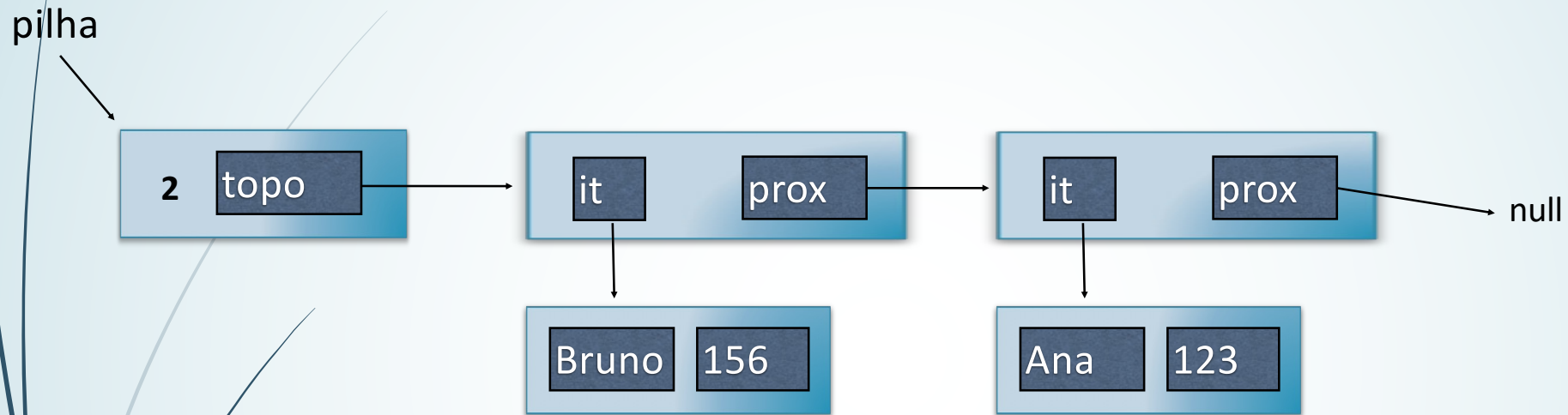
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



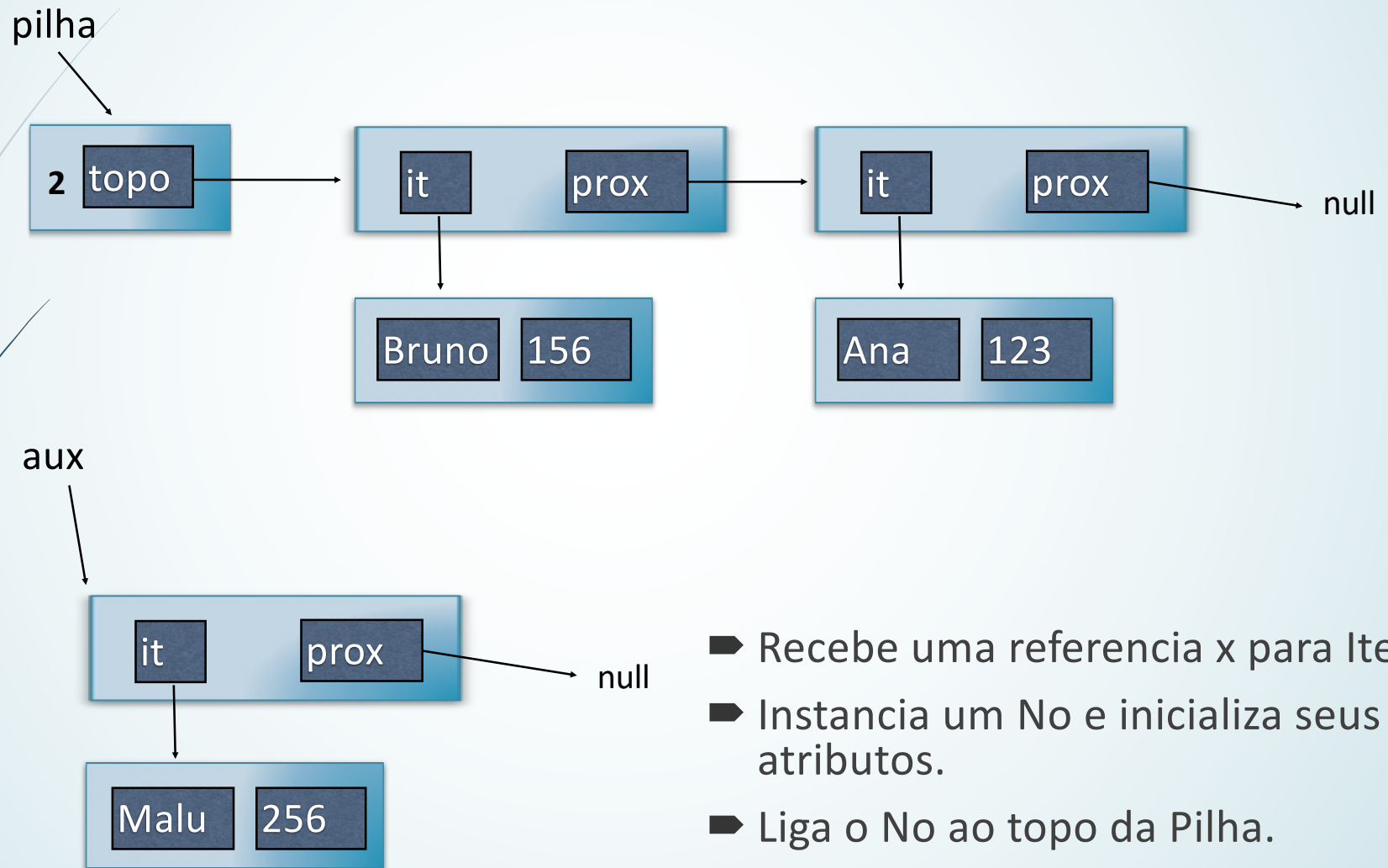
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



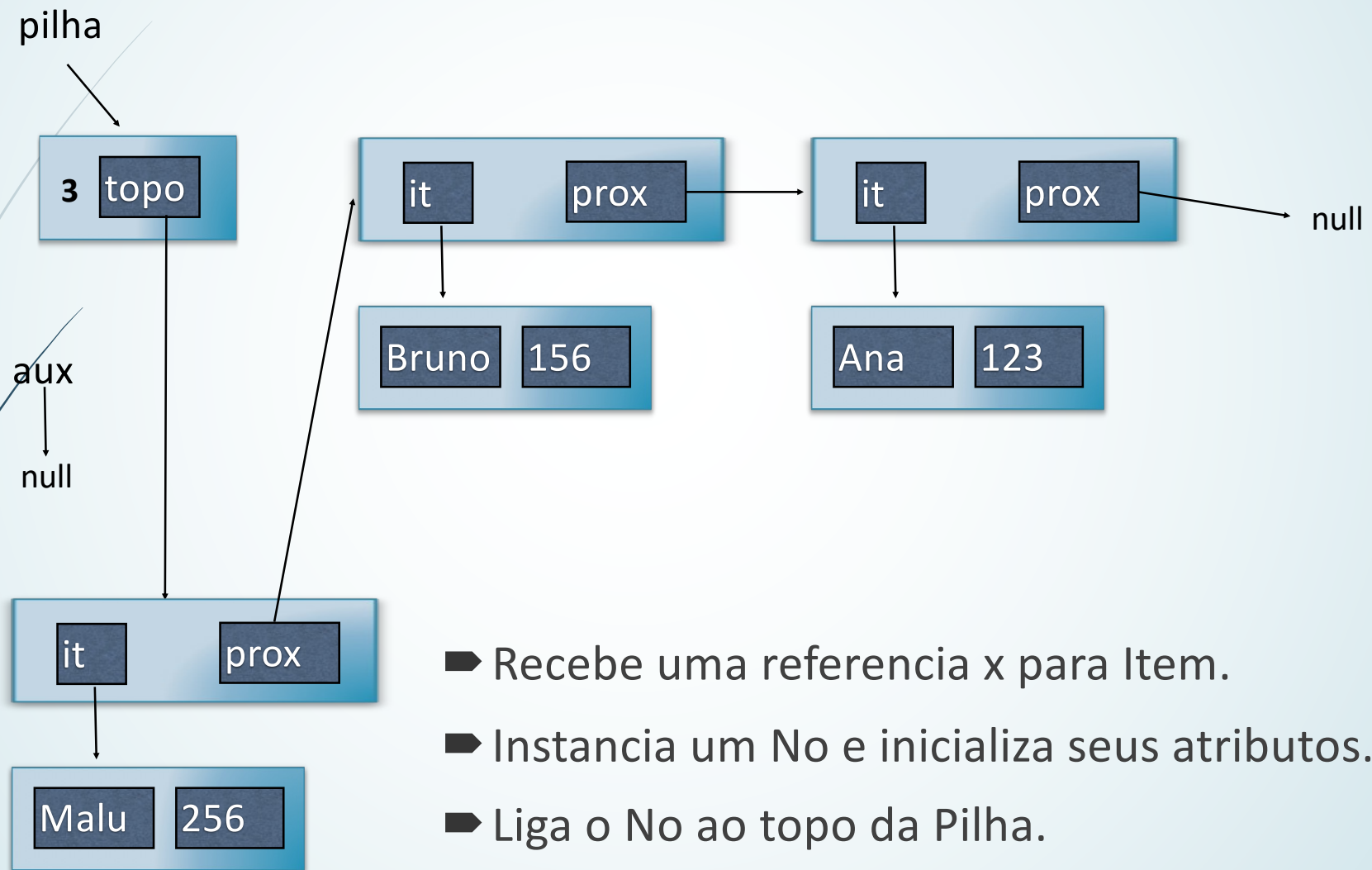
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



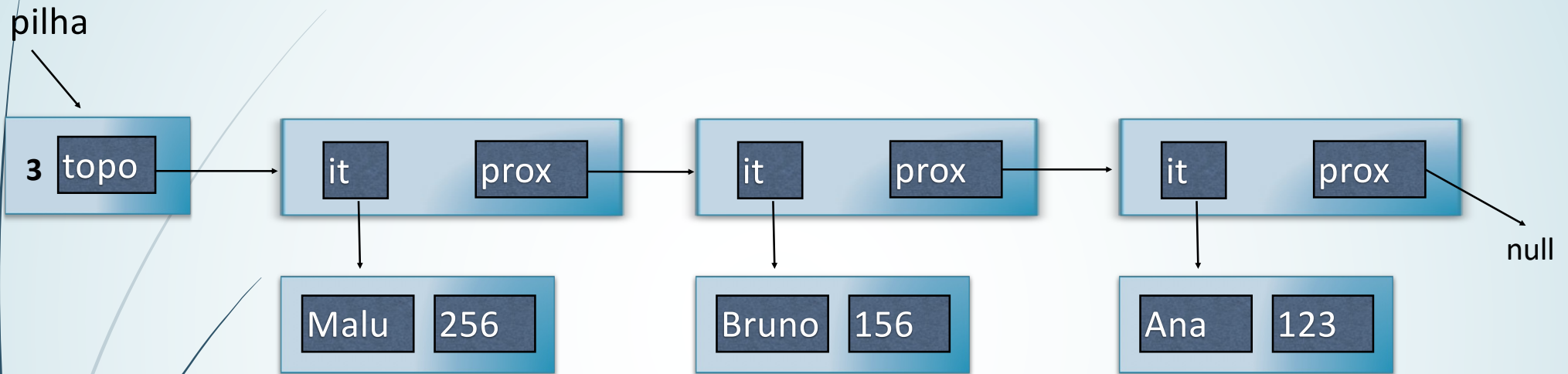
- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

push(Item x)



- Recebe uma referencia x para Item.
- Instancia um No e inicializa seus atributos.
- Liga o No ao topo da Pilha.

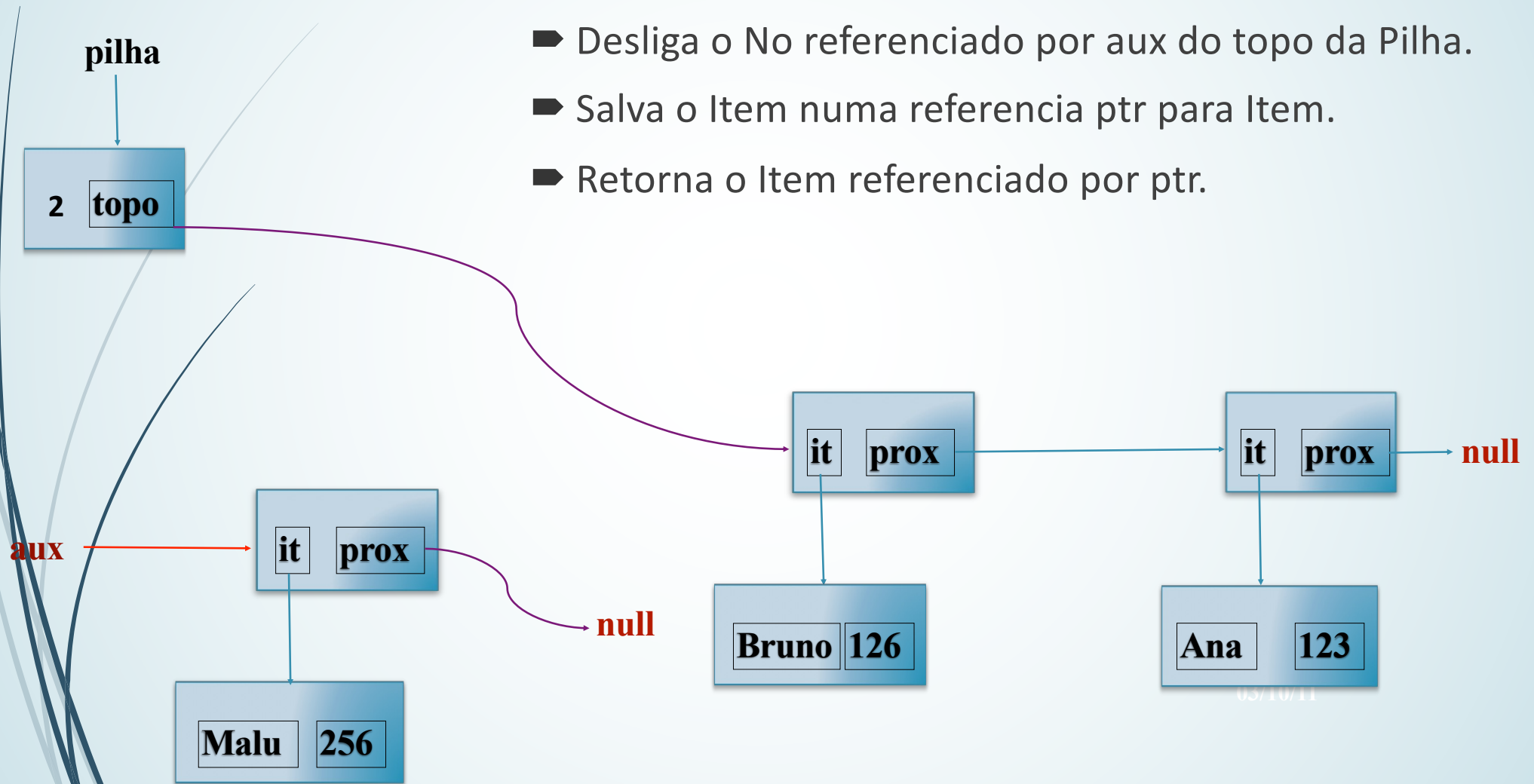
Item pop()

- Se a pilha não estiver vazia.
- Salva o No do topo numa referencia aux para Item.
- Desliga o No referenciado por aux do topo da Pilha.
- Salva o Item numa referencia ptr para Item.
- Retorna o Item referenciado por ptr.

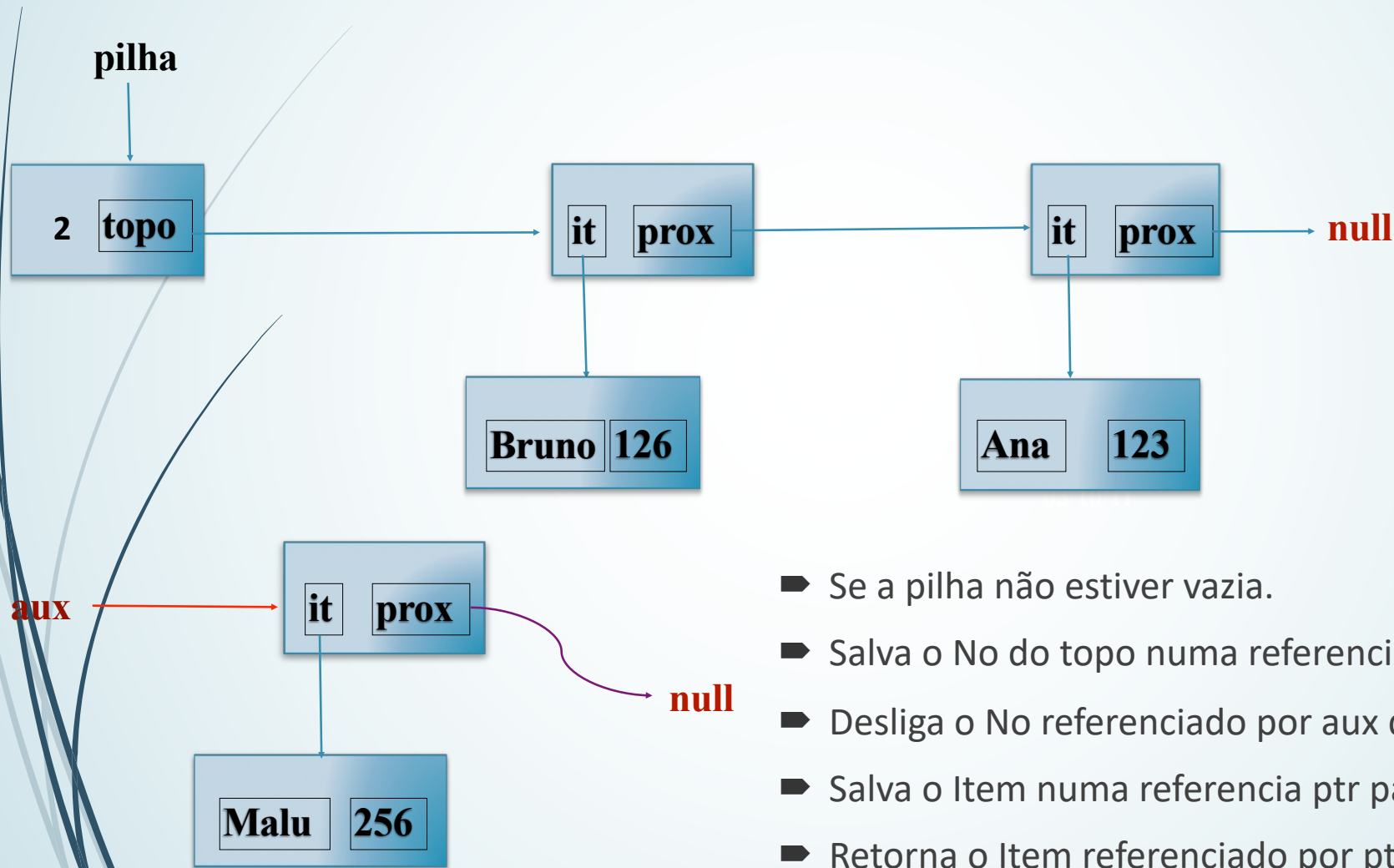


Item pop()

- Se a pilha não estiver vazia.
- Salva o No do topo numa referencia aux para Item.
- Desliga o No referenciado por aux do topo da Pilha.
- Salva o Item numa referencia ptr para Item.
- Retorna o Item referenciado por ptr.

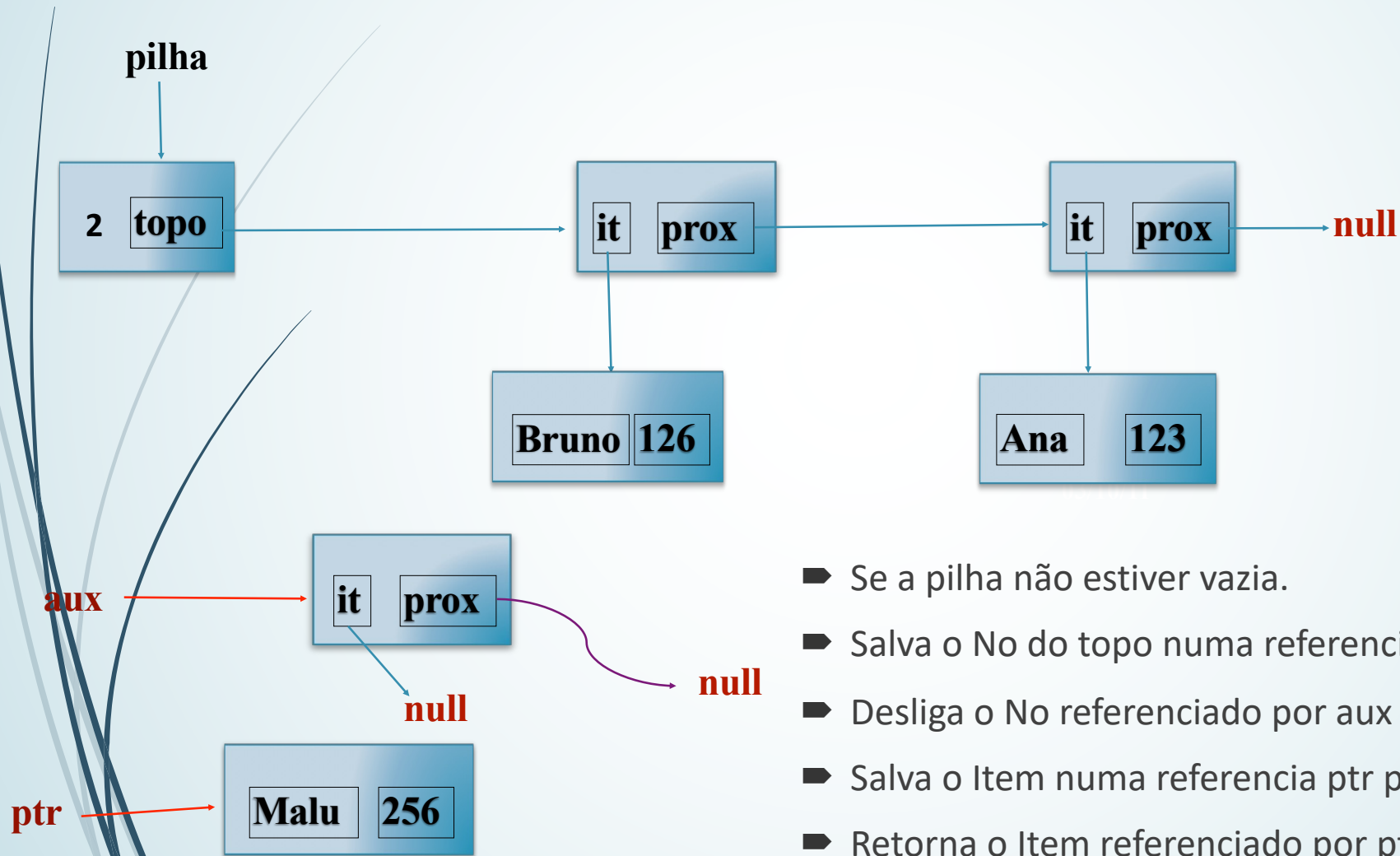


Item pop()



- Se a pilha não estiver vazia.
- Salva o No do topo numa referencia aux para Item.
- Desliga o No referenciado por aux do topo da Pilha.
- Salva o Item numa referencia ptr para Item.
- Retorna o Item referenciado por ptr.

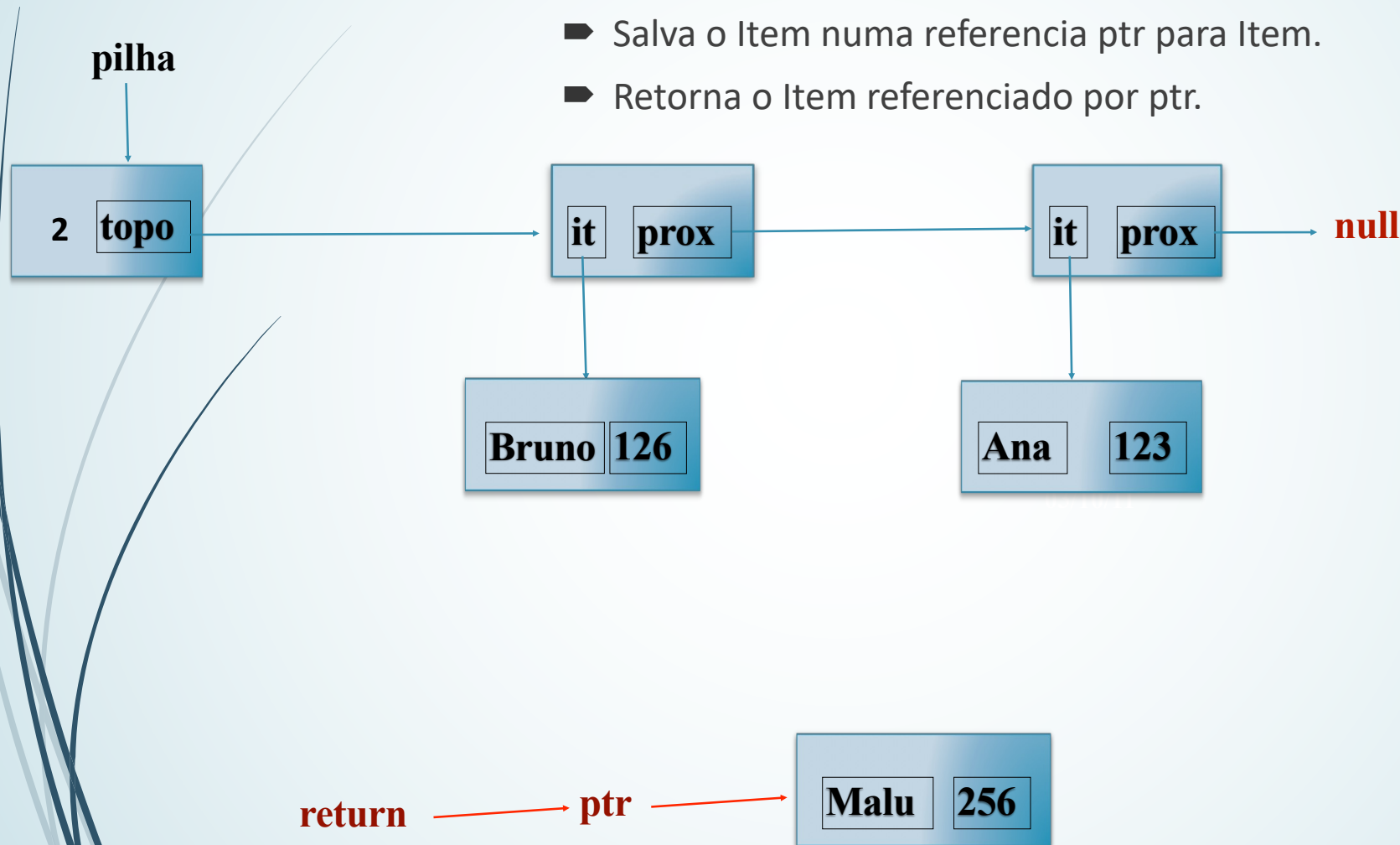
Item pop()



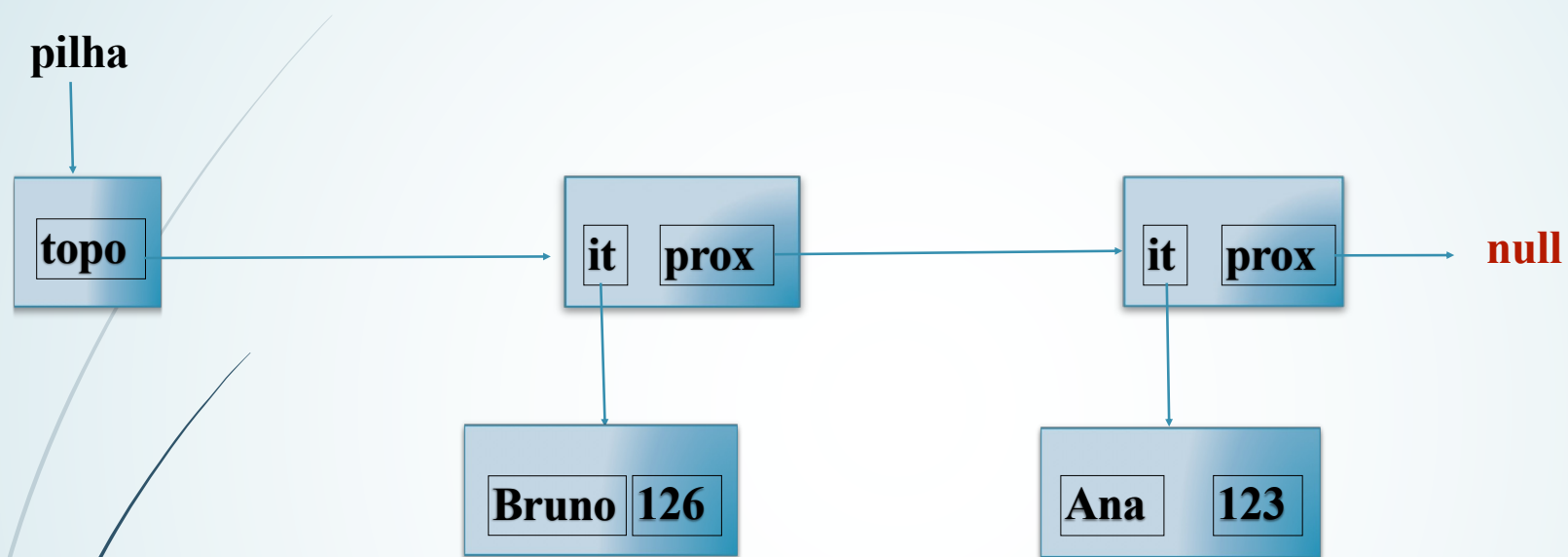
- Se a pilha não estiver vazia.
- Salva o No do topo numa referencia aux para Item.
- Desliga o No referenciado por aux do topo da Pilha.
- Salva o Item numa referencia ptr para Item.
- Retorna o Item referenciado por ptr.

Item pop()

- Se a pilha não estiver vazia.
- Salva o No do topo numa referencia aux para Item.
- Desliga o No referenciado por aux do topo da Pilha.
- Salva o Item numa referencia ptr para Item.
- Retorna o Item referenciado por ptr.



Estado final da Pilha



UML

Classe Livro

- titulo: String
- autor: String
- editora: String
- + getLivro(): String
- + Livro(String, String, String)

Classe Main (Base)

- leia: Scanner, static
- + main(): void, static
- + obtemLivro(): Livro, static

Classe Pilha

Classe No

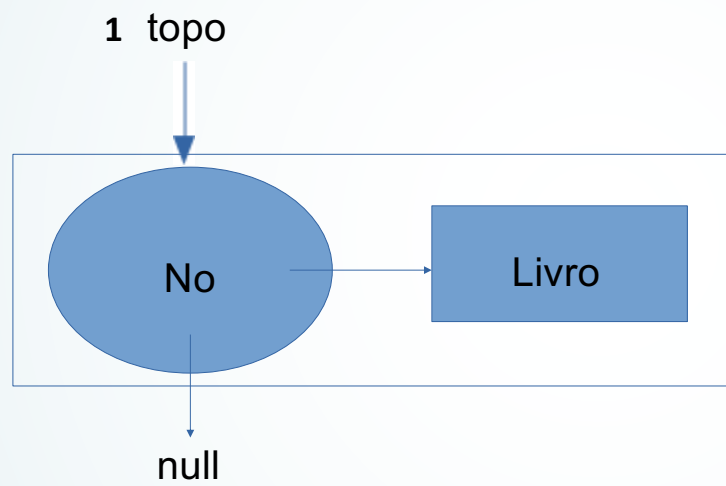
- dados: Livro
- prox: No
- + No()

- topo: No
- qde: int
- + empilha(Livro): void
- + desempilha(): Livro
- + vazia(): lógico
- + getQde(): int
- + Pilha()

Pilha vazia

0 topo
↓
null

Pilha com 1 livro



Pilha com 3 livros

