UFSC-CTC-INE

INE5384 - Estruturas de Dados

Pilhas

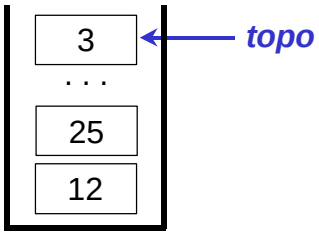
Prof. Ronaldo S. Mello

2002/2

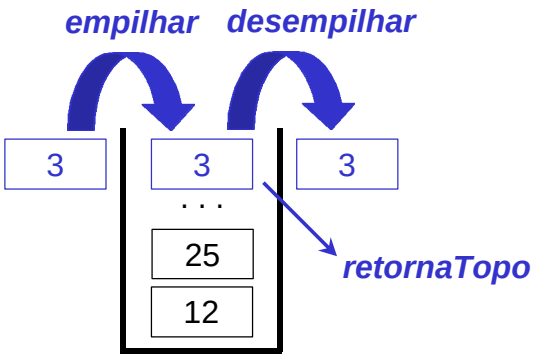
Pilha

- Uma pilha mantém uma seqüência de elementos dispostos um sobre o outro:
  - pilha de livros
  - pilha de caixas
  - pilha de “objetos” quaisquer, ...
- Uma pilha é um tipo especial de lista:
  - inserções e exclusões de elementos ocorrem em uma única extremidade da lista

Exemplo de Pilha



Operações sobre uma Pilha



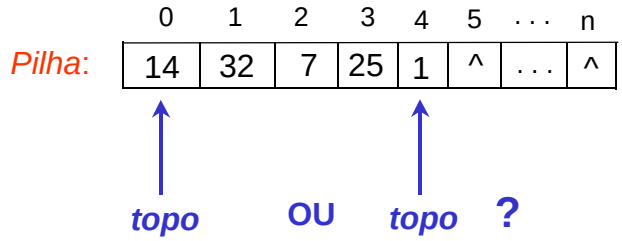
LIFO: “last-in; first-out”

Alternativas de Implementação

- Pilha como vetor
- Pilha como lista encadeada

Pilha como Vetor

NroElementos: 5



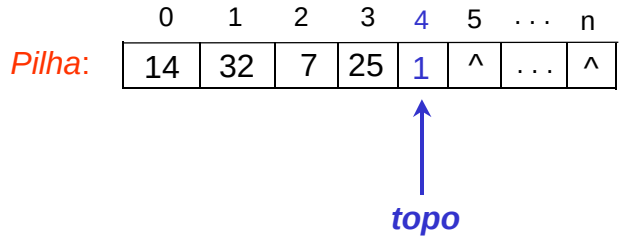
Onde deve ficar o topo da pilha:
No início ou no fim?

Vetor X Encadeamento

Operação	Vetor	Encadeamento
Inserção início	O(n)	O(1)
Inserção final	O(1)	O(1)
Inserção posição "x"	O(n)	O(n)
Pesquisa elemento "x"	O(n)	O(n)
Pesquisa posição "x"	O(1)	O(n)
Exclusão início	O(n)	O(1)
Exclusão final	O(1)	O(n)
Exclusão posição ou elemento "x"	O(n)	O(n)

Pilha como Vetor

NroElementos: 5



Onde deve ficar o topo da pilha?
No fim!

Implementação

```
Classe PilhaVetor
início
    pilha objeto[ ];
    nroElementos inteiro;
    ...
    Construtor PilhaVetor (tamanho inteiro);
    início
        pilha ← NOVO objeto[tamanho];
        nroElementos ← 0;
    fim;
fim
```

Implementação

```
Classe PilhaVetor
início
    pilha objeto[ ];
    nroElementos inteiro;
    ...
    Método empilhe(objeto Object)
    início
        se nroElementos = pilha.lenght então
            Exceção EstruturaCheia();
        pilha[nroElementos] ← objeto;
        nroElementos ← nroElementos + 1;
    fim;
fim
```

Implementação

```
Classe PilhaVetor
início
    pilha objeto[ ];
   nroElementos inteiro;
    ...
    Método desempilhe() retorna objeto;
   início
          resultado objeto;
          se nroElementos = 0 então
                Exceção EstruturaVazia();
          resultado ← pilha[nroElementos - 1];
          pilha [nroElementos - 1] ← NULL;
          nroElementos ← nroElementos - 1;
          retorna resultado;
    fim;
fim
```

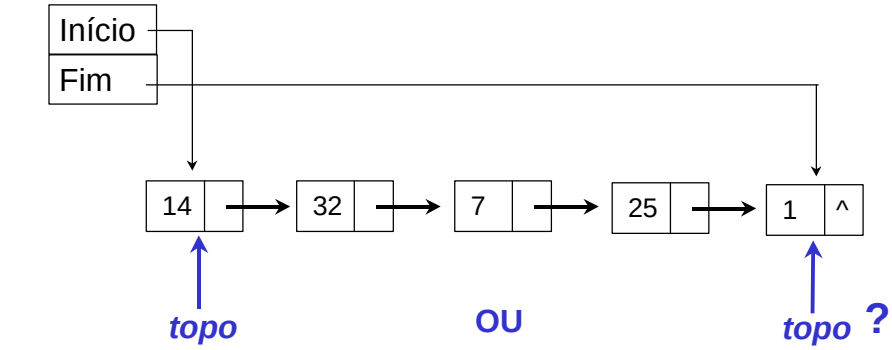
Implementação

```
Classe PilhaVetor
início
    pilha objeto[ ];
    nroElementos inteiro;
    ...
    Método retTopo() retorna objeto;
    início
          se nroElementos = 0 então
                Exceção EstruturaVazia();
          retorna pilha[nroElementos - 1];
    fim;
fim
```

Pilha como Lista Encadeada

NroElementos: 5

Pilha:



Onde deve ficar o topo da pilha: no início ou no fim?

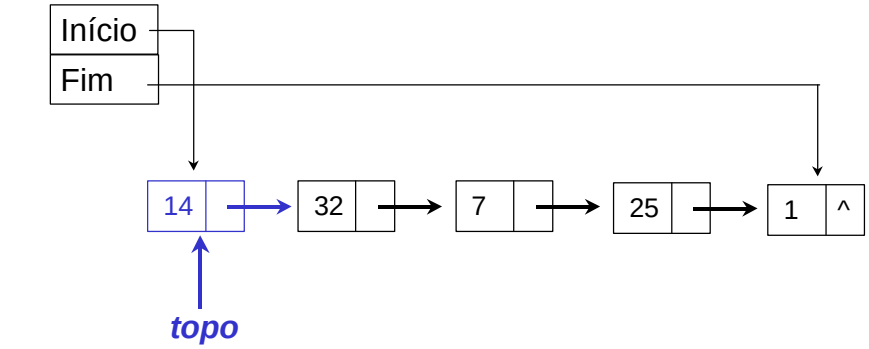
Vetor X Encadeamento

Operação	Vetor	Encadeamento
Inserção início	O(n)	O(1)
Inserção final	O(1)	O(1)
Inserção posição "x"	O(n)	O(n)
Pesquisa elemento "x"	O(n)	O(n)
Pesquisa posição "x"	O(1)	O(n)
Exclusão início	O(n)	O(1)
Exclusão final	O(1)	O(n)
Exclusão posição ou elemento "x"	O(n)	O(n)

Pilha como Lista Encadeada

NroElementos: 5

Pilha:



Onde deve ficar o topo da pilha? No início!

Implementação

```
Classe PilhaEncadeada
início
    pilha ListaEncadeada;
    nroElementos inteiro;
    ...
    Construtor PilhaEncadeada ();
    início
        pilha ← NOVO ListaEncadeada();
        nroElementos ← 0;
    fim;
fim
```

Exercícios

- Implementar na classe *PilhaEncadeada*:
  - empilhe(objeto object);
  - desempilhe() retorna object;
  - retTopo() retorna object;

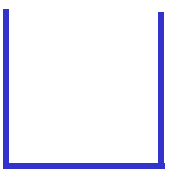
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos

```
Método X();
início
   instrução1;
   Y();
   instrução3;
   Z();
   instrução5;
fim;
```

```
Método Y();
início
   instrução1;
   instrução2;
   Z();
   instrução4;
fim;
```

```
Método Z();
início
   instrução1;
   instrução2;
fim;
```



Pilha de execução de métodos

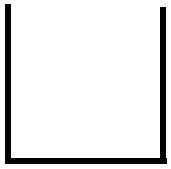
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos

```
Método X();
início
  instrução1;
  Y();
  instrução3;
  Z();
  instrução5;
fim;
```

```
Método Y();
início
  instrução1;
  instrução2;
  Z();
  instrução4;
fim;
```

```
Método Z();
início
  instrução1;
  instrução2;
fim;
```



Pilha de execução de métodos

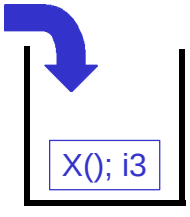
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos

```
Método X();
início
  instrução1;
  Y();
  instrução3;
  Z();
  instrução5;
fim;
```

```
Método Y();
início
  instrução1;
  instrução2;
  Z();
  instrução4;
fim;
```

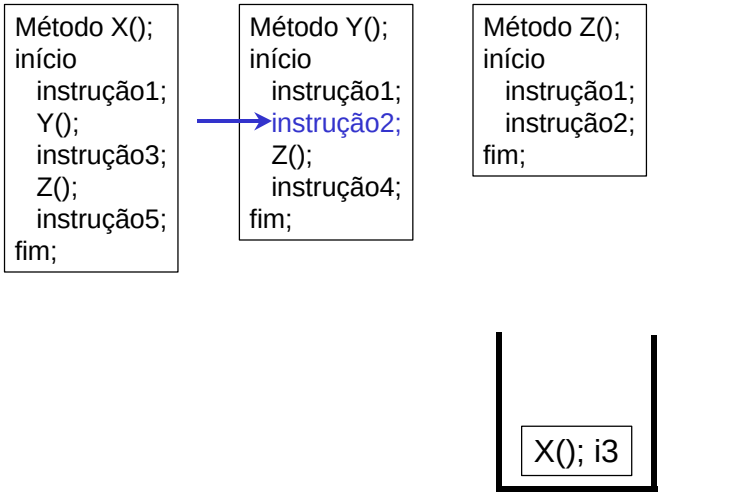
```
Método Z();
início
  instrução1;
  instrução2;
fim;
```



Pilha de execução de métodos

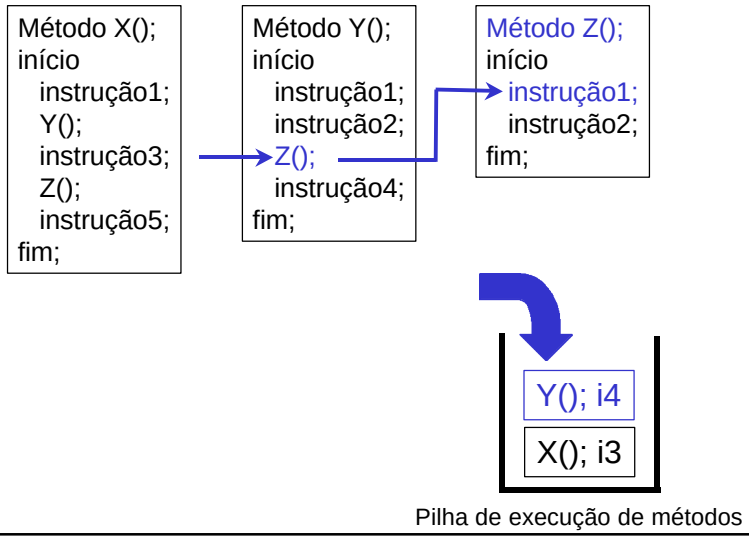
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



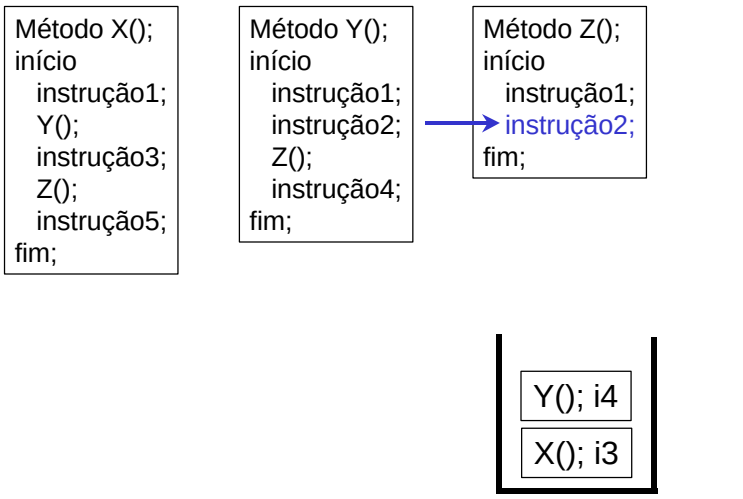
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



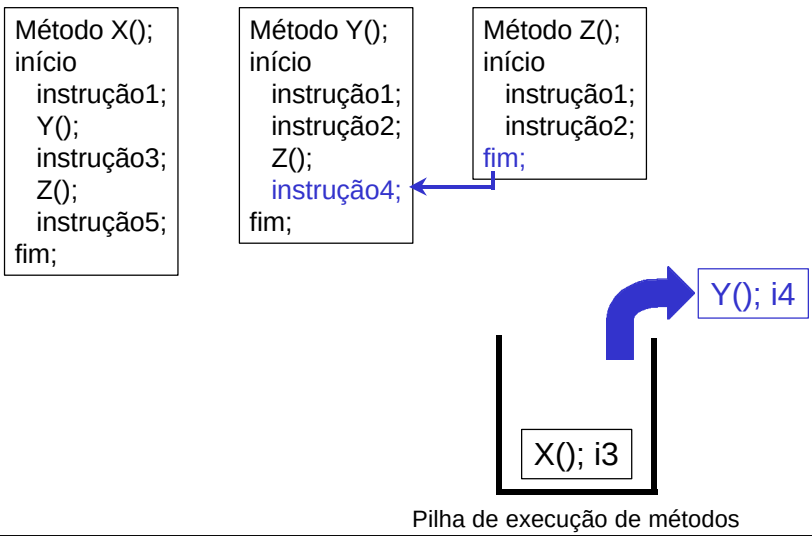
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



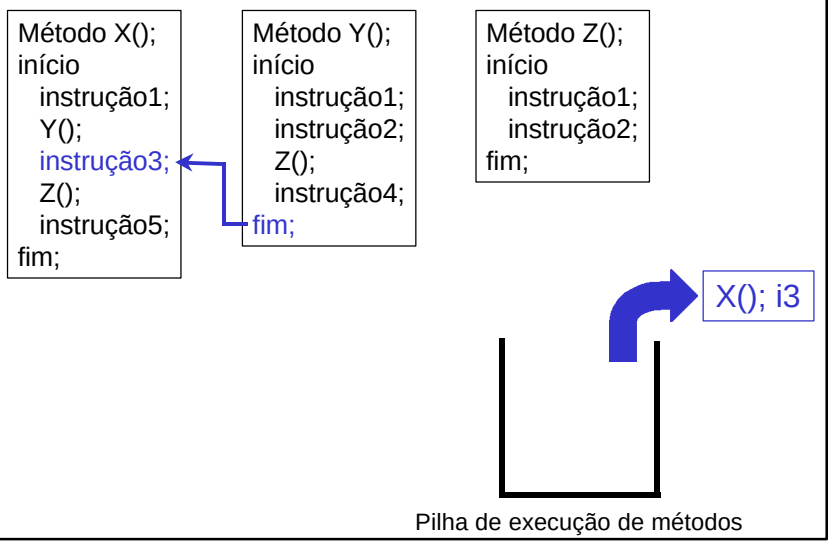
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



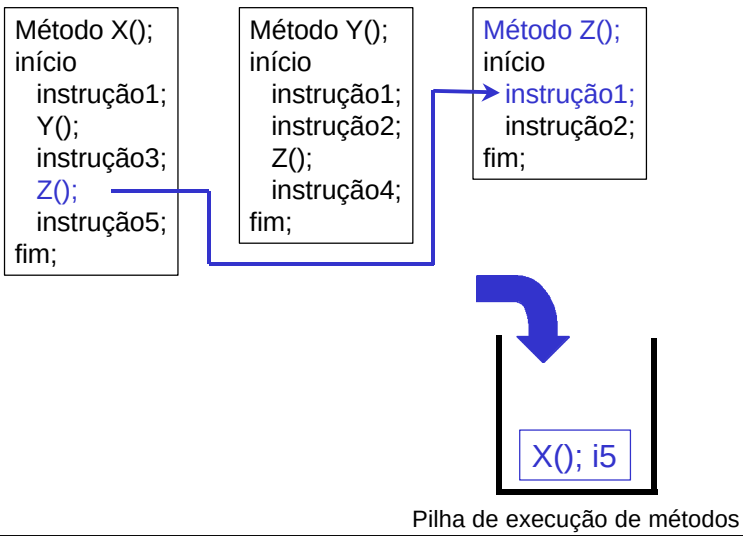
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



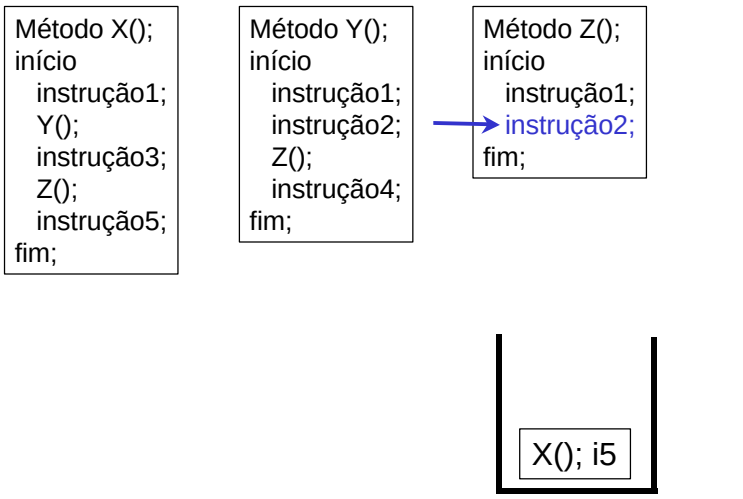
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



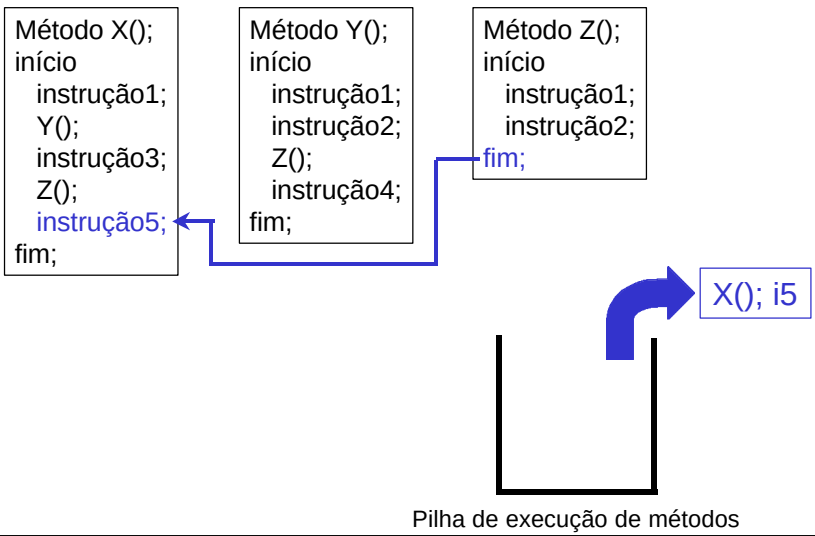
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos



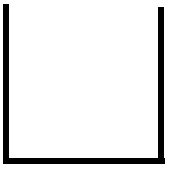
Exemplo de Aplicação de Pilhas

- Controle de chamadas de métodos

```
Método X();
início
   instrução1;
   Y();
   instrução3;
   Z();
   instrução5;
fim;
```

```
Método Y();
início
   instrução1;
   instrução2;
   Z();
   instrução4;
fim;
```

```
Método Z();
início
   instrução1;
   instrução2;
fim;
```



Pilha de execução de métodos

Exercício

Ler números inteiros positivos até que um zero seja informado (usar métodos *Ler()* e *Escrever()*). Quando o número -1 for lido, some os números lidos anteriormente mais o resultado da última operação, se houver, e exiba o resultado. Quando -2 for lido, faça o mesmo processo, porém multiplique ao invés de somar. Quando -3 for lido, subtraia do último elemento os números anteriores já lidos e o resultado da última operação, se houver.

Exemplo 1:	5
1	7
2	-3 → resultado: -22
-1 → resultado: 3	10
4	-1 → resultado: -12
2	0
-2 → resultado: 24	

Exercício

Ler números inteiros positivos até que um zero seja informado (usar métodos *Ler()* e *Escrever()*). Quando o número -1 for lido, some os números lidos anteriormente, mais o resultado da última operação, se houver, e exiba o resultado. Quando -2 for lido, faça o mesmo processo, porém multiplique ao invés de somar. Quando -3 for lido, subtraia do último elemento os números anteriores já lidos e o resultado da última operação, se houver.

Exemplo 2:	40
12	12
5	6
-2 → resultado: 60	1
4	-1 → resultado: 5
10	8
-3 → resultado: -54	0