

Estrutura de Dados

- Na computação, uma Estrutura de Dados é um modo particular de

armazenamento e organização de dados em um computador, de

modo que possam ser usados eficientemente;

Prof. Calvetti 2/95

Estrutura de Dados

Dentre as principais Estrutura de Dados existentes, temos:

- Pilha;
- Fila;
- Lista;
- Árvore Binária;
- Grafo; e
- Tabela de Hashing.

Prof. Calvetti 3/95

Pilha (Stack) – LIFO

- A Pilha ou *Stack* é uma estrutura de dados baseada no princípio LIFO (*Last In*, *First Out*), no qual um dado nela inserido em primeiro será removido dela por último:







Prof. Calvetti

4/94

Pilha (Stack) – LIFO

Existem duas funções que se aplicam às pilhas:

- push: Insere um dado no topo da Pilha; e

- *pop*: Remove o item do topo da Pilha.

Prof. Calvetti 5/95

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

E D C B A

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

$$E D C B \xrightarrow{push}$$

A

Autor. Prof Robson Calvetti - Todos os direitos reservados

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

E D C B

A

Autor: Prof. Robson Calvetti - Todos os direitos reserva

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

$$E D C \xrightarrow{push}$$

B

A

Prof. Calvetti

9/95

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

E D C

B

A

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

 $E D \xrightarrow{push}$

B

A

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

11/95

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

E D

R

A

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

12/95

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

 $E \xrightarrow{push}$

 O

 \Box

В

A

other: Prof Robson Calvetti - Todos os direitos reservados

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

E

 \mathbf{C}

7

В

A

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

14/95

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Pilha</u> <u>Saída</u>

push

E

 O

 \mathcal{I}

В

A

tor: Prof Robson Calvetti - Todos os direitos reservados

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

<u>Saída</u>

E

 O

7

В

Α

Autor: Prof. Robson Calvetti - Todos os direitos reservados

Prof. Calvetti

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

pop

<u>Saída</u>

E

 D

7

В

 \mathbf{A}

Prof. Calvetti

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

<u>Saída</u>

E

 C

7

В

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

pop

Saída

D E

В

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

<u>Saída</u>

DE

B

A

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

20/95

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

CDE

В

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

<u>Saída</u>

CDE

B

Pilha (Stack) – LIFO

Entrada

<u>Pilha</u>

Saída

BCDE

Pilha (Stack) – LIFO

Entrada

Pilha

Saída

BCDE

Pilha (Stack) – LIFO

Entrada

Pilha

Saída

A B C D E

Pilha (Stack) – LIFO

Entrada

Pilha

<u>Saída</u>

ABCDE

Pilha (Stack) – LIFO

<u>Entrada</u> <u>Saída</u>

E D C B A inversão de ordem A B C D E

ção, total ou parcial, deste conteúdo sem a prévia autorização por escrit

Pilha (Stack) – LIFO

Exemplos de utilização de Pilhas:

- Chamada de Subprogramas (métodos, funções etc.);
- Inversão de ordem de uma coleção;

- Análise de Expressões e Sintaxe (Lógicas e Matemáticas).

Prof. Calvetti 28/95

Pilha (Stack) – LIFO

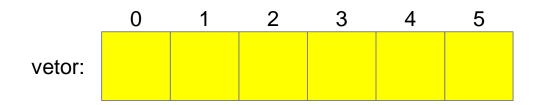
Outras funções que se aplicam às pilhas:

- size: Informa o tamanho da referida Pilha; e

- top: Informa o elemento no topo da Pilha, sem retirá-lo.

Prof. Calvetti 29/

Pilha (Stack) – LIFO



int vetor[] = new vetor[6];

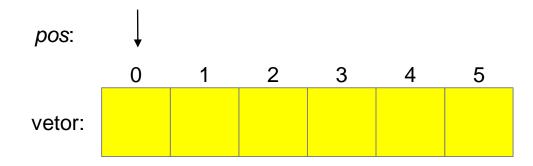
Vetor com o tamanho máximo necessário

Prof. Calvetti 30/95

Robson Calvetti - Todos os direitos reservados ©.

Estruturas de Dados e Análise de Algoritmos

Pilha (Stack) – LIFO

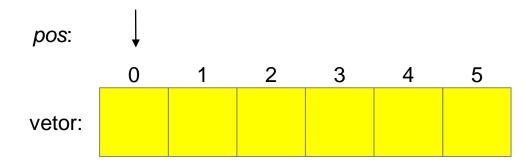


int pos; : : pos = 0;

Inicializa indicador de posição

Prof. Calvetti 31/95

Pilha (Stack) – LIFO

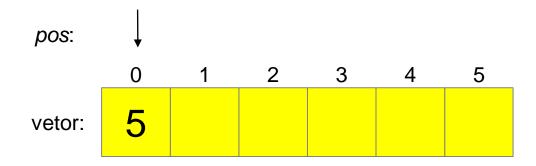


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(5);
```

Verifica se a pilha está cheia

Prof. Calvetti 32/95

Pilha (Stack) – LIFO

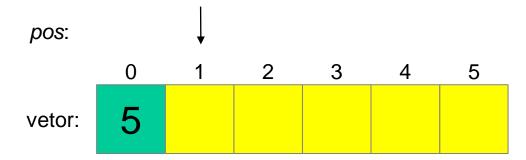


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(5);
```

Não está cheia! Coloca o elemento 5 na pilha

Prof. Calvetti 33/95

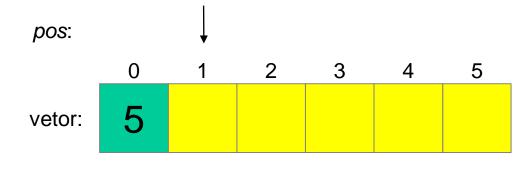
Pilha (Stack) – LIFO



Incrementa indicador de posição

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Pilha (Stack) – LIFO

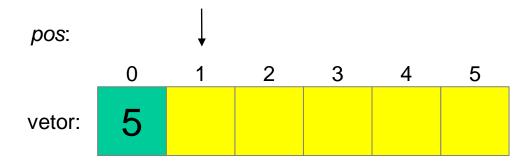


```
int topo;
:
topo = top();
//topo: 5
```

Consulta o topo da pilha

Prof. Calvetti 35/95

Pilha (Stack) – LIFO

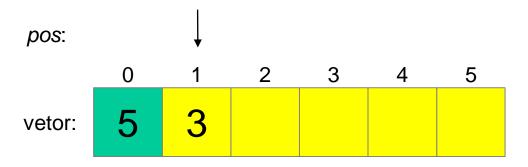


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(3);
```

Verifica se a pilha está cheia

Prof. Calvetti 36/95

Pilha (Stack) – LIFO

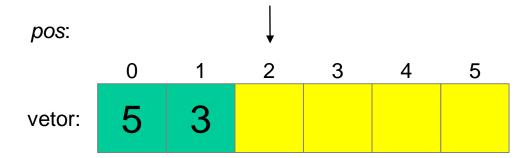


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(3);
```

Não está cheia! Coloca o elemento 3 na pilha

Prof. Calvetti 37/95

Pilha (Stack) – LIFO

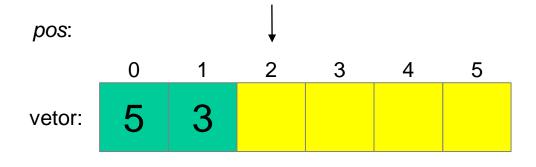


Incrementa indicador de posição

tor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

$Pilha\ (Stack) - LIFO$

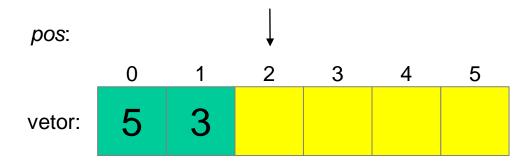


```
int topo;
:
:
topo = top();
//topo: 3
```

Consulta o topo da pilha

Prof. Calvetti 39/95

Pilha (Stack) – LIFO

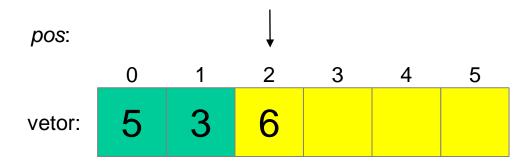


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(6);
```

Verifica se a pilha está cheia

Prof. Calvetti 40/95

Pilha (Stack) – LIFO

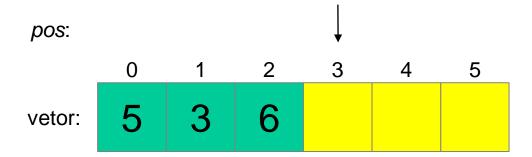


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(6);
```

Não está cheia! Coloca o elemento 6 na pilha

Prof. Calvetti 41/95



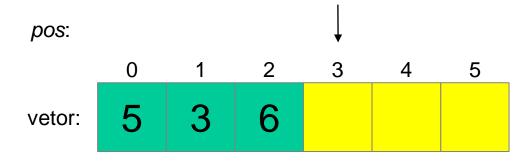


Incrementa indicador de posição

utor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti



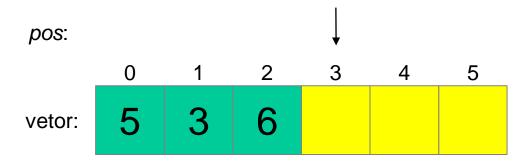


```
int topo;
:
topo = top();
//topo: 6
```

Consulta o topo da pilha

Prof. Calvetti 43/95

Pilha (Stack) – LIFO

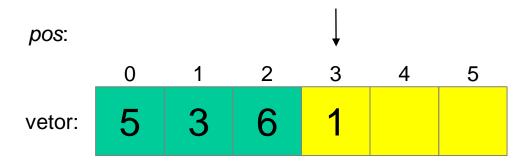


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(1);
```

Verifica se a pilha está cheia

Prof. Calvetti 44/95

Pilha (Stack) – LIFO

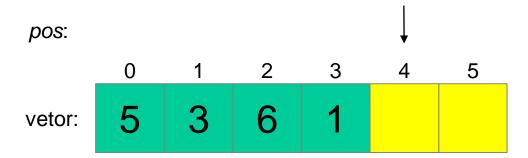


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(1);
```

Não está cheia! Coloca o elemento 1 na pilha

Prof. Calvetti 45/95

Pilha (Stack) – LIFO

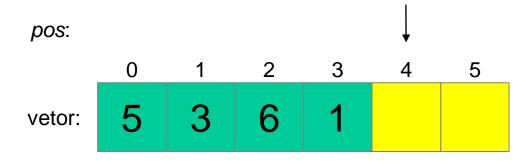


Incrementa indicador de posição

tor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

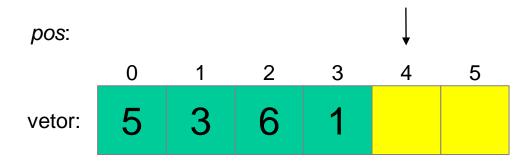


```
int topo;
:
topo = top();
//topo: 1
```

Consulta o topo da pilha

Prof. Calvetti 47/95

Pilha (Stack) – LIFO

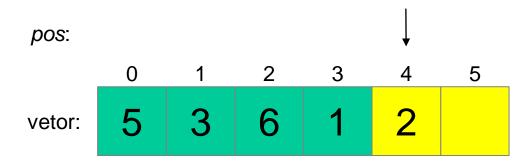


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(2);
```

Verifica se a pilha está cheia

Prof. Calvetti 48/95

Pilha (Stack) – LIFO

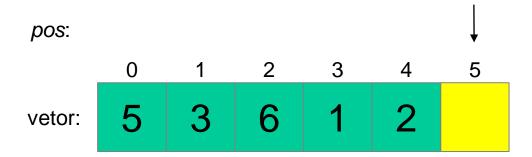


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(2);
```

Não está cheia! Coloca o elemento 2 na pilha

Prof. Calvetti 49/95



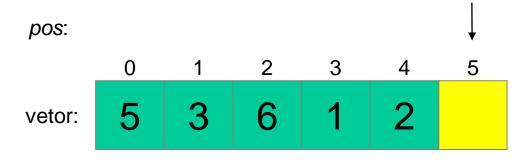


Incrementa indicador de posição

or: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

$Pilha\ (Stack) - LIFO$

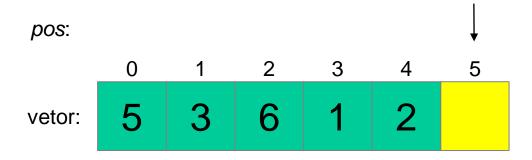


```
int topo;
:
topo = top();
//topo: 2
```

Consulta o topo da pilha

Prof. Calvetti 51/95

Pilha (Stack) – LIFO

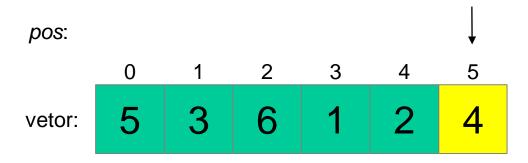


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(4);
```

Verifica se a pilha está cheia

Prof. Calvetti 52/95

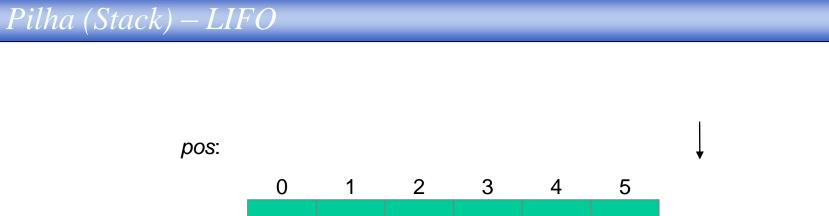
Pilha (Stack) – LIFO



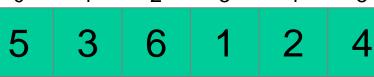
```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(4);
```

Não está cheia! Coloca o elemento 4 na pilha

Prof. Calvetti 53/95



vetor:

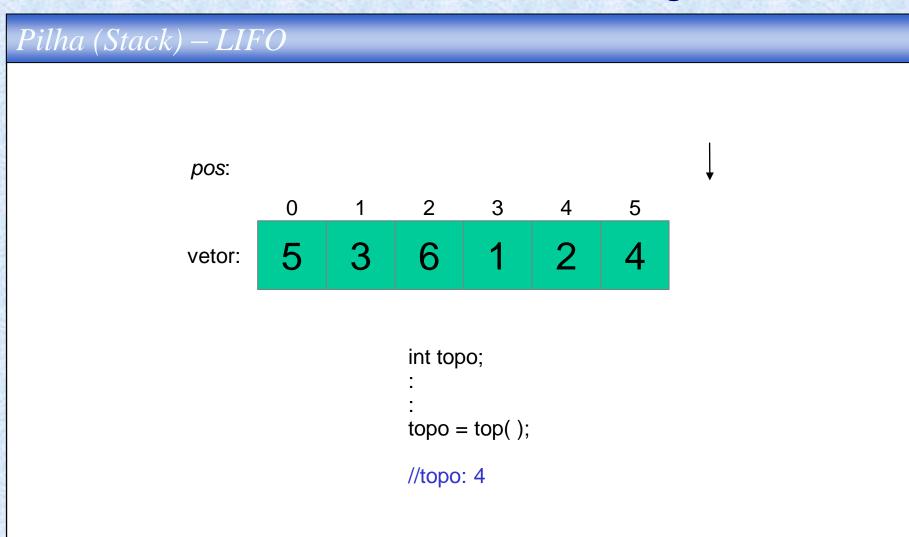


Incrementa indicador de posição

tor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

54/95



Consulta o topo da pilha

Prof. Calvetti 55/95

Pilha (Stack) – LIFO

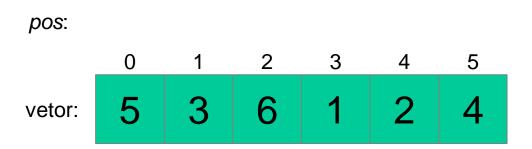


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(7);
```

Verifica se a pilha está cheia

Prof. Calvetti 56/95

Pilha (Stack) – LIFO

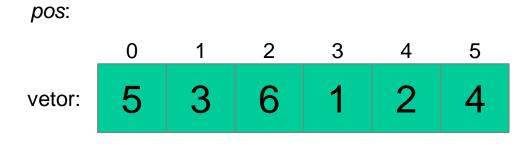


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(7);
```

Está cheia! Não coloca o elemento 7 na pilha

Prof. Calvetti 57/95

Pilha (Stack) – LIFO

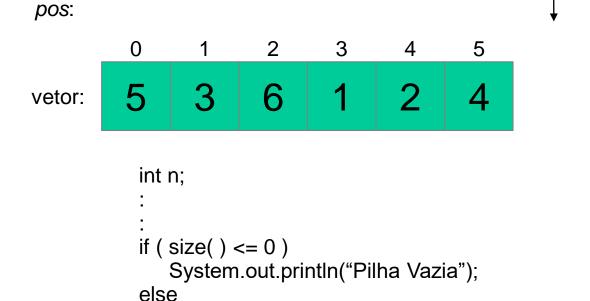


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 58/95

Pilha (Stack) – LIFO



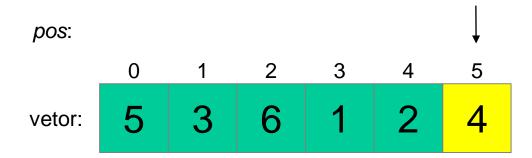
Não está vazia! Retira o topo da pilha

n = pop();

//n: 4

Prof. Calvetti 59/95

Pilha (Stack) – LIFO

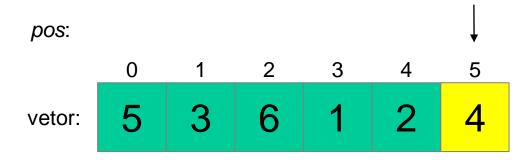


Decrementa indicador de posição

tor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

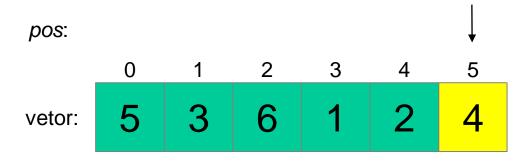


```
int topo;
:
topo = top();
//topo: 2
```

Consulta o topo da pilha

Prof. Calvetti 61/95

Pilha (Stack) – LIFO

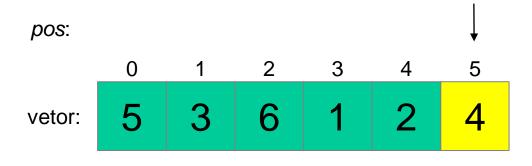


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 62/95

Pilha (Stack) – LIFO

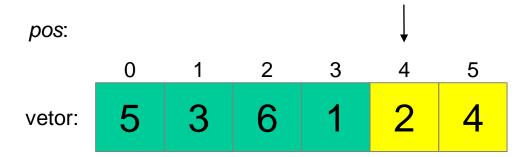


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );
//n: 2</pre>
```

Não está vazia! Retira o topo da pilha

Prof. Calvetti 63/95

Pilha (Stack) – LIFO

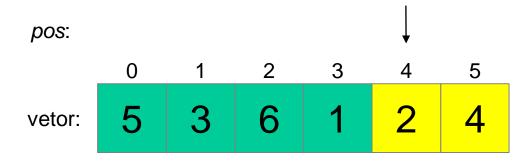


Decrementa indicador de posição

or: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

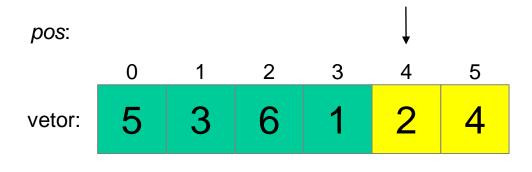


```
int topo;
:
:
topo = top();
//topo: 1
```

Consulta o topo da pilha

Prof. Calvetti 65/95

Pilha (Stack) – LIFO

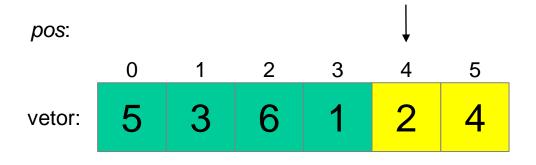


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 66/95

Pilha (Stack) – LIFO

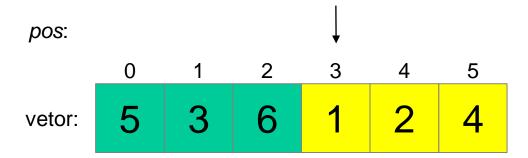


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );
//n: 1</pre>
```

Não está vazia! Retira o topo da pilha

Prof. Calvetti 67/95

Pilha (Stack) – LIFO

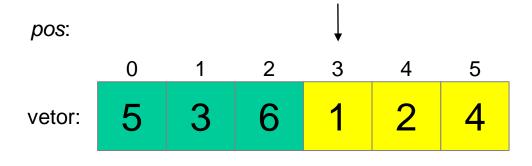


Decrementa indicador de posição

Autor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

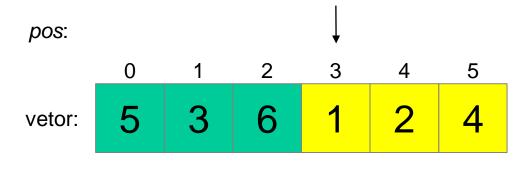


```
int topo;
:
:
topo = top();
//topo: 6
```

Consulta o topo da pilha

Prof. Calvetti 69/95

Pilha (Stack) – LIFO

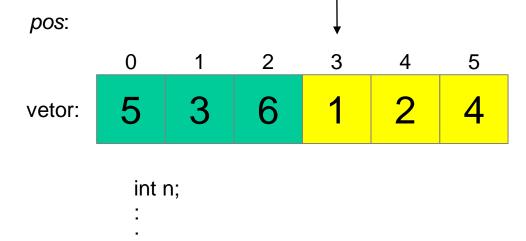


```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 70/95

Pilha (Stack) – LIFO



//n: 6

else

if $(size() \le 0)$

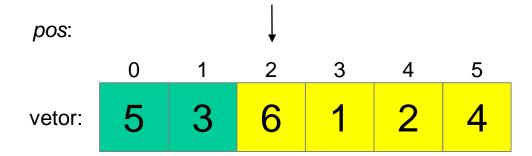
n = pop();

Não está vazia! Retira o topo da pilha

System.out.println("Pilha Vazia");

Prof. Calvetti 71/95

Pilha (Stack) – LIFO

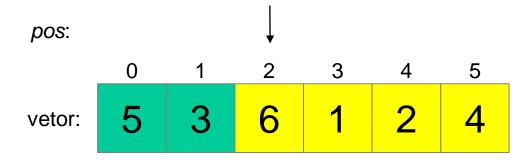


Decrementa indicador de posição

or: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti 72/95

Pilha (Stack) – LIFO

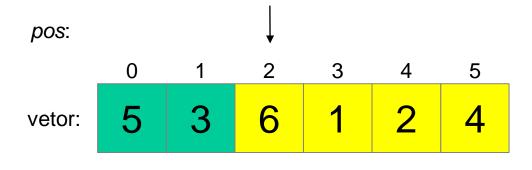


```
int topo;
:
topo = top();
//topo: 3
```

Consulta o topo da pilha

Prof. Calvetti 73/95

Pilha (Stack) – LIFO

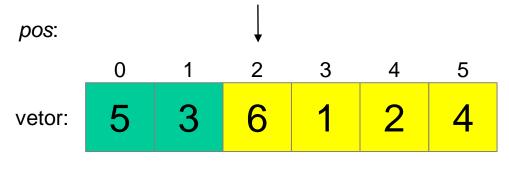


```
int n;
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 74/95

Pilha (Stack) – LIFO

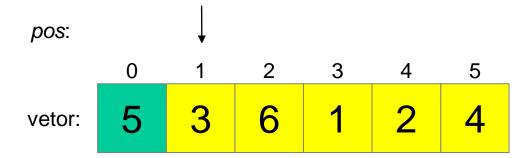


```
int n;
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );
//n: 3</pre>
```

Não está vazia! Retira o topo da pilha

Prof. Calvetti 75/95

Pilha (Stack) – LIFO

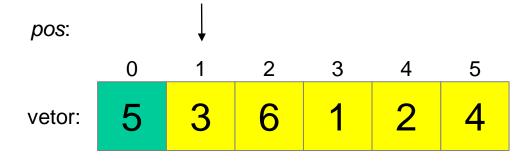


Decrementa indicador de posição

Nutor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti 76/95

Pilha (Stack) – LIFO

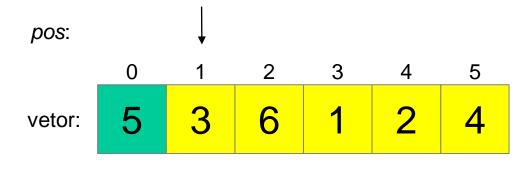


```
int topo;
:
:
topo = top();
//topo: 5
```

Consulta o topo da pilha

Prof. Calvetti 77/95

Pilha (Stack) – LIFO



```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 78/95

Pilha (Stack) – LIFO

```
pos:

0 1 2 3 4 5

vetor:

5 3 6 1 2 4
```

```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );
//n: 5</pre>
```

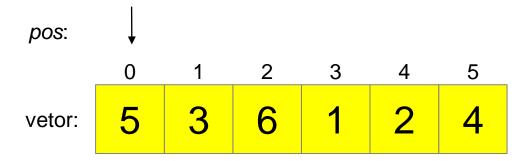
Não está vazia! Retira o topo da pilha

Prof. Calvetti 79/95

i - Todos os direitos reservados ©.

Estruturas de Dados e Análise de Algoritmos

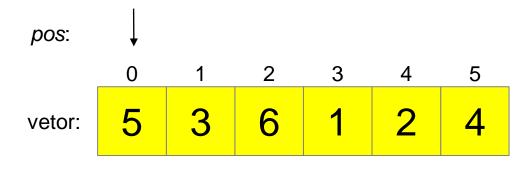
Pilha (Stack) – LIFO



Decrementa indicador de posição

Prof. Calvetti 80/95

Pilha (Stack) – LIFO



```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Verifica se a pilha está vazia

Prof. Calvetti 81/95

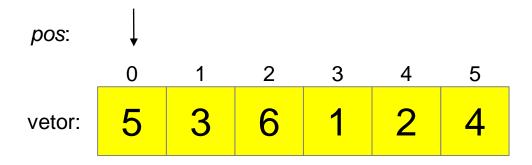
Pilha (Stack) – LIFO

```
int n;
:
:
if ( size( ) <= 0 )
    System.out.println("Pilha Vazia");
else
    n = pop( );</pre>
```

Está vazia! Não retira o topo da pilha

Prof. Calvetti 82/95

Pilha (Stack) – LIFO

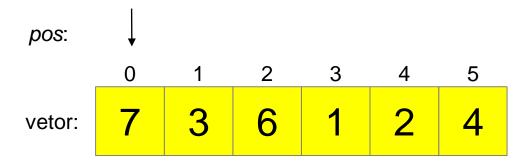


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(7);
```

Verifica se a pilha está cheia

Prof. Calvetti 83/95

Pilha (Stack) – LIFO

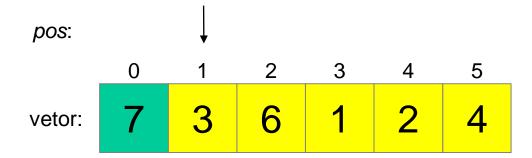


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(7);
```

Não está cheia! Coloca o elemento 7 na pilha

Prof. Calvetti 84/95

Pilha (Stack) – LIFO

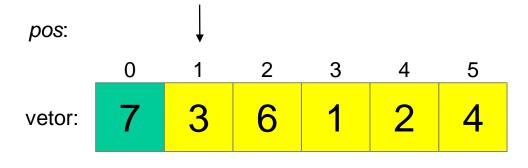


Incrementa indicador de posição

or: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti 85/95

Pilha (Stack) – LIFO

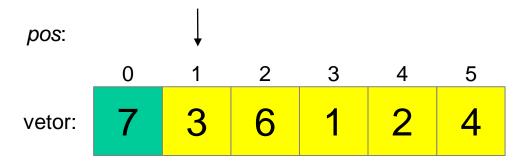


```
int topo;
:
topo = top();
//topo: 7
```

Consulta o topo da pilha

Prof. Calvetti 86/95

Pilha (Stack) – LIFO

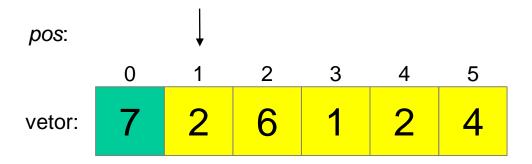


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(2);
```

Verifica se a pilha está cheia

Prof. Calvetti 87/95

Pilha (Stack) – LIFO

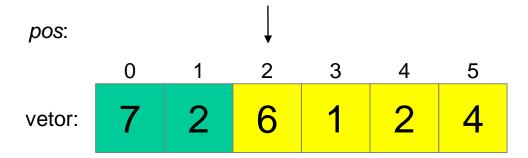


```
if ( size( ) >= vetor.length( ) )
    System.out.println("Pilha Cheia");
else
    push(2);
```

Não está cheia! Coloca o elemento 2 na pilha

Prof. Calvetti 88/95

Pilha (Stack) – LIFO

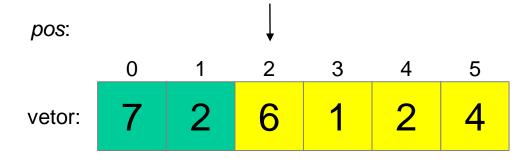


Incrementa indicador de posição

or: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

Pilha (Stack) – LIFO

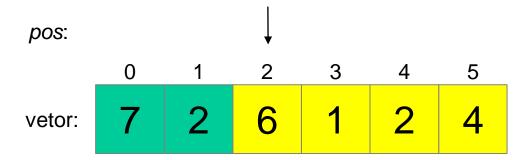


```
int topo;
:
:
topo = top();
//topo:2
```

Consulta o topo da pilha

Prof. Calvetti 90/95

Pilha (Stack) – LIFO



Repete qualquer operação: push, pop, size e top, conforme necessidade...

Insere ou retira elementos da pilha...

Prof. Calvetti 91/95

Pilha (Stack) – LIFO

Exemplo de códigos escrito em Java:

Codificação dos Métodos

utor: Prof. Robson Calvetti - Todos os direitos reservados ©.

Prof. Calvetti

92/95

Pilha (Stack) – LIFO

```
public
      static
            int
                     iTAM = 10;
public static int
                     iPilha[] = new int[iTAM];
public static
                     iPos = 0;
            int
public static int size()
   return iPos;
public static int top()
   return iPilha[iPos-1];
```

Métodos size() e top()

Prof. Calvetti 93/95

Pilha (Stack) – LIFO

```
public
      static
                     iTAM = 10;
             int
      static int
public
                      iPilha[] = new int[iTAM];
public static int
                      iPos = 0;
public static int push(int iC)
   if( iPos >= iTAM) return 0;
   return iPilha[iPos++] = iC;
public static int pop()
   if( iPos == 0) return 0;
   return iPilha[ --iPos ];
```

Método push() e pop()

Prof. Calvetti 94/95

FIM