

TECNOLOGIA EM INFORMÁTICA

Estrutura de Dados

Prof. Glauco da Silva
contato: glaucoslv@gmail.com

Roteiro

- *Objetivos*
- *Pilhas*
- *Exercícios*

Roteiro

- ***Objetivos***
- *Pilhas*
- *Exercícios*

Objetivos

- *Apresentar o conceito de pilhas, sua forma de funcionamento e como utilizá-las*

Roteiro

- *Objetivos*
- ***Pilhas***
- *Exercícios*

Pilhas

- *Permitem acesso somente a um item de dado: o último inserido*
- *São utilizadas para representar estruturas de dados mais complexas, como árvores binárias e grafos*
- *São utilizadas para validar expressões aritméticas*
- *Trabalham com a arquitetura LIFO (Last In First Out)*

Pilhas

- *Aplicações*

- *Arquitetura de microprocessadores: Quando uma função é chamada, seu endereço de retorno e seus argumentos são colocados em uma pilha; quando ela retorna, eles são retirados da pilha.*

- *Calculadoras: Utilizam as pilhas para armazenar resultados intermediários das operações*

Pilhas

- *Analogias*

- *Pilha de pratos;*

- *Pilha de cartas;*

- *Pilha de revistas;*

- *Latas empilhadas no corredor do supermercado;*

- *etc.*

Pilhas

- *Funções*

- *Empilhar (push)*

- *Adiciona um elemento no topo da pilha*

- *Desempilhar (pop)*

- *Remove o elemento que está no topo da pilha*

- *Ver o topo (peek)*

- *Torna possível ver qual é o valor armazenado no topo da pilha*

Pilhas

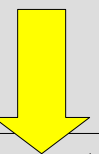
Pilha.java

```
public class Pilha {
    private int tamanho;
    private long[] pilhaVetor;
    private int top;

    public Pilha(int tam){
        tamanho = tam;
        pilhaVetor = new long[tamanho];
        top = -1;
    }

    public void push(long elem){
        pilhaVetor[++top] = elem;
    }

    public long pop(){
        return pilhaVetor[top--];
    }
}
```



Pilhas

Pilha.java

```
public long peek(){
    return pilhaVetor[top];
}

public boolean estaVazia(){
    return (top == -1);
}

public boolean estaCheia(){
    return (top == tamanho-1);
}
```

Pilhas

- *Exemplo*

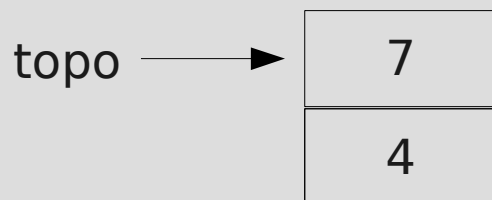
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

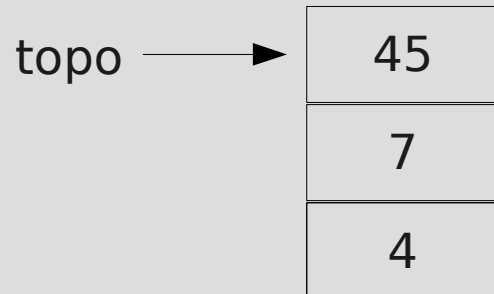
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

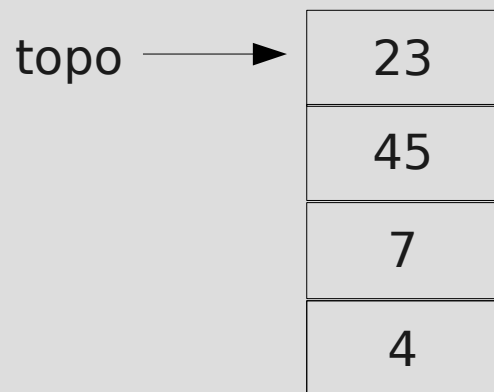
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

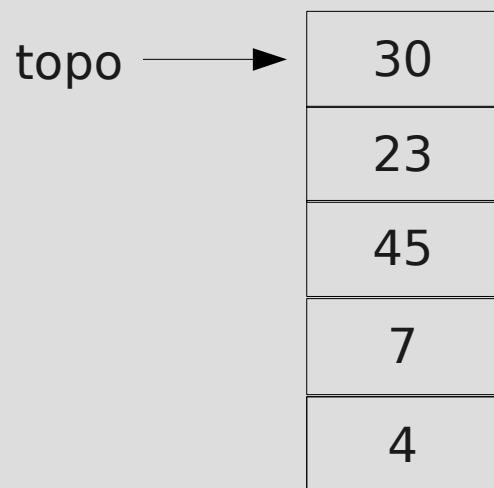
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

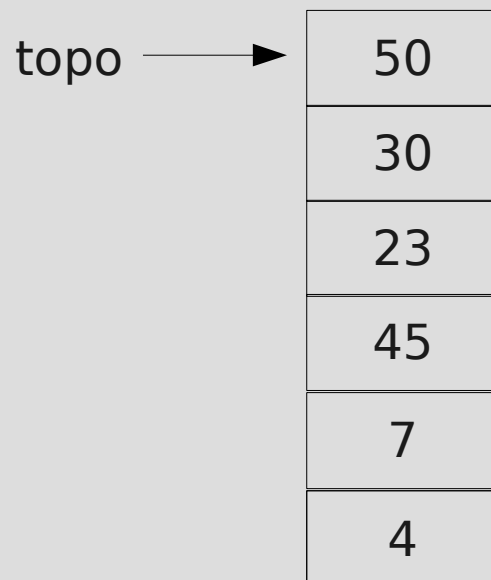
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

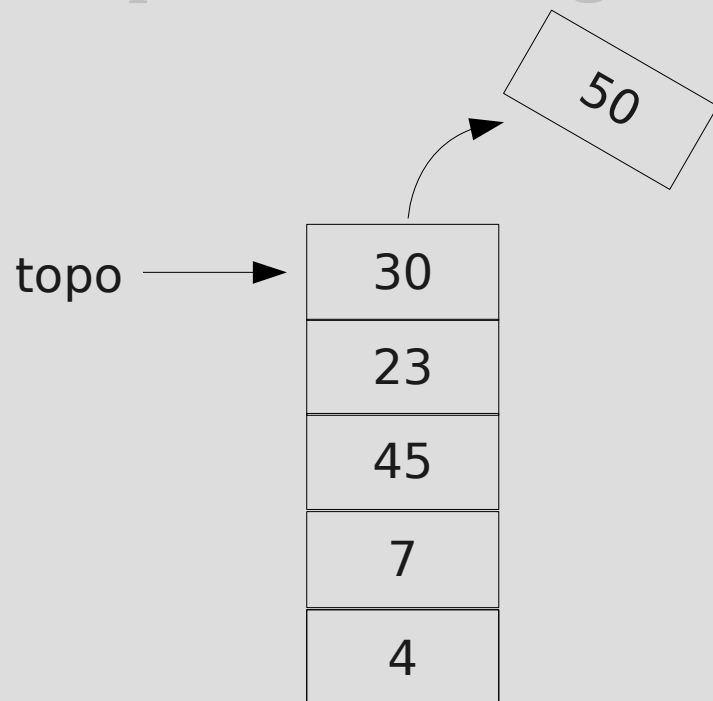
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

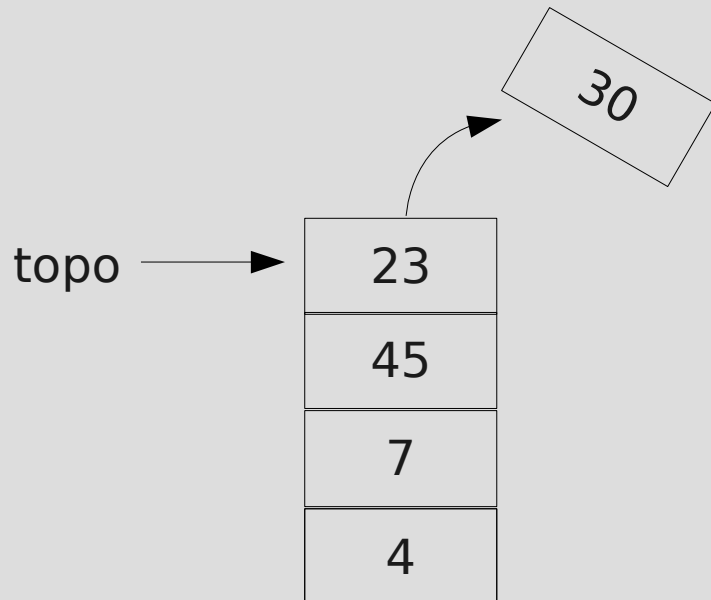
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

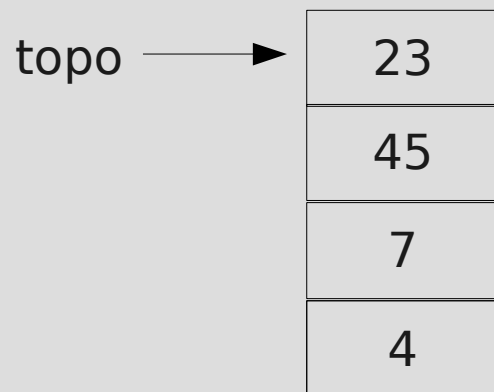
→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

→ *Inserir os elementos 4, 7, 45, 23, 30 e 50 em uma pilha e em seguida remover dois itens*



Pilhas

- *Exemplo*

- *Inverter uma palavra*

- *O usuário fornece uma palavra, os caracteres são empilhados e em seguida exibidos*

Pilhas

Inverter.java

```
public class Inverter {
    private String entrada;
    private String saida;

    public Inverter(String input){
        entrada = input;
    }

    public String fazerInversao(){
        int tamPilha = entrada.length();
        Pilha pilha = new Pilha(tamPilha);

        for (int i=0; i<entrada.length(); i++){
            char character = entrada.charAt(i);
            pilha.push(character);
        }
        saida = "";
    }
}
```



Pilhas

Inverter.java

```
while (!pilha.estaVazia()){
    char character = (char) pilha.pop();
    saida = saida + character;
}
return saida;
}

public static void main(String[] args){
    String entrada, saida;
    Scanner in = new Scanner(System.in);
    String opcao = "S";
    while (!opcao.equalsIgnoreCase("N")){
        System.out.println("Entre com uma string: ");
        entrada = in.next();
        if (entrada.equals(""))
            break;
    }
}
```



Pilhas

Inverter.java

```
Inverter invertida = new Inverter(entrada);
saida = invertida.fazerInversao();
System.out.println("String invertida: " + saida);
System.out.println("Deseja inverter outra palavra? [S/N]");
opcao = in.next();
    }
}
}
```

Pilhas

- *Exemplo*

- *Verificar expressões de fechamentos (“()”; “[]” e “{ }”)*
- *O usuário fornece uma expressão utilizando os delimitadores e o programa verifica se o conteúdo é válido, empilhando os elementos e verificando se os fechamentos se casam*

Pilhas

- *Exemplo*

- *Verificar expressões de fechamentos (“()”; “[]” e “{ }”)*

- *a[b] correto*

- *a{b[c]d}e correto*

- *a{b(c)d}e incorreto*

- *a[b{c}d]e incorreto*

- *a{b(c) incorreto*

Pilhas

- *Exemplo*

→ *Verificar expressões de fechamentos (“()”; “[]” e “{ }”)*

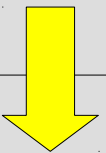
‣ *a{b(c[d]e)f}*

Caracter lido	Conteúdo da pilha
a	
{	{
b	{
({ (
c	{ (
[{ ([
d	{ ([
]	{ (
e	{ (
)	{
f	{
}	

Pilhas

Delimitadores.java

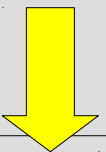
```
public class Delimitadores {  
    private String entrada;  
  
    public Delimitadores(String input){  
        entrada = input;  
    }  
  
    public void verifica(){  
        int tamPilha = entrada.length();  
        Pilha pilha = new Pilha(tamPilha);  
  
        for (int i=0; i<entrada.length(); i++){  
            char caracter = entrada.charAt(i);
```



Pilhas

Delimitadores.java

```
switch(caracter){
    case '{':
    case '[':
    case '(':
        pilha.push(String.valueOf(caracter));
        break;
    case '}':
    case ']':
    case ')':
        if (!pilha.estaVazia()){
            String ch = pilha.pop();
            char chx = ch.charAt(0);
            if ((caracter=='}' && chx != '{') ||
                (caracter==']' && chx != '[') ||
                (caracter==')' && chx != '('))
                System.out.println("Erro: " +caracter+ " em "+i);
        }
        else
            System.out.println("Erro: " +caracter+ " em "+i);
        break;
    default:
        break;
}
```



Pilhas

Delimitadores.java

```
}  
    }  
    if (!pilha.estaVazia()){  
        System.out.println("Erro: Faltando delimitador da direita.");  
    }  
}  
  
public static void main(String[] args){  
    String input;  
    Scanner in = new Scanner(System.in);  
    String opcao = "S";  
    while (!opcao.equalsIgnoreCase("N")){  
        System.out.println("Entre com uma string que possua delimitadores: ");  
        input = in.next();  
  
        Delimitadores sequencia = new Delimitadores(input);  
        sequencia.verifica();  
        System.out.println("Deseja verificar outra sequencia? [S/N]");  
        opcao = in.next();  
    }  
}
```

Roteiro

- *Objetivos*
- *Pilhas*
- ***Exercícios***

Exercícios

- *Suponha que você empilhe os elementos 20, 25, 3, 53, 57 e 8 em uma pilha. Em seguida você remove 3 itens, apresente a pilha final e a ordem em que os elementos são removidos.*
- *Baseado no exercício de inverter palavras dado em sala, escreva um programa que inverta as letras de cada palavra de um texto terminado por ponto (“.”) preservando a ordem das palavras. Por exemplo, dado o texto:
ESTE EXERCICIO EH MUITO FACIL.
a saída deve ser
ETSE OICICREXE HE OTIUM LICAF.*