



Pilhas

Prof. Allan Rodrigo Leite
Estruturas de dados



Estrutura de dados

- Estruturas definem estratégias para
 - Armazenar um conjunto de dados
 - Visitar e recuperar os dados armazenados

- Pilha
 - É uma das estruturas de dados mais simples
 - O acesso aos elementos é realizado pelo topo da pilha
 - Todo elemento novo é introduzido na pilha que, por sua vez, passa a ser o novo topo
 - O único elemento que pode ser acessado ou removido é o elemento no topo



Pilhas

■ Aplicações

- Resolução de parênteses em expressões matemáticas
- Retirada de vagões de uma composição férrea
- Retirada de containers em um navio cargueiro

■ Política de acesso

- Os elementos da pilha são retirados na ordem inversa em que foram introduzidos
- **LIFO** – **L**ast **I**n, **F**irst **O**ut



Pilhas

- Operações de uma pilha

- Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

- Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)

Pilhas

- Operações de uma pilha

- Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

- Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)

Pilha } topo ← NULL

Pilhas

■ Operações de uma pilha

□ Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

□ Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)



Pilhas

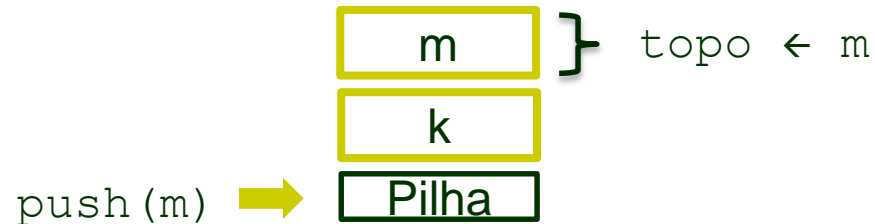
- Operações de uma pilha

- Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

- Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)



Pilhas

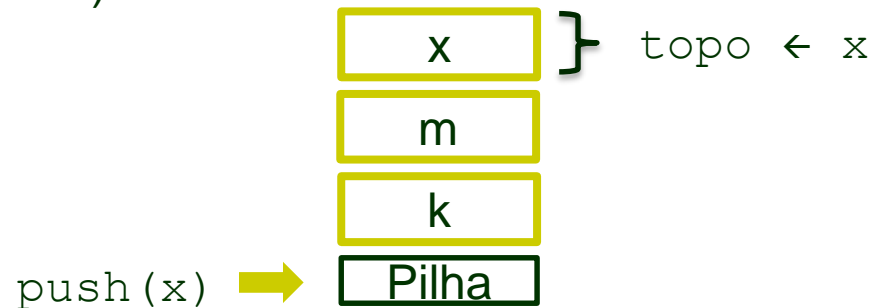
■ Operações de uma pilha

□ Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

□ Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)



Pilhas

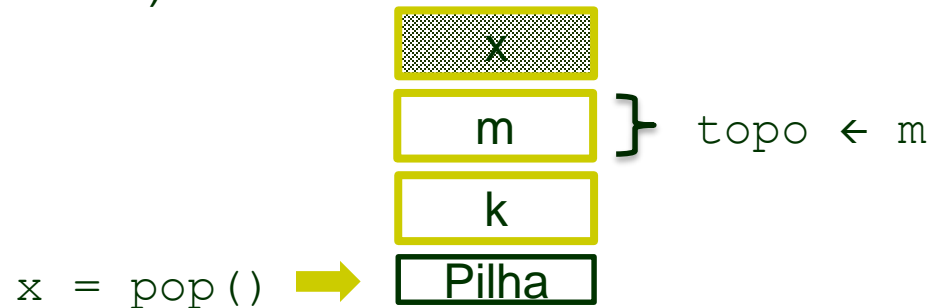
■ Operações de uma pilha

□ Empilhar – **Push**

- Insere um novo elemento, introduzindo-o no topo da pilha

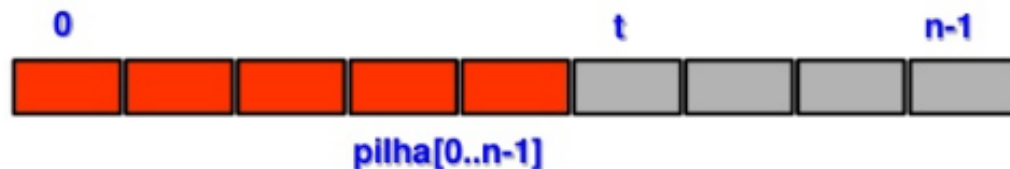
□ Desempilhar – **Pop**

- Remove o elemento contido no topo da pilha, atualizando o topo (quando possível)



Implementação de pilhas estáticas

- Implementação com vetor estático
 - Pilha será armazenada em um vetor chamado `pilha[N]`
 - Onde N é o tamanho máximo da pilha
 - A parte ocupada pela pilha será



Implementação de pilhas estáticas

PilhaEstatica
<ul style="list-style-type: none">- pilha Object[]- topo int
<ul style="list-style-type: none">+ void push(Object o)+ Object pop()+ boolean isFull()+ boolean isEmpty()+ clear()



Implementação de pilhas estáticas

```
public interface PilhaInterface {  
    public void push(Object o) throws Exception;  
    public Object pop() throws Exception;  
  
    public boolean isEmpty();  
    public boolean isFull();  
    public void clear();  
}
```

Implementação de pilhas estáticas

```
public class PilhaEstatica implements PilhaInterface {  
    private Object[] pilha;  
    private int topo;  
  
    public PilhaEstatica(int capacidade) {  
        pilha = new Object[capacidade];  
        topo = 0;  
    }  
  
    ...  
}
```

Implementação de pilhas estáticas

```
public class PilhaEstatica implements PilhaInterface {  
    ...  
  
    public boolean isEmpty() {  
        return topo == 0;  
    }  
  
    public boolean isFull() {  
        return topo == pilha.length;  
    }  
  
    ...  
}
```

Implementação de pilhas estáticas

```
public class PilhaEstatica implements PilhaInterface {  
    ...  
  
    public void push(Object o) throws Exception {  
        if (this.isFull()) {  
            throw new Exception("Pilha cheia");  
        }  
        System.out.println("Adicionando " + o);  
        pilha[topo++] = o; //insere na próxima posição livre  
    }  
  
    ...  
}
```

Implementação de pilhas estáticas

```
public class PilhaEstatica implements PilhaInterface {  
    ...  
    public Object pop() throws Exception {  
        if (this.isEmpty()) {  
            throw new Exception("Pilha vazia");  
        }  
        Object o = pilha[--topo]; //remove elemento  
        pilha[topo] = null;  
        System.out.println("Removendo " + o);  
        return o;  
    }  
    ...  
}
```


Implementação de pilhas estáticas

```
public class PilhaEstatica implements PilhaInterface {  
    ...  
  
    public void clear() {  
        while (!this.isEmpty()) {  
            try {  
                this.pop();  
            } catch (Exception e) { e.printStackTrace(); }  
        }  
    }  
}
```

Implementação de pilhas estáticas

■ Exercício

- Cria uma pilha
- Adicione os seguintes valores
 - 20,0
 - 20,8
 - 20,3
 - 44,5
 - 20,9
- Remova os dois últimos elementos da pilha
- Esvazie a pilha