



# Pilhas Dinâmicas

Prof. Allan Rodrigo Leite  
Estruturas de dados

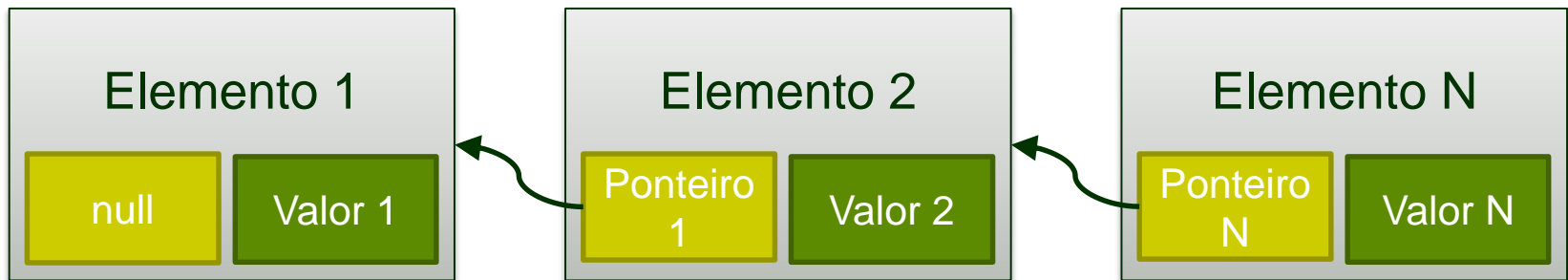


# Pilhas dinâmicas

- Utilizado quando a quantidade máxima de elementos é desconhecida
- Neste caso, é possível implementar uma pilha dinâmica empregando uma lista encadeada
- Como podemos representar esta estrutura?

# Pilhas dinâmicas

- Utilizado quando a quantidade máxima de elementos é desconhecida
- Neste caso, é possível implementar uma pilha dinâmica empregando uma lista encadeada
- Como podemos representar esta estrutura?





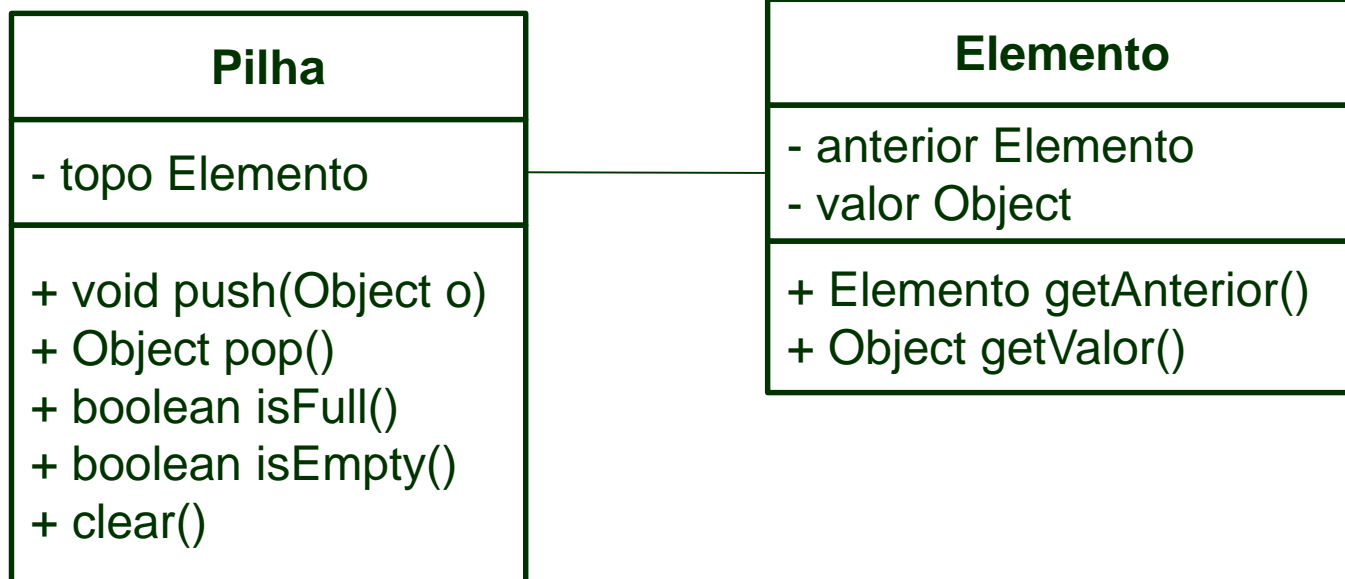
# Implementação de pilhas dinâmicas

## ■ Operações básicas

- ☐ Criar uma pilha
- ☐ Inserir um novo elemento no topo da pilha (push)
- ☐ Remover o elemento no topo da pilha (pop)
- ☐ Verificar se a pilha está vazia
- ☐ Esvaziar a pilha

# Implementação de pilhas dinâmicas

## ■ Estrutura de uma pilha dinâmica





# Implementação de pilhas dinâmicas

```
public class Elemento {  
    private Elemento anterior;  
    private Object valor;  
  
    public Elemento(Object v, Elemento e) {  
        anterior = e;  
        valor = v;  
    }  
  
    public Elemento getAnterior() { return anterior; }  
    public Object getValor() { return valor; }  
}
```

# Implementação de pilhas dinâmicas

```
public class PilhaDinamica implements PilhaInterface {  
    private Elemento topo;  
  
    public boolean isEmpty() {  
        return topo == null;  
    }  
  
    public boolean isFull() {  
        return false;  
    }  
  
    ...  
}
```



# Implementação de pilhas dinâmicas

```
public class PilhaDinamica implements PilhaInterface {  
    ...  
  
    public void push(Object o) throws Exception {  
        System.out.println("Adicionando " + o);  
        topo = new Elemento(o, topo);  
    }  
  
    ...  
}
```



# Implementação de pilhas dinâmicas

```
public class PilhaDinamica implements PilhaInterface {  
    ...  
    public Object pop() throws Exception {  
        if (this.isEmpty()) {  
            throw new Exception("Pilha vazia");  
        }  
        Object o = topo.getValor();  
        topo = topo.getAnterior();  
        System.out.println("Removendo " + o);  
        return o;  
    }  
    ...  
}
```

# Implementação de pilhas dinâmicas

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

# Percorrendo uma pilha

## ■ Estática

```
public void percorrer() {  
    for (int i = topo - 1; i >= 0; i++) {  
        System.out.println("Valor: " + pilha[i]);  
    }  
}
```

## ■ Dinâmica

```
public void percorrer() {  
    Elemento e = topo;  
    while (e != null) {  
        System.out.println("Valor: " + e.getValor());  
        e = e.getAnterior();  
    }  
}
```

# Exercícios

- Utilizando as pilhas estáticas e dinâmicas implementadas, faça:
  - Um programa que insira dados em uma pilha A e em seguida remova os elementos da pilha A e insira-os na pilha B em ordem inversa
  - Um programa que valida erros gramaticais em parênteses de expressões matemáticas
    - $(A + B) * C) / D$  //errado
    - $((A + B) * C) / D$  //correto
  - Um programa que verifica se uma palavra é um palíndromo
    - ANA
    - ARARA
    - MATAM
    - SOCORRAM-ME, SUBI NO ÔNIBUS EM MARROCOS
    - LUZA ROCELINA, A NAMORADA DO MANUEL, LEU NA MODA DA ROMANA: ANIL É COR AZUL

# Exercícios

- Utilizando as pilhas estáticas e dinâmicas implementadas, faça:
  - Um algoritmo para gerenciar as vagas em um estacionamento, onde os veículos estacionados são controlados pelas placas. A retirada dos veículos deve ocorrer na ordem inversa que os veículos são estacionados e o estacionamento deve possuir um limite de veículos. Exemplo:
    - Capacidade do estacionamento: 5 veículos
    - Adiciona MIX-7891
    - Adiciona MAD-1238
    - Adiciona QHA-9492
    - Remove QHA-9492
    - Exibe veículos estacionados: MIX-7891 e MAD-1238
  - Um algoritmo que simule a pilha de execução de funções recursivas, considerando uma função que calcule o fatorial de um dado número. A cada recursão (elemento da pilha), deve ser armazenado o resultado do fatorial até o momento. Exemplo:
    - $5! = [1! = 1] [2! = 2] [3! = 6] [4! = 24] [5! = 120]$