

# Estrutura de Dados e Algoritmos com Java

## PILHAS (STACKS)

<loiane.training />



12

# Estrutura de Dados e Algoritmos com Java

<loiane.training />

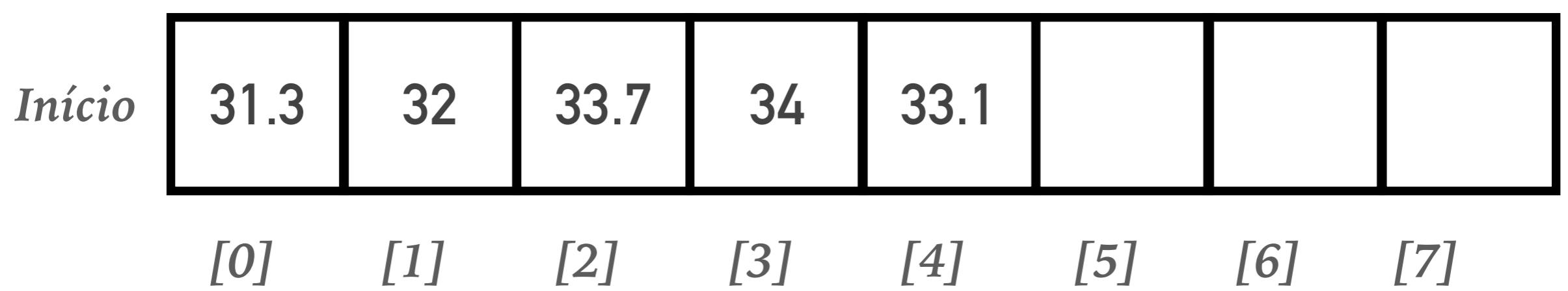
## PILHAS (STACKS)

*Introdução*



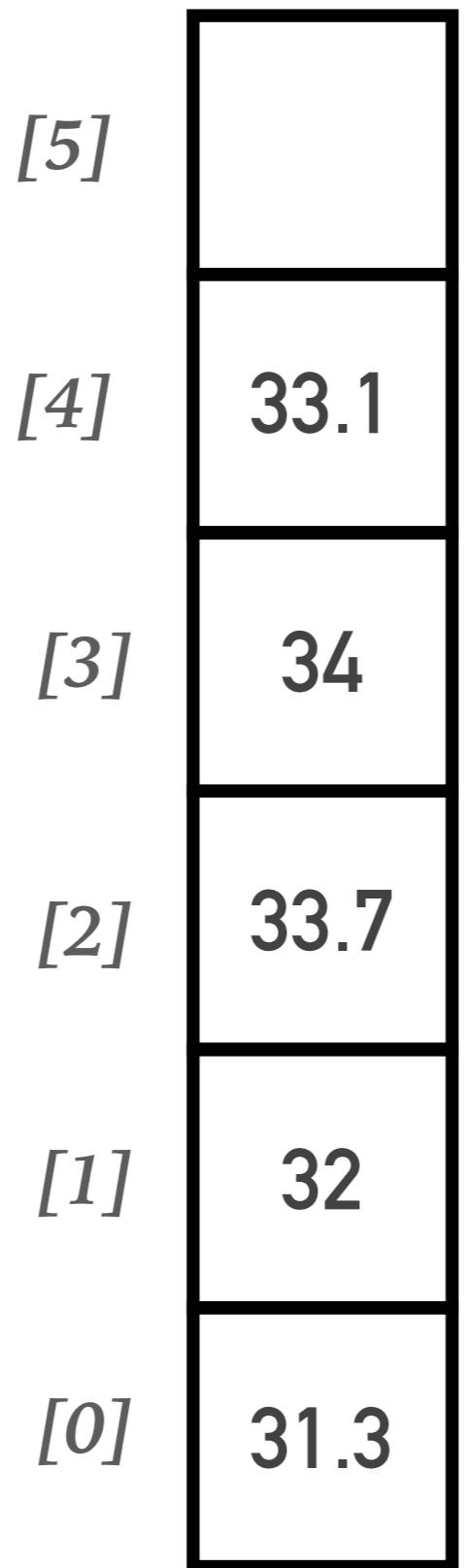


# *Vetor, Array, Lista*



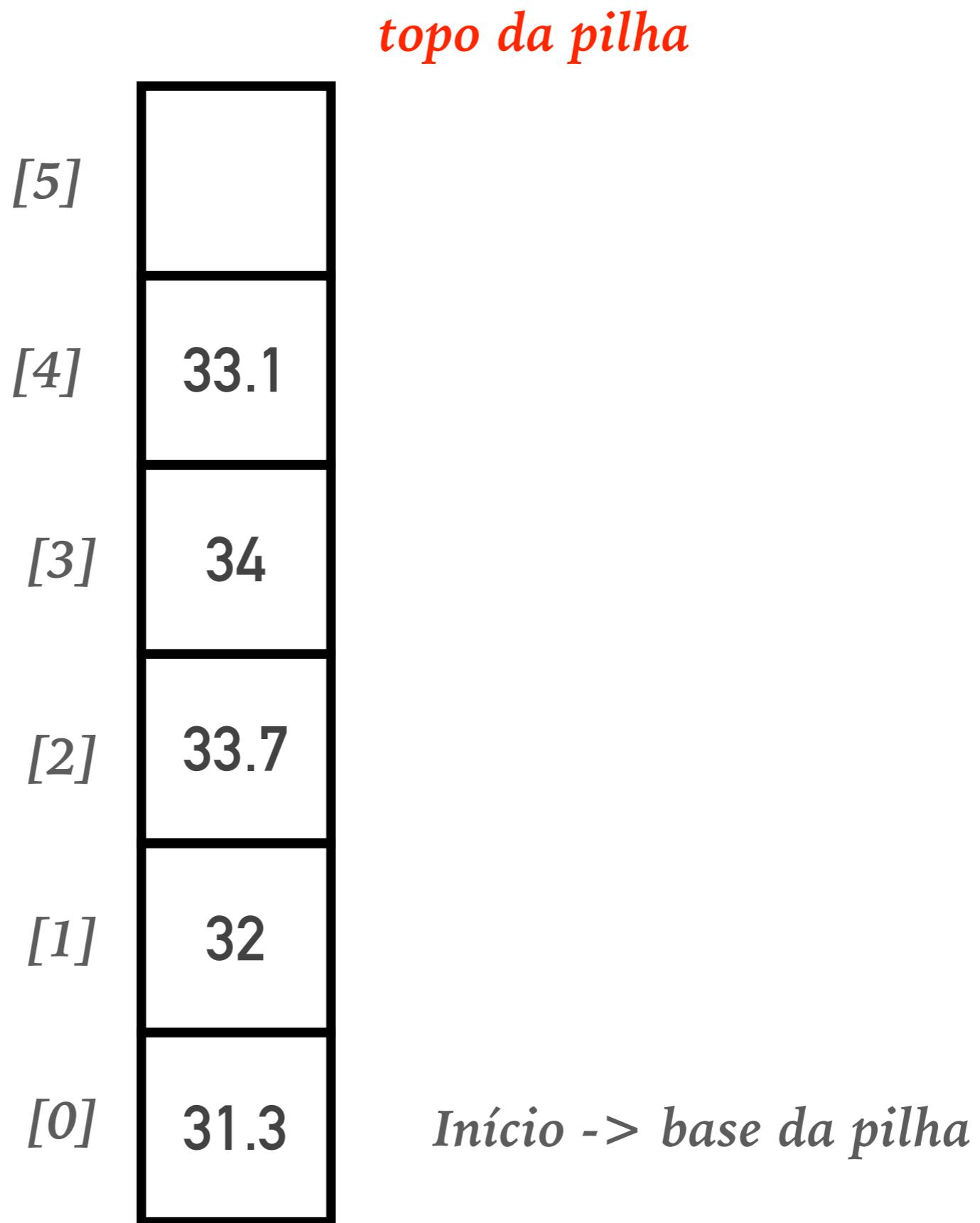
*tamanho = 5*

# Pilha



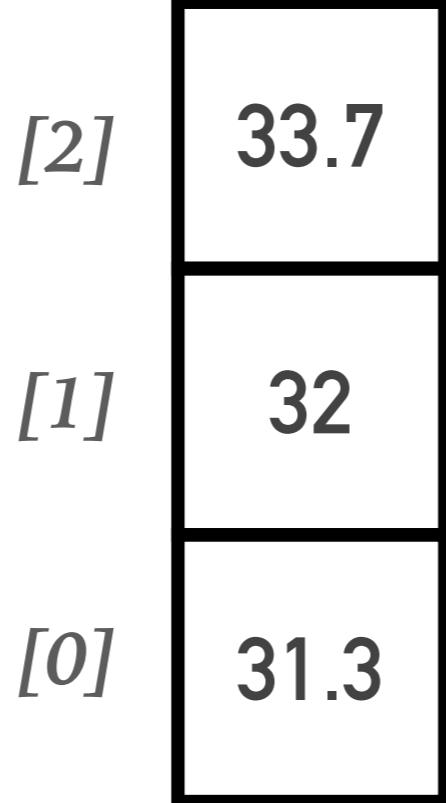
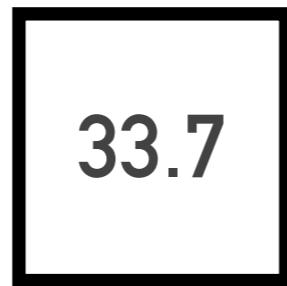
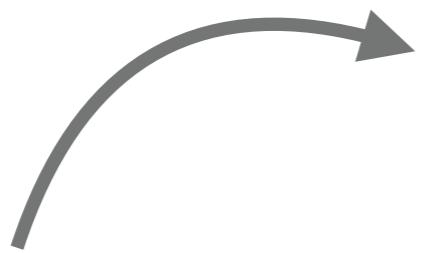
*Início -> base da pilha*

# Pilha



# Pilha

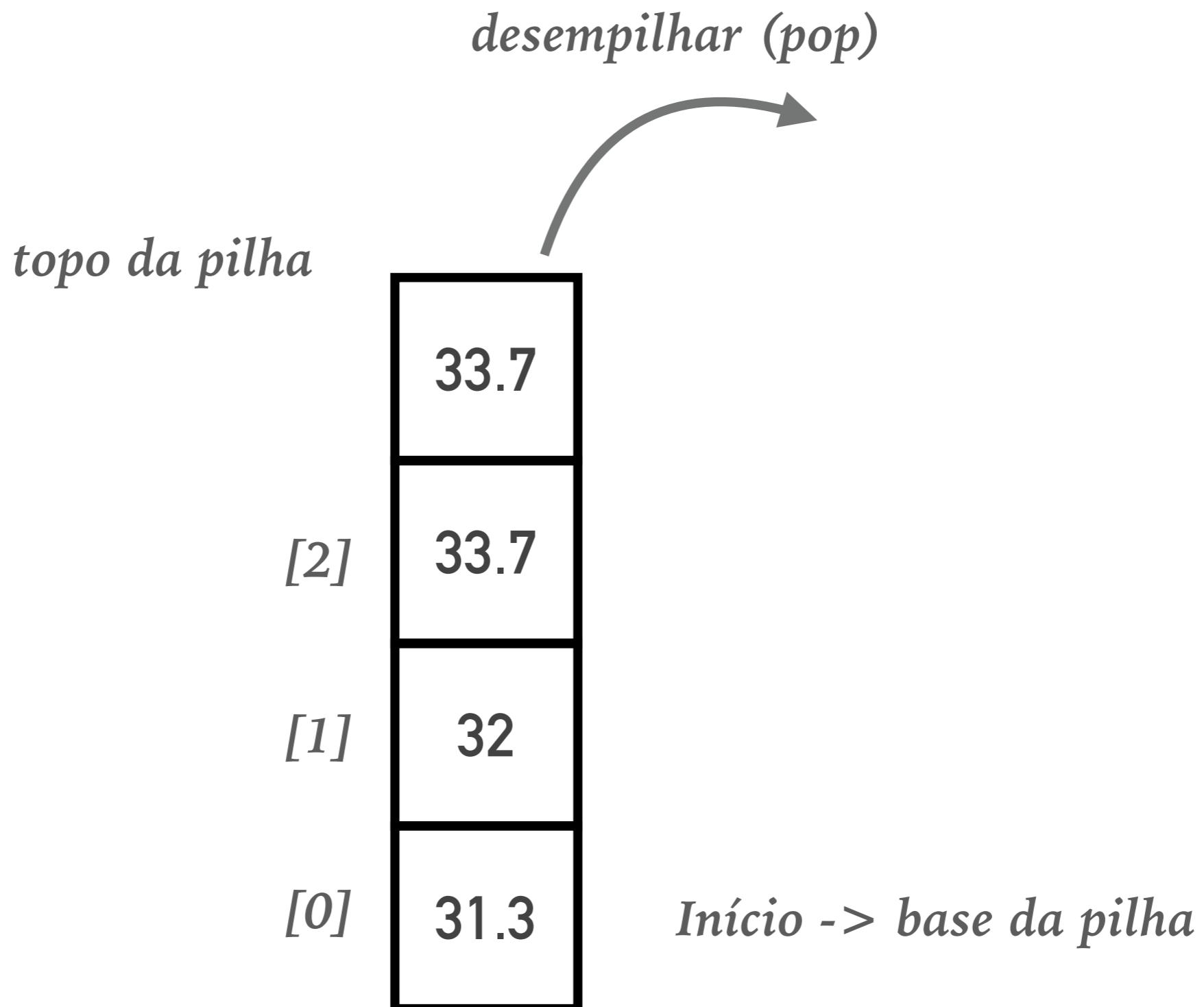
*empilhar (push)*



*topo da pilha*

*Início -> base da pilha*

# Pilha



**LIFO**

# LIFO

*Last In First Out*

# LIFO

*Last In First Out*

Último a entrar, primeiro a sair

# PILHAS (STACKS): INTRODUÇÃO

---

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
- Definição

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento
  - Verificar se pilha está vazia

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha
  - API Java Stack

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição
  - Verificar quantidade de elementos na pilha
  - Empilhar elemento
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha
  - API Java Stack
  - Exercícios

# DIAGRAMA CLASSE PILHA

---

C	Pilha	
f	elementos	T[]
f	tamanho	int
m	Pilha(int)	
m	empilha(T)	void
m	aumentaCapacidade()	void
m	estaVazia()	boolean
m	topo()	T
m	desempilha()	T
m	tamanho()	int
m	toString()	String

# DIAGRAMA CLASSE PILHA

---

C Pilha	
f elementos	T[]
f tamanho	int
m Pilha(int)	
m empilha(T)	void
m aumentaCapacidade()	void
m estaVazia()	boolean
m topo()	T
m desempilha()	T
m tamanho()	int
m toString()	String

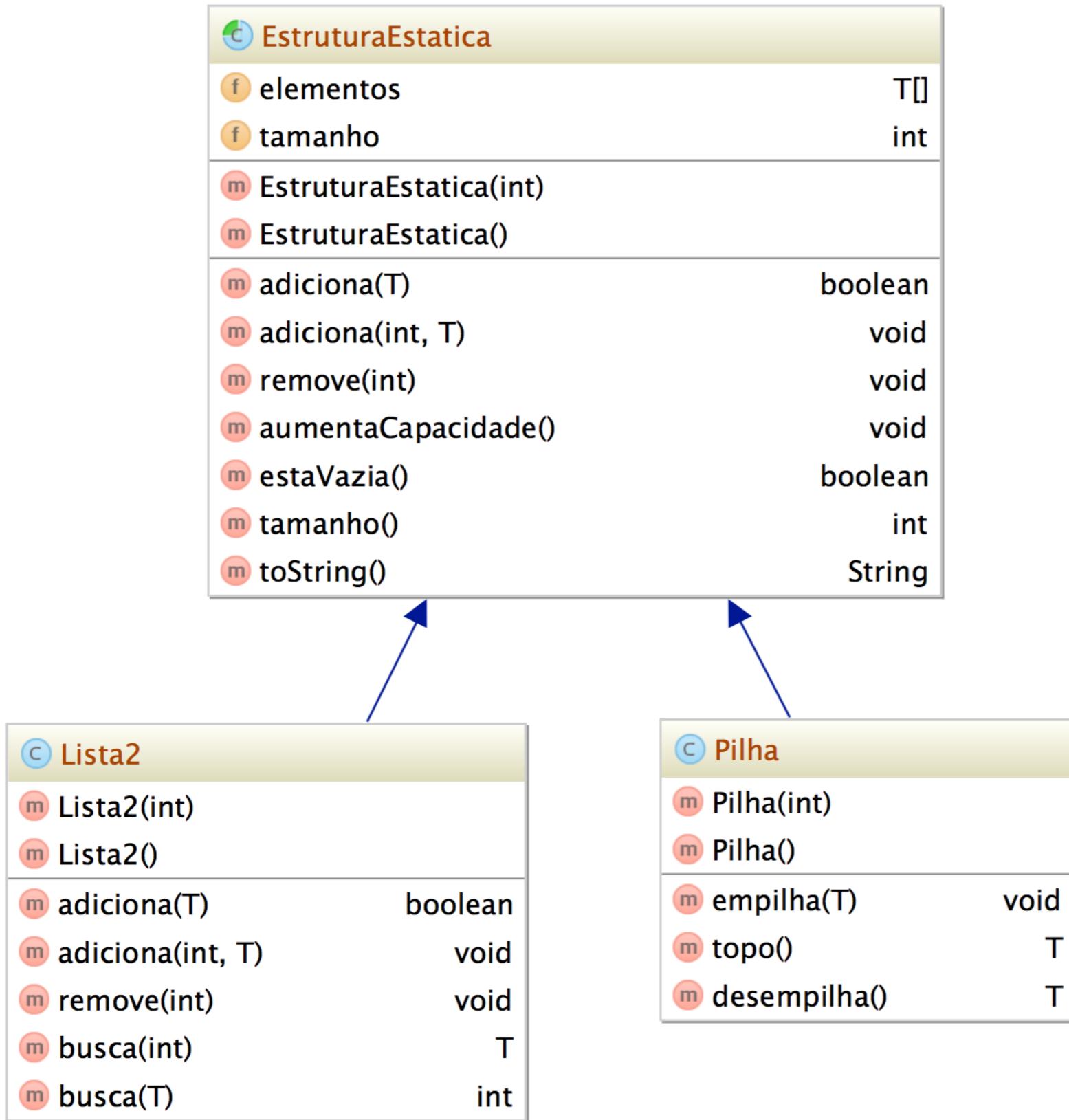
# CLASSE PILHA

---

```
public class Pilha<T> {  
  
    private T[] elementos;  
    private int tamanho;  
  
    public Pilha(int capacidade) {  
        elementos = (T[]) new Object[capacidade];  
        tamanho = 0;  
    }  
  
    public int tamanho(){  
        return this.tamanho;  
    }  
  
    public String toString(){  
        //mesmo código que a classe Lista/Vetor  
    }  
}
```

# VAMOS REFATORAR?

---



# CLASSE REFATORADA

---

```
public class Pilha<T> extends EstruturaEstatica<T>{  
  
    public Pilha(int capacidade) {  
        super(capacidade);  
    }  
  
    public Pilha() {  
        super();  
    }  
}
```

# TESTE

---

```
public class Aula13 {  
    public static void main(String[] args) {  
        Pilha<Integer> pilha = new Pilha<Integer>(5);  
        System.out.println(pilha.tamanho());  
        System.out.println(pilha);  
    }  
}
```

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha
  - API Java Stack
  - Exercícios

14

# Estrutura de Dados e Algoritmos com Java

<loiane.training />

## PILHAS (STACKS)

*Empilhar elemento*

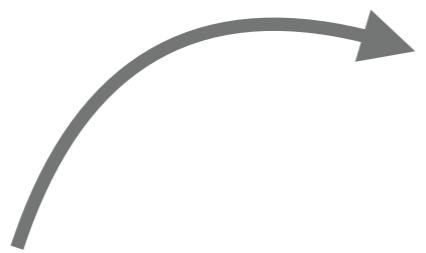
# LIFO

*Last In First Out*

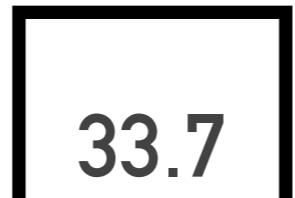
Último a entrar, primeiro a sair

# Pilha

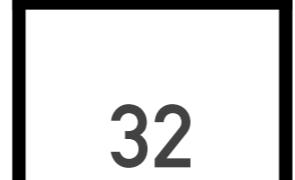
*empilhar (push)*



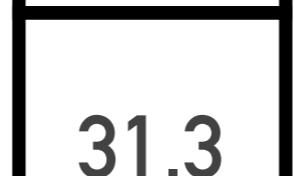
[2]



[1]



[0]



*topo da pilha*

*Início -> base da pilha*

# EMPILHAR (ADICIONAR ELEMENTO)

---

```
public void empilha(T elemento) {  
    aumentaCapacidade();  
    elementos[tamanho] = elemento;  
    tamanho++;  
    //elementos[tamanho++] = elemento;  
}
```

# EMPILHAR (ADICIONAR ELEMENTO) - REUSANDO CÓDIGO

---

```
public void empilha(T elemento) {  
    super.adiciona(elemento);  
}
```

# TESTE

---

```
Pilha<Integer> pilha = new Pilha<Integer>(5);

for (int i=1;i<10; i++){
    pilha.empilha(i);
}

System.out.println(pilha.tamanho());
System.out.println(pilha);
```

# STACK OVERFLOW EXCEPTION

---



# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento 
  - Verificar se pilha está vazia
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha
  - API Java Stack
  - Exercícios

18

# Estrutura de Dados e Algoritmos com Java

<loiane.training />

## PILHAS (STACKS)

*Verificar pilha vazia*

# VERIFICAR PILHA VAZIA

---

```
public boolean estaVazia(){
    return tamanho == 0;
}
```

# TESTE

---

```
Pilha<Integer> pilha = new Pilha<Integer>(5);

System.out.println(pilha.estaVazia());

for (int i=1;i<10; i++){
    pilha.empilha(i);
}

System.out.println(pilha.estaVazia());
System.out.println(pilha.tamanho());
System.out.println(pilha);
```

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento 
  - Verificar se pilha está vazia 
  - Espiar/Verificar elemento topo da pilha
  - Desempilhar elemento da pilha
  - API Java Stack
  - Exercícios

16

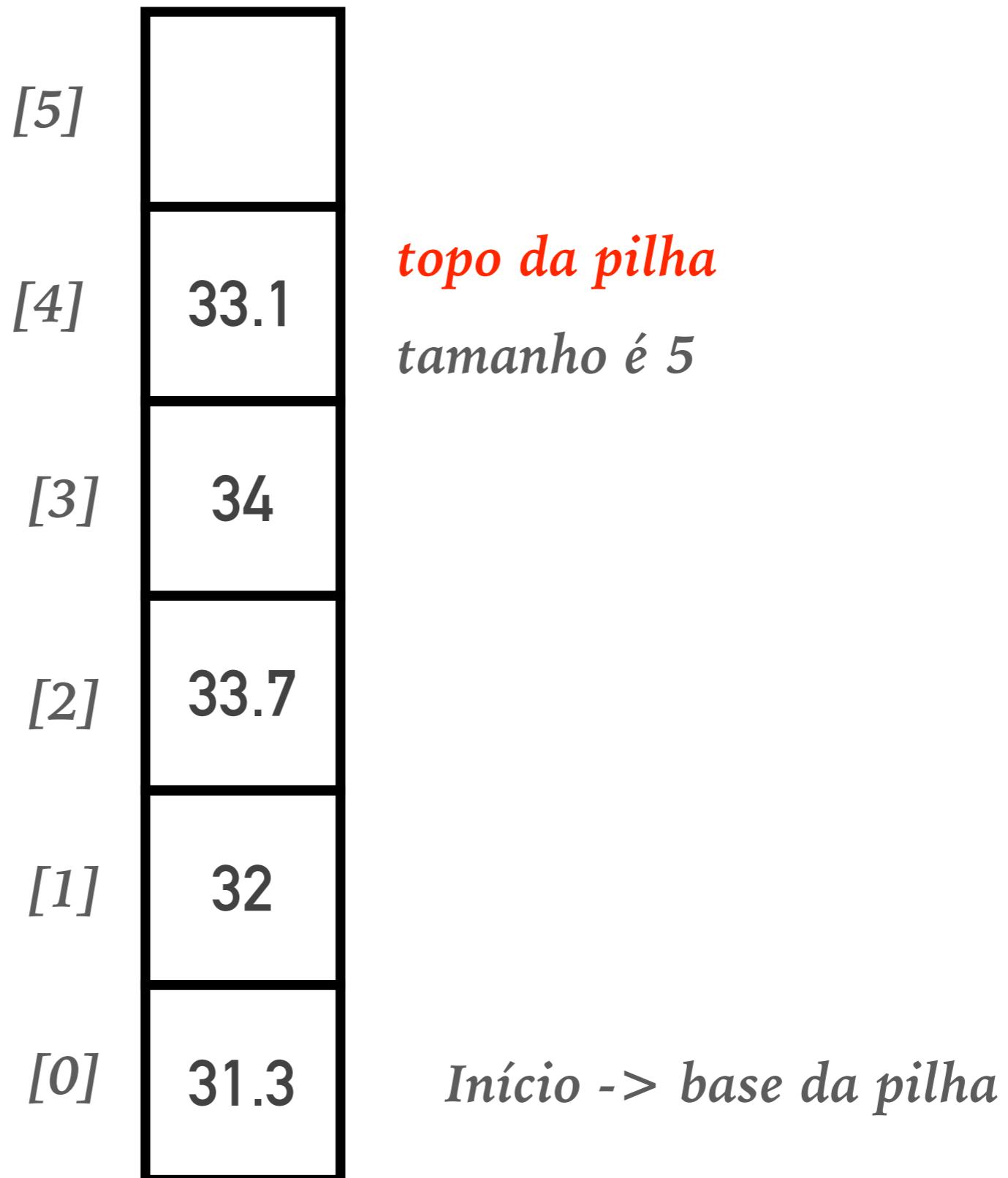
# Estrutura de Dados e Algoritmos com Java

<loiane.training />

## PILHAS (STACKS)

*Espiar topo da pilha*

# Pilha



# ESPIAR TOPO DA PILHA (PEEK)

---

```
public T topo(){
    if (this.estaVazia())){
        return null;
    }
    return elementos[tamanho-1];
}
```

# TESTE

---

```
Pilha<Integer> pilha = new Pilha<Integer>(5);

System.out.println(pilha.estaVazia());

for (int i=1;i<10; i++){
    pilha.empilha(i);
}

System.out.println(pilha.estaVazia());
System.out.println(pilha.tamanho());
System.out.println(pilha);

System.out.println(pilha.topo());
```

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento 
  - Verificar se pilha está vazia 
  - Espiar/Verificar elemento topo da pilha 
  - Desempilhar elemento da pilha
  - API Java Stack
  - Exercícios

17

# Estrutura de Dados e Algoritmos com Java

<loiane.training />

## PILHAS (STACKS)

*Desempilhar elemento*

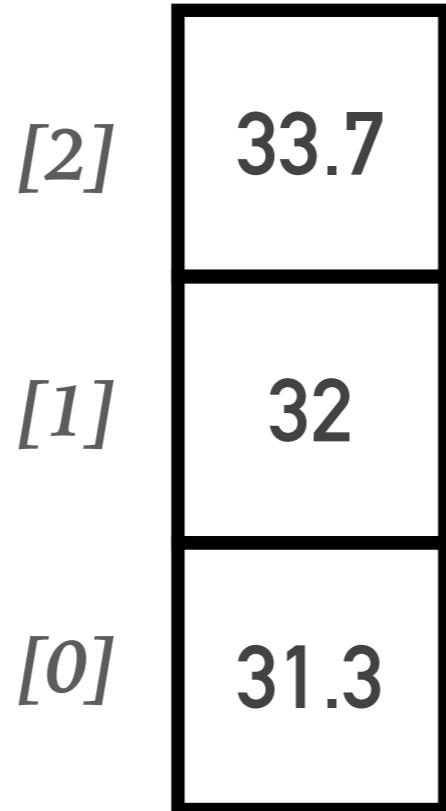
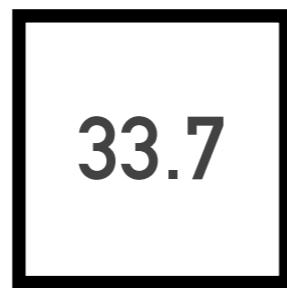
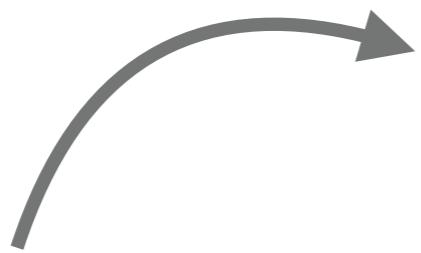
# LIFO

*Last In First Out*

Último a entrar, primeiro a sair

# Pilha

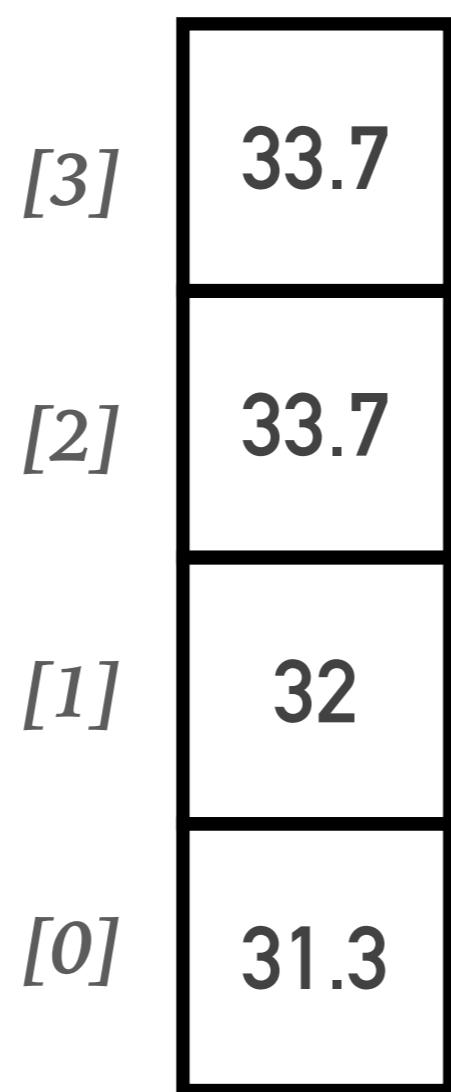
*empilhar (push)*



*topo da pilha*

*Início -> base da pilha*

# Pilha

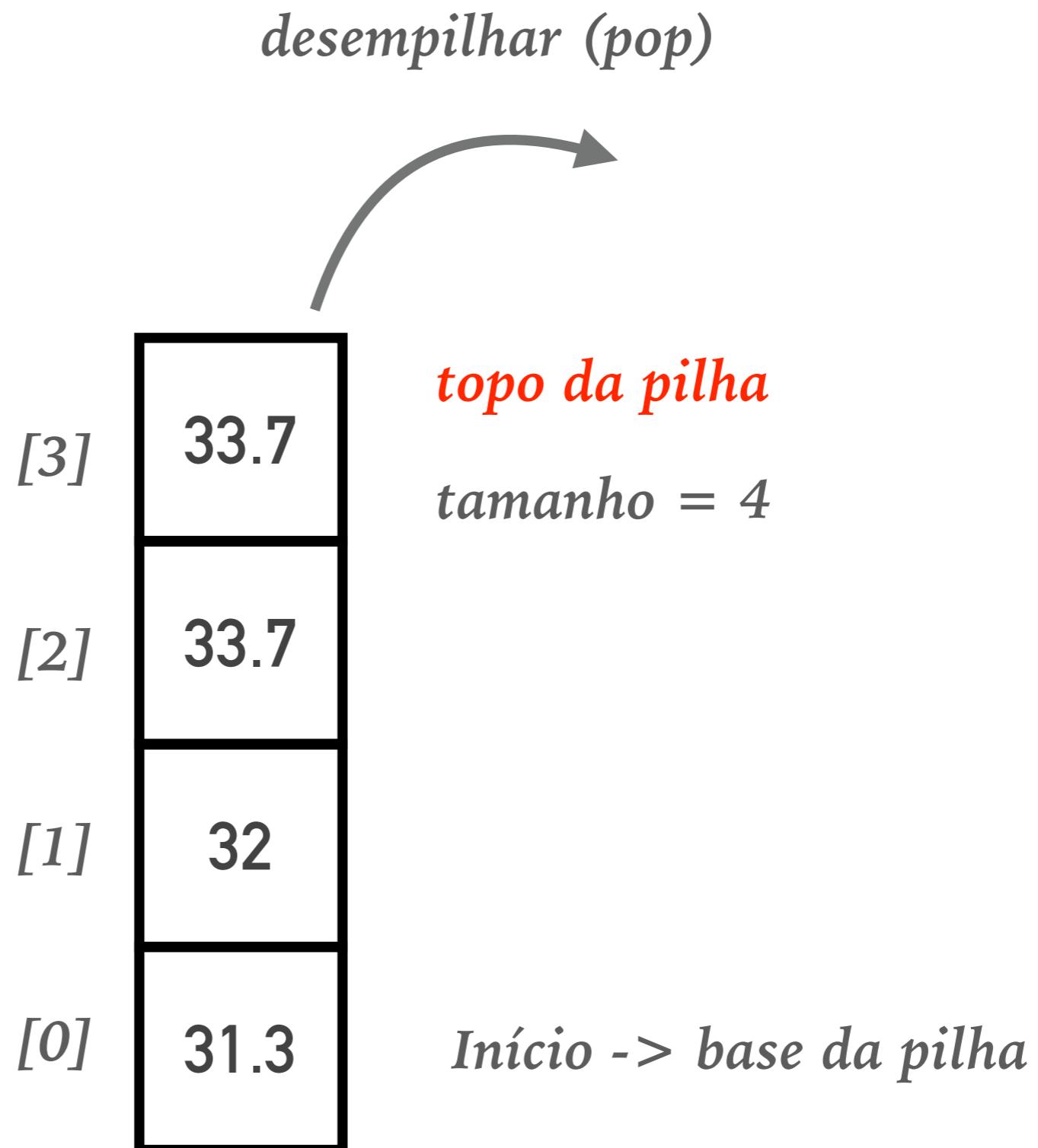


*topo da pilha*

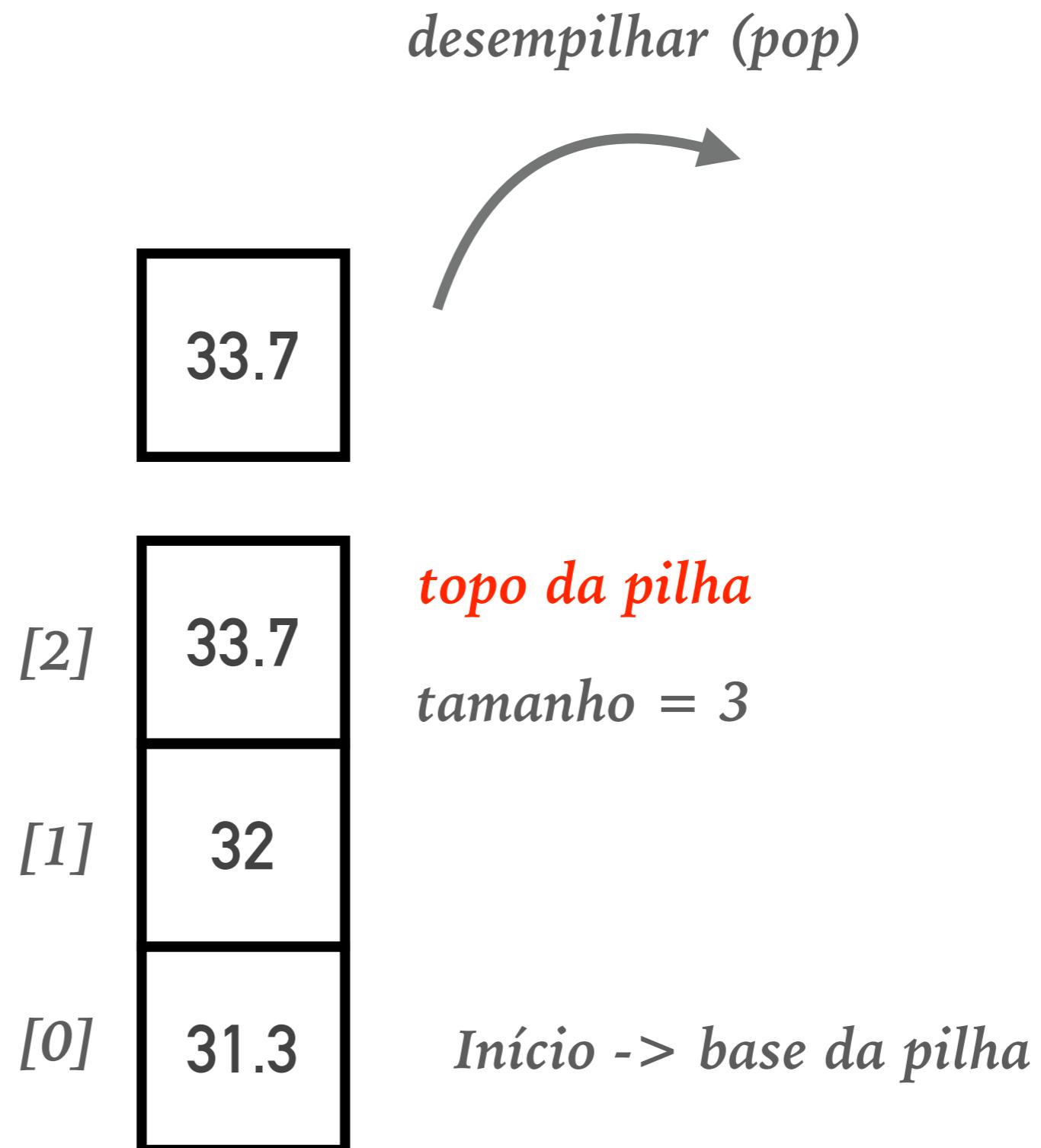
*tamanho = 4*

*Início -> base da pilha*

# Pilha



# Pilha



# REMOVER ELEMENTO DA PILHA

---

```
public T desempilha(){
    if (this.estaVazia()){
        return null;
    }
    return elementos[--tamanho];
}
```

# TESTE

---

```
Pilha<Integer> pilha = new Pilha<Integer>();  
  
System.out.println(pilha.estaVazia());  
  
for (int i=1;i<10; i++){  
    pilha.empilha(i);  
}  
  
System.out.println(pilha.estaVazia());  
System.out.println(pilha.tamanho());  
System.out.println(pilha);  
  
System.out.println(pilha.topo());  
  
int elemento = pilha.desempilha();  
System.out.println(elemento);  
System.out.println(pilha.tamanho());  
System.out.println(pilha);  
System.out.println(pilha.topo());
```

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento 
  - Verificar se pilha está vazia 
  - Espiar/Verificar elemento topo da pilha 
  - Desempilhar elemento da pilha 
- API Java Stack
- Exercícios



18

# Estrutura de Dados e Algoritmos com Java

<loiane.training />

## PILHAS (STACKS)

*API Java Stack*

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();  
  
System.out.println(pilha.isEmpty());  
  
for (int i=1;i<10; i++){  
    pilha.push(i);  
}  
  
System.out.println(pilha.isEmpty());  
System.out.println(pilha.size());  
System.out.println(pilha);  
  
System.out.println(pilha.peek());  
  
int elemento = pilha.pop();  
System.out.println(elemento);  
System.out.println(pilha.size());  
System.out.println(pilha);  
System.out.println(pilha.peek());
```

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();  
  
System.out.println(pilha.isEmpty()); ➔ estaVazia  
  
for (int i=1;i<10; i++){  
    pilha.push(i);  
}  
  
System.out.println(pilha.isEmpty());  
System.out.println(pilha.size());  
System.out.println(pilha);  
  
System.out.println(pilha.peek());  
  
int elemento = pilha.pop();  
System.out.println(elemento);  
System.out.println(pilha.size());  
System.out.println(pilha);  
System.out.println(pilha.peek());
```

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();
```

```
System.out.println(pilha.isEmpty());
```

```
for (int i=1;i<10; i++){
    pilha.push(i); → empilha
}
```

```
System.out.println(pilha.isEmpty());
```

```
System.out.println(pilha.size());
```

```
System.out.println(pilha);
```

```
System.out.println(pilha.peek());
```

```
int elemento = pilha.pop();
System.out.println(elemento);
System.out.println(pilha.size());
System.out.println(pilha);
System.out.println(pilha.peek());
```

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();  
  
System.out.println(pilha.isEmpty());  
  
for (int i=1;i<10; i++){  
    pilha.push(i);  
}  
  
System.out.println(pilha.isEmpty());  
System.out.println(pilha.size()); → tamanho  
System.out.println(pilha);  
  
System.out.println(pilha.peek());  
  
int elemento = pilha.pop();  
System.out.println(elemento);  
System.out.println(pilha.size());  
System.out.println(pilha);  
System.out.println(pilha.peek());
```

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();
```

```
System.out.println(pilha.isEmpty());
```

```
for (int i=1;i<10; i++){
    pilha.push(i);
}
```

```
System.out.println(pilha.isEmpty());
```

```
System.out.println(pilha.size());
```

```
System.out.println(pilha);
```

**System.out.println(pilha.peek()); → topo**

```
int elemento = pilha.pop();
```

```
System.out.println(elemento);
```

```
System.out.println(pilha.size());
```

```
System.out.println(pilha);
```

```
System.out.println(pilha.peek());
```

# CLASSE STACK - API JAVA

---

```
Stack<Integer> pilha = new Stack<Integer>();  
  
System.out.println(pilha.isEmpty());  
  
for (int i=1;i<10; i++){  
    pilha.push(i);  
}  
  
System.out.println(pilha.isEmpty());  
System.out.println(pilha.size());  
System.out.println(pilha);  
  
System.out.println(pilha.peek());  
  
int elemento = pilha.pop(); ➔ desempilha  
System.out.println(elemento);  
System.out.println(pilha.size());  
System.out.println(pilha);  
System.out.println(pilha.peek());
```

# PILHAS (STACKS): INTRODUÇÃO

---

- Classe Pilha
  - Definição 
  - Verificar quantidade de elementos na pilha 
  - Empilhar elemento 
  - Verificar se pilha está vazia 
  - Espiar/Verificar elemento topo da pilha 
  - Desempilhar elemento da pilha 
  - API Java Stack 
  - Exercícios



## EXERCÍCIOS

*<http://goo.gl/Ad5lAe>*

GRÁTIS

# Estrutura de Dados e Algoritmos com Java

Estrutura de Dados

## Estrutura de Dados e Algoritmos com Java

INICIAR CURSO

HOME

### O QUE VAMOS APRENDER NESSE CURSO?

- Vetores (Arrays)
- Pilhas (Stacks)
- Filas (Queues)
- Listas Encadeadas (Linked Lists)
- Listas Duplamente Encadeadas (Doubly-Linked Lists)
- Conjuntos (Sets)
- Tabelas de Hashing (HashTables)
- Árvores (Trees)
- Grafos (Graphs)
- Algoritmos de Ordenação:
  - Bolha (Bubble Sort)

GRÁTIS



\* REQUER JAVA BÁSICO



ACESSO ILIMITADO



CERTIFICADO DO CURSO



Download código fonte e certificado  
Cadastro em:

<http://loiane.training>

obrigada



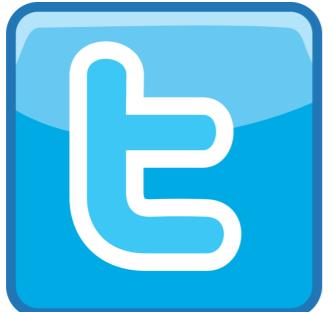
<http://loiane.training>



<http://loiane.com>



<facebook.com/loianegroner>



<twitter.com/loiane>



<https://github.com/loiane>



<youtube.com/loianegroner>