



SÃO  
PAULO  
TECH  
SCHOOL

# **Estrutura de Dados**

## **Pilha**

**Célia Taniwaki**

# Tópicos da Aula

- O que é pilha
- Operações da pilha
- Classe Stack
- Implementando a pilha

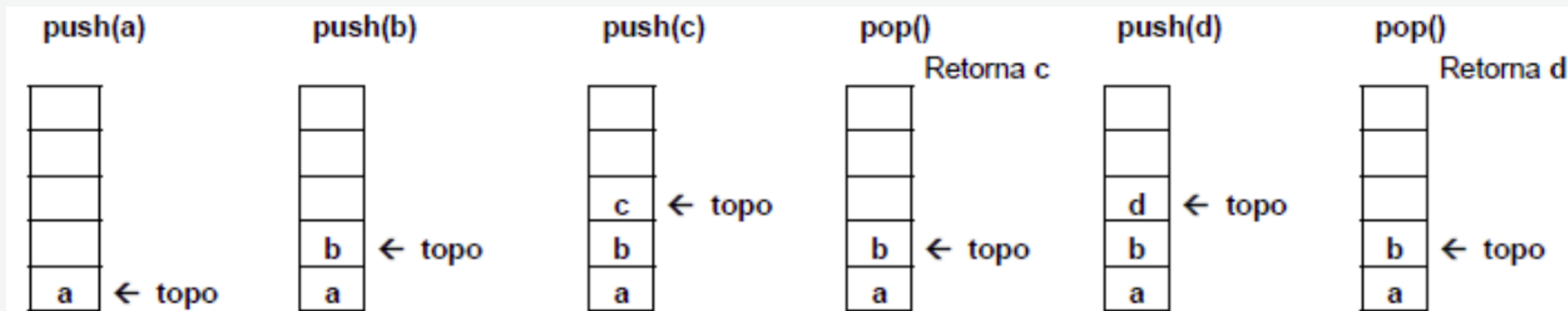
# Pilha

Estrutura de Dados caracterizada por:

- Armazenar elementos do **mesmo tipo (em linguagens tipadas)**
- A inserção e a remoção de elementos é sempre feita pelo **topo da pilha** (fim da sequência).  
(Analogia com pilha de livros ou prato)
- Novo elemento inserido passa a ser o **topo da pilha**
- Elemento a ser **removido** é o que está no **topo**
- **LIFO** (*Last-In First-Out*) - o último a entrar será o primeiro

# Operações na pilha

- Empilhar ou **Push**
  - **Inserir** um elemento no topo da pilha
- Desempilhar ou **Pop**
  - **Remover** um elemento do topo da pilha



# Classe Stack

Suponha uma pilha inicialmente vazia e simule a execução das operações, desenhando o conteúdo da pilha a cada operação e anotando o que é desempilhado:

```
push(5)  
push(7)  
push(2)  
pop()  
pop()  
push(14)  
push(11)  
push(21)  
push(3)  
pop()  
pop()  
pop()  
push(3)
```

# Classe Stack

Java fornece a classe **Stack** do pacote **java.util**

- Essa classe é herdeira da classe Vector do Java
- Forma de instanciar a classe Stack é semelhante ao ArrayList:

*Stack<tipo> pilha = new Stack<tipo>();*

Métodos da operação pilha:

- **isEmpty()** - retorna *true* se a pilha está vazia, *false* caso contrario
- **push(elemento)** - empilha elemento na pilha
- **pop()** - desempilha e retorna o elemento do topo da pilha
- **peek()** - retorna o elemento do topo da pilha, sem desempilhar

# Implementação da pilha

A pilha pode ser implementada através de **Vetores**:

- Nesse caso, a pilha será armazenada num vetor **pilha** de capacidade **N** = `pilha.length`.
- A variável **topo** indica o índice do topo da pilha, assumindo -1 caso a pilha esteja vazia, e N-1 caso a pilha esteja cheia, sendo  $N = \text{pilha.length}$ .



The background is a dark, almost black, space filled with intricate, glowing patterns. These patterns consist of numerous thin, curved lines that flow and swirl across the frame, creating a sense of movement and depth. Interspersed among these lines are many small, bright white dots and larger, fainter circular bokeh-like shapes, which contribute to a cosmic or digital aesthetic. The overall effect is one of a vast, dynamic, and ethereal environment.

# Fazendo a classe Pilha

# Implementação da classe Pilha

- Criar um projeto chamado **exemplo-pilha**
- Criar uma classe chamada Pilha
- Implementar a classe pilha como vetor usando:
  - Atributos (encapsulados):
    - **int topo** // contém o índice do topo da pilha
    - **int[ ] pilha** //vetor que representa a pilha
  - Construtor, que recebe a **capacidade** da pilha, cria o vetor para a pilha, com o tamanho igual à capacidade e inicializa o **topo** com **-1**

# Implementação da classe Pilha

## Métodos:

- **boolean isEmpty()** - Devolve true caso a pilha esteja vazia
- **boolean isFull()** - Devolve true caso a pilha esteja cheia
- **void push (int info)** - Se a pilha não está cheia, então incrementa o topo e atribui info para pilha[topo]
- **int pop()** - Se a pilha não está vazia, então devolve elemento de pilha[topo] e decrementa topo, senão retorna -1
- **int peek()** - se pilha não está vazia, então devolve elemento de pilha[topo], senão devolve -1
- **void exibe()** - Se a pilha está vazia, exibe "Pilha vazia", senão exibe os elementos da pilha

## Teste a classe Pilha

- Crie nesse mesmo projeto a classe **Teste**, configurando para que essa classe tenha o método **main**:
  - Crie um objeto da classe **Pilha**
  - **Empilhe** valores na pilha
  - **Desempilhe** um por um e **exiba-os**

Você verá que eles não exibidos na ordem inversa da que são empilhados.

# Aplicação da pilha

Quando devo usar pilha para armazenar dados?

- Quando desejo usar o dados na ordem inversa da inserção

## **Exemplo;**

- Ação de **desfazer** dos editores: a última ação realizada é a primeira a ser desfeita
- Conversão de número **decimal para binário**: utilizando o método da divisão - dividimos o número decimal por 2, sucessivamente, até que o resultado seja 0.

O número binário é formado pelo resto da divisão, considerando-os de trás para frente

**Agradeço**  
a sua atenção!

**Célia Taniwaki**

**[celia.taniwaki@sptech.school](mailto:celia.taniwaki@sptech.school)**

**SÃO  
PAULO  
TECH  
SCHOOL**