BandTec

DIGITAL SCHOOL

BandTec
DIGITAL SCHOOL

# ED

# Estrutura de Dados e Armazenamento

Pilha

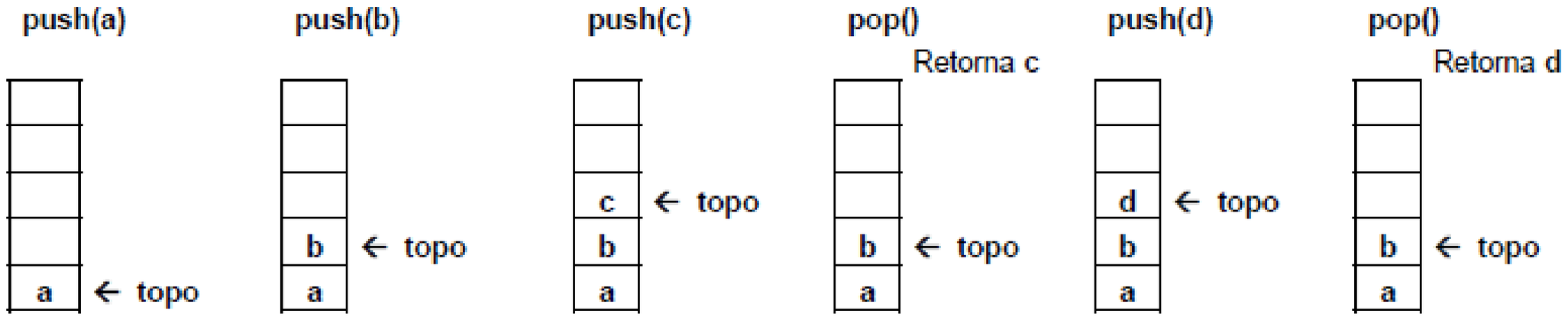
# Pilha

- Estrutura de dados que se caracteriza por:
  - Armazenar elementos de mesmo tipo
  - A inserção e a remoção de elementos é sempre feita pelo topo da pilha (fim da sequência).

    (Analogia com pilha de pratos ou de livros)
  - 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 a sair

# Operações na pilha

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



# Exercício

- 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 (30)

# 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>();
```

- Não aceita tipos primitivos
- Métodos de operação da pilha:
  - isEmpty() – devolve true se a pilha está vazia, e false caso contrário
  - 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 de capacidade N.
    - 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.

# Implementação da classe Pilha

- Criar um projeto chamado exemplo-pilha
- Criar uma classe chamada Pilha
- Implementar a classe Pilha, utilizando a pilha como vetor com
  - 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 topo com -1

# Implementação da classe Pilha

- Métodos da classe Pilha
  - boolean isEmpty( ) // devolve true caso a pilha esteja vazia
  - boolean isFull( ) // devolve true caso a pilha esteja cheia
  - void push (int info) // se pilha não está cheia, então increm. topo
    // e empilha info na pilha[topo], senão exibe msg
    // "Pilha cheia"
  - int pop ( ) // se 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 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.
  - No método main:
    - Crie um objeto da classe Pilha
    - Empilhe valores na pilha
    - Depois desempilhe um por um e exiba-os

    (Você verá que eles serão exibidos na ordem inversa da que foram empilhados)

# Aplicação da pilha

- Quando devo usar pilha para armazenar os dados?
- Resposta: quando desejo usar os dados exatamente na ordem inversa em que foram inseridos
- Exemplos:
  - Ação de desfazer de editores de texto (Word ou Visual Studio, por exemplo) – a última ação realizada é a primeira a ser desfeita – isso significa que o editor de texto utiliza uma pilha para armazenar as ações realizadas
  - Conversão de número decimal para binário, utilizando o método da divisão – dividimos o número decimal por 2, sucessivamente, até que o resultado seja zero. O número binário é formado pelos restos da divisão, considerando-os de trás para frente.

# Obrigada!

BandTec DIGITAL SCHOOL

Em caso de dúvidas, entre em contato com:
celia.taniwaki@bandtec.com.br