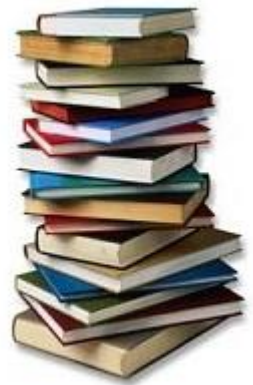

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
ESCOLA POLITÉCNICA E DE ARTES
ESTRUTURA DE DADOS ORIENTADA A OBJETOS
ADS1232
PROF. MSC. ANIBAL SANTOS JUKEMURA

PILHAS

PILHA (STACK)

- Pilha

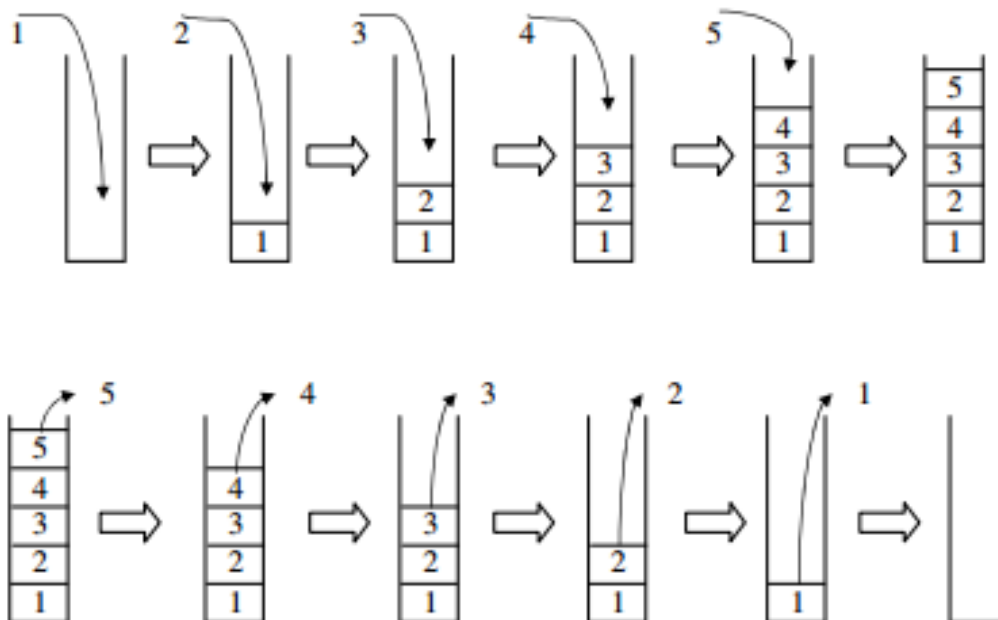
- estrutura de dados linear em que a inserção e a remoção de elementos de uma sequência se faz pela mesma extremidade, designada por **topo** da pilha
- uma pilha pode ser considerada como uma *restrição* de lista
- porque é uma estrutura de dados mais simples que a lista, é possível obter implementações mais eficazes
- o conceito de iterador **não se aplica** a esta estrutura de dados
- a pilha é uma estrutura do tipo LIFO (*Last-In-First-Out*)
- Operações mais usuais:
 - criar uma pilha vazia
 - adicionar/remover um elemento a uma pilha
 - verificar qual o último elemento colocado na pilha
 - verifica se a pilha está vazia (ou cheia no caso de implementação por vetores com tamanho fixo)





PILHA (STACK)

- Operações com pilhas



PILHA (STACK)

USOS:

- Solução de expressões matemáticas
 - Calculadora HP-12c
- Guardar variáveis locais em chamadas recursivas de função

PILHA (STACK)



Implementações:

- API JAVA
- Estruturas estáticas: vetores
- Estruturas dinâmicas: listas simplesmente encadeadas

PILHA (STACK)

- Declaração:
 - **import** java.util.Stack;
 - **Stack<TIPO>** pilha = new **Stack<TIPO>**();
- Principais métodos:
 - **boolean empty()**: testa se a pilha está vazia.
 - **E peek()**: retorna o objeto que está no topo da pilha sem removê-lo.
 - **E pop()**: remove (desempilha) o objeto que está no topo da pilha.
 - **push (E item)**: empilha o objeto E na pilha.
 - **int search(Object O)**: retorna a posição de um objeto que está na pilha.

PILHA (STACK)

- Métodos herdados da classe Vector
 - **clear()**: esvazia a pilha.
 - **boolean contains(Object O)**: verifica se o objeto está na pilha.
 - **Iterator <E> iterator()**: iterator usado para retornar os elementos da pilha (da base ao topo).
 - **E firstElement ()**: retorna o primeiro elemento da pilha sem removê-lo.
 - **int size()**: retorna o tamanho da pilha.

(Cuidado): esses demais métodos foram herdados da classe Vector e descaracterizam o funcionamento de uma pilha, apesar de poderem ser utilizados.

add, addAll, addAll, addElement, capacity, clone, containsAll, elementAt, elements, ensureCapacity, equals, get, indexOf, insertElementAt, lastElement, lastIndexOf, listIterator, remove, removeAll, removeAllElements, removeElement, removeElementAt, removeRange, retainAll, set, setElementAt, setSize, subList.

Exemplo em sala de aula



Implementar os métodos demonstrados anteriormente

Exercício em Sala

Utilize uma pilha para imprimir a mensagem a seguir:

COPA DO MUNDO

Empilhe, letra por letra:

XYOYDXYNXYYUYYYM OD YAPYOC

Dica: empilhe e desempilhe conforme necessidade.




```
1  import java.util.Stack;
2  import java.util.Iterator;
3
4  public class TestaPilha {
5      public static void main(String[] args)
6      {
7          Stack<Integer> pilha = new Stack<Integer>();
8          if (pilha.empty())
9              System.out.println("Pilha vazia");
10         pilha.push(100);
11         pilha.push(200);
12         pilha.push(300);
13         System.out.println("Pilha: " + pilha);
14         pilha.pop();
15         System.out.println("Pilha: " + pilha);
16         pilha.push(500);
17         System.out.println("Pilha: " + pilha);
18         if (pilha.contains(200))
19         {
20             System.out.println("Pilha contem elemento 200");
21             System.out.println("Posicao encontrada: " + pilha.search(200));
22         }
23         System.out.println("Elemento no topo: " + pilha.peek());
24         System.out.println("Pilha: " + pilha);
25         System.out.println("Elemento na base: " + pilha.firstElement());
26         System.out.println("Pilha: " + pilha);
27         Iterator<Integer> it = pilha.iterator();
28         while (it.hasNext())
29         {
30             System.out.print(" " + it.next());
31         }
32         System.out.println("");
33     }
34 }
```

IMPLEMENTAÇÃO DE ESTRUTURAS DE DADOS

PILHAS DE OBJETOS (polimorfismo) (Stack)

Implementar os métodos da classe pilha.

- `public Pilha(int n)`
- `public boolean pilhaVazia()`
- `public int tamanho()`
- `public void empilhar(Object valor)`
- `public Object desempilhar()`
- `public void limpaPilha()`
- `public Object exhibeTopo()`
- `public void imprimir()`

