

INSTITUTO FEDERAL  
Paraíba  
Campus João Pessoa

Aula

4

# Programação e Estrutura de Dados

Atualizada em 26/08/2019

## Pilhas

*Abordagem Orientada a Objetos*



**Professor:**

Dr. Alex Sandro da Cunha Rêgo



[alex@ifpb.edu.br](mailto:alex@ifpb.edu.br)



- **Definição**

- Estrutura de dados em que as operações de **inserção/remoção** são realizadas em apenas uma das extremidades da sequência de dados

- **Características**

- O último element a entrar é o primeiro a sair (**Last In First Out - LIFO**)

- **Exemplos**

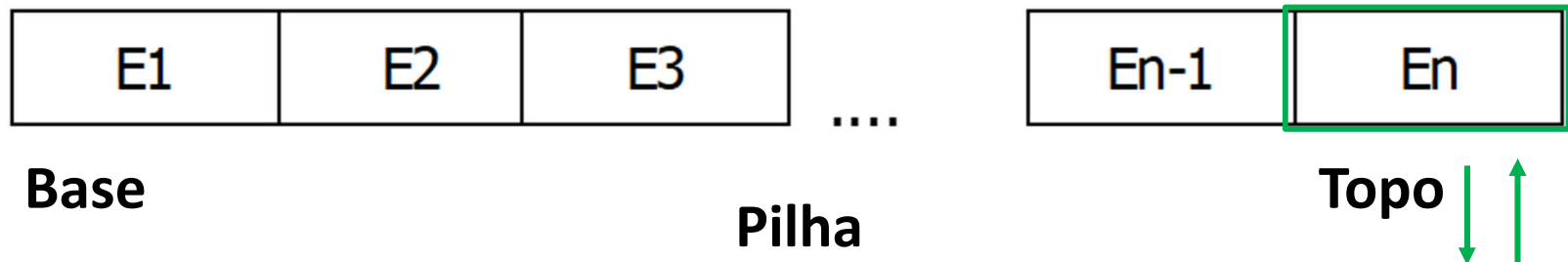
- Preparação para distribuição de pratos (restaurante)
- Rua sem saída estreita, onde só entra ou sai um carro por vez
- Pilha de cartas de um baralho



- **Propriedades**

- ❑ Relação ordenada de  $n$  elementos
- ❑  $E_1$  é o elemento da **base** da pilha
- ❑  $E_n$  é o elemento do **topo** da pilha
- ❑ A inserção do mais recente elemento  $v$  torna-o o elemento da vez para remoção
- ❑ A retirada é feita sempre sob o elemento  $E_n$

- **Representação visual**



# Pilhas



- **Ilustração**

**Topo** da pilha  
(acessível)



**Base** da pilha  
(inacessível)



**Empilhando** um  
novo livro no  
**topo**

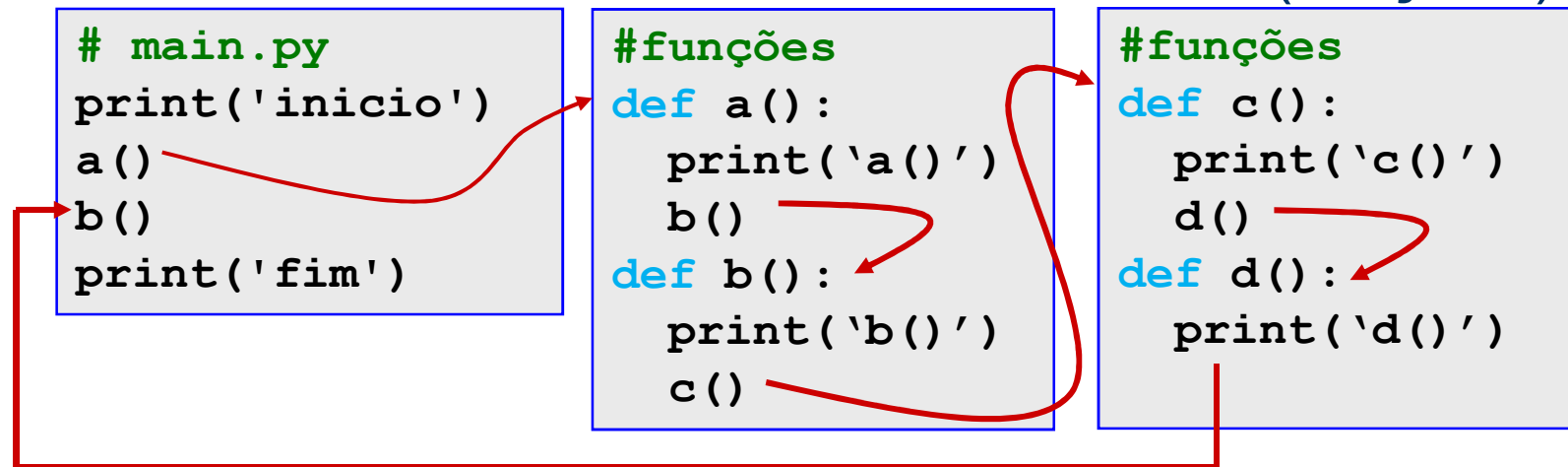


**Desempilhando** um  
livro a partir do **topo**



- O uso da pilha na computação

- Controle de chamadas de sub-rotinas (funções)



- Inverter a ordem de chegada dos elementos



- Avaliar expressões aritméticas





## • Operações primárias

Operação	Descrição
<code>inicializa()</code>	Inicializar os controladores da pilha
<code>estaVazia()</code>	Verificar se a pilha está vazia
<code>estaCheia()</code>	Verificar se a pilha está cheia
<code>tamanho()</code>	Obter o número de elementos da pilha
<code>elemento()</code>	Obter o conteúdo armazenado em uma posição da pilha
<code>busca()</code>	Determinar a posição ( $E_i$ ) em que um valor se encontra
<code>modificar()</code>	Alterar o valor armazenado em uma posição ( $E_i$ )
<code>empilha()</code>	Adiciona um novo elemento ao topo da pilha
<code>desempilha()</code>	Remove o elemento localizado no topo da pilha
<code>imprimir()</code>	Exibe a sequência de elementos da pilha



**NOTA:** As operações podem sofrer variações de código de acordo com a técnica de implementação pilha.



- Implementação em Python

- Sequencial

- ✓ O objeto `list` de python acrescenta a facilidade do acesso indexado a um array e a flexibilidade de acrécimo de elementos de forma dinâmica

```
>>> lista = [25,12,45,63]
>>> lista.append(45)
>>> lista
[25, 12, 45, 63, 45]
>>> print(lista[0], lista[3])
25 63
>>> lista.insert(0,11)
>>> lista
[11, 25, 12, 45, 63, 45]
```

- ✓ O esforço para implementação de uma **estrutura de dados Pilha** de forma encadeada é reduzido drasticamente
    - ✓ O cuidado é direcionado ao **tratamento de exceções**



## ● Estrutura de Classes

### Pilha.py

```
# TAD da Lista e suas Operacoes
class Pilha:

    def __init__(self): # TAD
        self.__dados = []
    def estaVazia(self):
        ...
    def tamanho(self):
        ...
    def imprime(self):
        ...
    def empilha(self, valor):
        ...
    def desempilha(self):
        ...
    def __str__(self):
        ...
```

### main.py

```
# main.py

from Pilha import Pilha

p = Pilha()

p.empilha(20)

p.empilha(30)

p.empilha(40)

p.imprime() # 40,30,20
print(p.estaVazia())

print(p.desempilha())

...
```



# Laboratório



Vamos implementar nossa **Pilha Sequencial** O.O. em **Python**?



# Pilha Encadeada



- **Definição**

- ☐ Os nós (elementos) da pilha são alocados dinamicamente e conectados entre si por referências, manualmente via código

- **Operações**

- ☐ As mesmas operações da pilha sequencial
  - ✓ Preservar a prototipagem das operações
- ☐ O que vai mudar é a implementação interna

Quais adaptações serão necessárias realizar para adaptar o código da **Lista Encadeada** à implementação da **Pilha Encadeada**?

# Laboratório



Vamos então implementar a **Pilha** usando a técnica **encadeada**?



# Referências Bibliográficas



- VILLAS, M., V.; FERREIRA, A., G.,M.; LEROY, P. G.; MIRANDA, C.; BOCKMAN, C., L. Estruturas de Dados. Rio de Janeiro: Campus, 1993.
- MILLER,B.; RANUM,D. Problem Solving with Algorithms and Data Structres. [sl], 2013.
  - <http://interactivepython.org/runestone/static/pythonds/index.html>
- Python Data Structure and Algorithms Tutorial.
  - [https://www.tutorialspoint.com/python/python\\_data\\_structure.htm](https://www.tutorialspoint.com/python/python_data_structure.htm)