**Movimento LIFO em Python**

**Mário Leite**

...

Movimentações e rearranjos de elementos em uma estrutura de dados é sempre um terror para os estudantes iniciantes na disciplina “Estrutura de Dados” nos cursos de Programação. É verdade: qualquer assunto sobre Pilhas não é muito trivial, pois a lógica envolvida nesse tipo de organização de dados é complexa.

O conceito de **LIFO** (**L**ast **I**n, **F**irst **O**ut - "Último a Entrar, Primeiro a Sair") é um princípio de organização e manipulação de dados em estruturas. Esse conceito é aplicável às pilhas (*stacks*), em que define uma estrutura de dados do tipo "seco e empilhe", ou seja, é possível apenas adicionar ou remover elementos do topo da pilha usando algumas funções básicas: **push**, **pop** e **peek**. Esse tipo de movimento em pilhas funciona assim:

1. **Último a Entrar**: O último elemento que for adicionado à pilha será o primeiro a ser retirado; em outras palavras, a pilha segue o comportamento "*o que chega por último é o primeiro a sair*".
2. **Operações principais**:
   * **Push**: Quando você adiciona um novo elemento à pilha, ele é colocado no topo; essa operação é chamada de "push".
   * **Pop**: Quando um elemento é retirado da pilha, ele é removido do topo; essa operação de remoção é chamada de "pop".
   * **Peek** ou **Top**: Retorna o elemento no topo da pilha sem removê-lo, permitindo que seja evidenciado o último elemento adicionado.

Algumas aplicações de **LIFO** podem ser citadas:

* Chamadas de funções: O processo de chamadas e retornos de funções em um programa segue o modelo **LIFO**. As funções são empilhadas conforme chamadas e desempilhadas quando retornam.
* Navegação em navegadores: O histórico de páginas visitadas pode ser armazenado em uma pilha, de forma que a última página visitada é a primeira a ser acessada ao clicar no botão "Voltar".
* Algoritmos de busca: Algoritmos como DFS (Depth-First Search) utilizam pilhas para explorar nós em grafos ou árvores de maneira recursiva.

O programa **“LIFOPilha”** é um exemplo simples de código que executa esse tipo de movimento, mostrando as principias funções que podem ser usadas na movimentação dos elementos.

-------------------------------------------------------------------------------------------------------------------------------------

'''

**LIFOPilha.py**

---------------------------------------------------------------------------------

Cria uma classe para mostrar as principais ações disponíveis com uso de uma pilha do tipo **LIFO**, como uma estrutura de dados indexada usando lista.

---------------------------------------------------------------------------------

'''

**class** **ClsPilha**:

**def** **\_\_init\_\_**(**self**): *#cria o "Construtor" da classe*

**self**.LstItens = []

**def** **FazerPush**(**self**, item):

**self**.LstItens.**append**(item)

**def** **FazerPop**(**self**):

**if**(**not** **self**.**VeririficarPilhaVazia** ()):

**return** **self**.LstItens.pop()

**else**:

raise **IndexError**("A Pilha está vazia")

**def** **FazerPeek**(**self**):

**if**(**not** **self**.**VeririficarPilhaVazia** ()):

**return** **self**.LstItens [-1]

**else**:

**raise** **IndexError**("A Pilha está vazia")

**def** **VeririficarPilhaVazia**(**self**):

**return** **len**(**self**.LstItens) == 0

**def** **VerificarSize**(**self**):

**return** **len**(**self**.LstItens)

*#==========================================================================================*

*##Programa principal*

**Pilha** = **ClsPilha**()

**Pilha**.**FazerPush**(1)

**Pilha**.**FazerPush**(2)

**Pilha**.**FazerPush**(3)

**Pilha**.**FazerPush**(4)

**Pilha**.**FazerPush**(5)

**print**("Elementos na Pilha:", **Pilha**.**VerificarSize** ())

**print**("Elemento no topo da Pilha:", **Pilha**.**FazerPeek** ())

**while**(**not** **Pilha**.**VeririficarPilhaVazia**()):

removido = **Pilha**.**FazerPop**()

**print**("Removendo elemento:", removido)

**if**(**not** **Pilha**.**VeririficarPilhaVazia**()):

**print**("Elemento no topo da Pilha após remoção:", **Pilha**.**FazerPeek** ())

**#Fim do programa "DetalhesDePilha" ----------------------------------------------**



**Figura 3.3 - Saída do programa “DetalhesDePilha”**