

Estrutura de Dados em Python

Prof. Nisston Moraes Tavares de Melo

Introdução ao Python

- Pilha
- Operações
- Aplicações
- Exemplo
- Exercício

Estrutura de dados em Python

Dra. Ana Carolina Costa de Oliveira
prof2121@iesp.edu.br

1

Pilha

Pilhas



Pilhas

A – você está ocupado com um projeto de longo prazo

B – é interrompido por um colega solicitando ajuda em um outro projeto

C – enquanto estiver trabalhando em B, alguém da contabilidade aparece para uma reunião sobre despesas de viagem

D – durante a reunião, recebe um telefonema de emergência de alguém de vendas e passa alguns minutos resolvendo um problema relacionado a um novo produto

Quando tiver terminado o telefonema **D**, voltará para a reunião **C**; quando tiver acabado com **C**, voltará para o projeto **B** e quando tiver terminado com **B**, poderá finalmente voltar para o projeto **A**

D

C

B

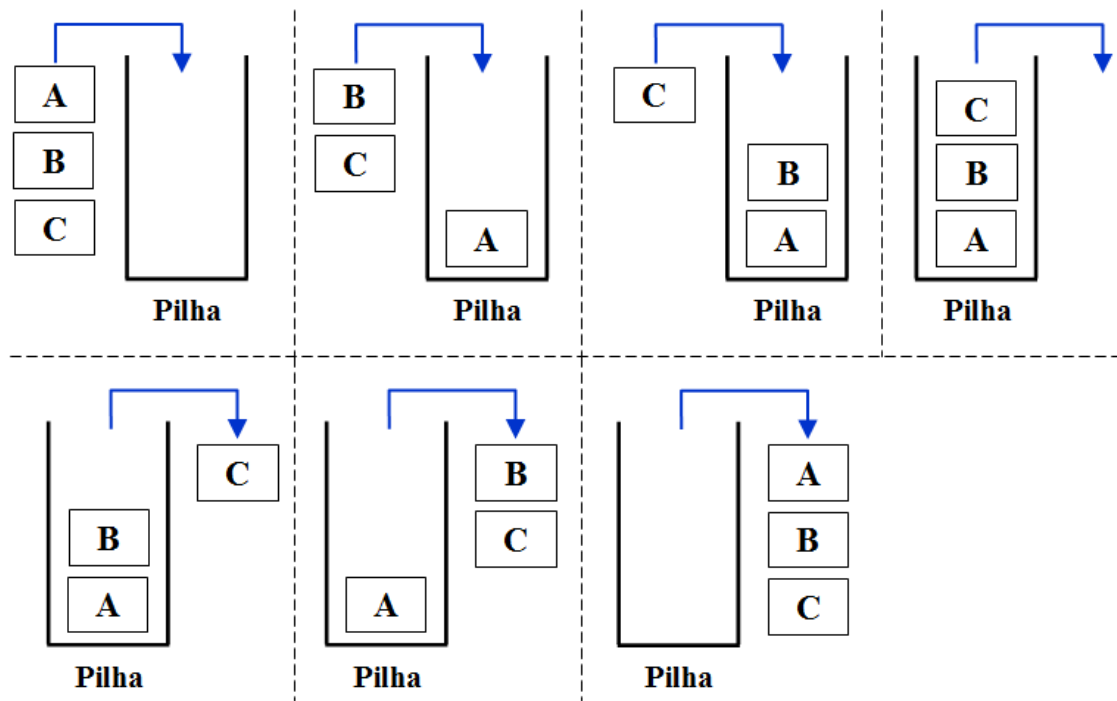
A

Pilhas

- Permite acesso a um item de dados: o último item inserido.
- Se o último item for removido, o item anterior ao último inserido poderá ser acessado.



Exemplo



Aplicações de Pilhas

- **Navegação em histórico de páginas:** Ao criar um navegador da web simplificado, você pode usar uma pilha para armazenar as páginas visitadas. Quando o usuário pressionar o botão "voltar", você poderá desempilhar a página anterior da pilha.
- **Implementação de uma calculadora:** Ao criar uma calculadora que permite desfazer operações, uma pilha pode ser usada para rastrear as operações e seus resultados, permitindo que o usuário desfaça a operação mais recente.
- **Validação de expressões matemáticas:** Você pode usar uma pilha para avaliar se uma expressão matemática, como uma expressão com parênteses, está corretamente balanceada. Ao percorrer a expressão, você pode empilhar os parênteses de abertura e desempilhá-los quando encontrar os parênteses de fechamento correspondentes.
- **Rastreamento de chamadas de função:** Em linguagens de programação, como Python, você pode usar uma pilha para rastrear as chamadas de função. Quando uma função é chamada, você a empilha na pilha de chamadas, e quando a função retorna, você a desempilha.

Aplicações de Pilhas

- **Processamento de texto reverso:** Para inverter a ordem das palavras em uma frase ou inverter a ordem dos caracteres em uma string, você pode usar uma pilha para armazenar as palavras ou caracteres na ordem inversa.
- **Solução de problemas recursivos:** Em algoritmos recursivos, como a recursão em árvores, você pode usar uma pilha para evitar o estouro de pilha de chamadas recursivas, implementando uma versão iterativa do algoritmo.
- **Histórico de ações em jogos:** Em jogos de tabuleiro ou aplicativos de jogos, você pode usar uma pilha para rastrear as ações dos jogadores, permitindo que eles desfaçam ou refaçam jogadas anteriores.
- **Implementação de um sistema de undo/redo:** Em aplicativos de edição de texto, gráficos ou design, uma pilha pode ser usada para manter um histórico de ações, permitindo que os usuários desfaçam e refaçam operações.

Pilha – Operações

- **Empilhar**

- Colocar um item de dados no topo da pilha

- **Desempilhar**

- Remover um item do topo da pilha

- **Ver o topo**

- Mostra o elemento que está no topo da pilha

Último-A-Entrar-Primeiro-A-Sair (LIFO – Last-In- First-Out)

PILHAS

```
class Pilha:
    def __init__(self, capacidade):
        self.capacidade = capacidade
        self.items = []

    def empilhar(self, item):
        if not self.is_full():
            self.items.append(item)
        else:
            print("A pilha está cheia. Não é possível empilhar o item.")

    def desempilhar(self):
        if not self.is_empty():
            return self.items.pop()
        else:
            print("A pilha está vazia. Não é possível desempilhar.")
```

PILHAS

```
def is_full(self):  
    return len(self.items) == self.capacidade  
  
def is_empty(self):  
    return len(self.items) == 0  
  
def ver_topo(self):  
    if not self.is_empty():  
        return self.items[-1]  
    else:  
        print("A pilha está vazia. Não há topo para visualizar.")
```

PILHAS

```
def visualizar(self):  
    if not self.is_empty():  
        print("Itens da pilha:")  
        for item in self.items:  
            print(item)  
    else:  
        print("A pilha está vazia. Não há itens para visualizar.")
```



Exercício.



uniesp

Centro Universitário