

Prática 3 Estruturas de Dados

- Pilhas e Filas
- Listas Ligadas
- Criação de classes para estas estruturas de dados (ADT)

Exercício 3.1 – Pilha (Stack)

Crie uma classe implementando o tipo de dados abstracto Pilha (Stack) com base numa lista. Deve implementar o seguinte conjunto de métodos (interface):

- **push(e)** – Insere o elemento “e” no topo da pilha.
- **pop()** – Remove o elemento no topo da pilha (devolvendo-o).
- **top()** – Devolve uma referência para o elemento no topo da pilha.
- **size()** – Devolve o tamanho (número de elementos) na pilha.
- **isEmpty()** – Devolve se a pilha está vazia.
- **clear()** – Reinicia a pilha, sem elementos.

Teste com código disponibilizado no elearning (**pilha_teste.py**).

Exercício 3.2 – Fila (Queue)

Crie uma classe implementando o tipo de dados abstracto Fila (Queue) com base numa lista. Deve implementar a seguinte interface:

- **add(e)** – Adiciona o elemento “e” à fila. Se estiver cheia resulta num erro.
- **remove()** – Remove e devolve um elemento da fila. Ocorre um erro se estiver vazia.
- **first()** – Devolve o elemento no início da fila (sem o remover).
- **size()** – Devolve o tamanho (número de elementos) na fila.
- **isEmpty()** – Devolve se a fila está vazia.
- **clear()** – Reinicia a fila, sem elementos.

Teste com código disponibilizado no elearning (**fila_teste.py**).

Exercício 3.3 - Palíndromo

Crie uma função palindromo() que detecte se uma sequência de letras e dígitos é um palíndromo (ou seja, se a sequência lida do início para o fim é igual a sequência lida do fim para o início).

Numa primeira fase a sua função pode considerar todos os caracteres. Depois de ter a função a funcionar deve melhorá-la para ignorar todos os caracteres que não são letras ou dígitos.

Por exemplo as sequências “somos” e “O galo nada no lago” são palíndromos, o último apenas ignorando caracteres que não são letras.

Deve utilizar as estruturas de dados dos exercícios anteriores que considerar adequadas e fazer uso da importação de módulos para ter acesso às classes que necessitar. Exemplo:

```
from MyStack import Stack
```

Comece por testar com uma lista de strings definidas por si, contendo palíndromos e não palíndromos. Numa segunda fase, desenvolva um pequeno programa que permita a um utilizador introduzir frases para verificação, até que escreva uma de comprimento zero.

Exercício 3.4 - Uma outra implementação da Pilha (Stack)

Desenvolva uma nova implementação do tipo de dados abstrato Pilha (Stack), desta vez usando para a sua implementação a estrutura de dados LinkedList fornecida (ficheiro **LinkedList.py**). Deve manter a interface da classe inalterada por forma a que a sua resolução do problema do palíndromo funcione simplesmente alterando o import da classe Pilha/Stack.

Teste com o programa que desenvolveu para o palíndromo.

Exercício 3.5– Uma outra implementação da Fila (Queue) [TPC]

Desenvolva uma nova implementação do tipo de dados abstrato Fila (Queue), desta vez usando para a sua implementação a estrutura de dados LinkedList fornecida. Deve manter a interface da classe por forma a que o problema do palíndromo funcione simplesmente alterando o import.

Teste com o programa que desenvolveu para o palíndromo.

Exercício 3.6- Calculadora [TPC]

Usando como base a(s) estrutura(s) de dados que considerar adequadas, desenvolva uma calculadora com as quatro operações aritméticas básicas que funcione com a notação pós-fixa (Reverse Polish Notation).

Na notação pós-fixa os operandos são colocados antes dos operadores. Assim **2 + 3** passa a ser expresso por **2 3 +**. Esta notação dispensa o uso de parêntesis e tem uma implementação muito simples baseada numa pilha. Sempre que aparecer um número este é carregado para a pilha. Sempre que aparece um operador, são retirados os dois últimos da pilha e o resultado da operação é colocado na pilha. Se a pilha não tiver o número de operandos necessários, então temos um erro sintático na expressão.

Desenvolva um programa de teste para a sua calculadora com algumas expressões concretas.

Desenvolva, também, um programa que permita que as expressões sejam introduzidas pelo utilizador.