

# Palíndromo

Willian Magnum Albeche

## Desafio proposto

Desenvolvimento de um programa, em linguagem de programação Java, que leia uma cadeia de caracteres e que indique se esta cadeia de caracteres é ou não um palíndromo.

Possuindo algumas restrições:

Exercício individual.

A leitura da cadeia de caracteres deve ser caractere a caractere até que apenas a tecla [Enter] seja pressionada.

NÃO é permitido usar Strings.

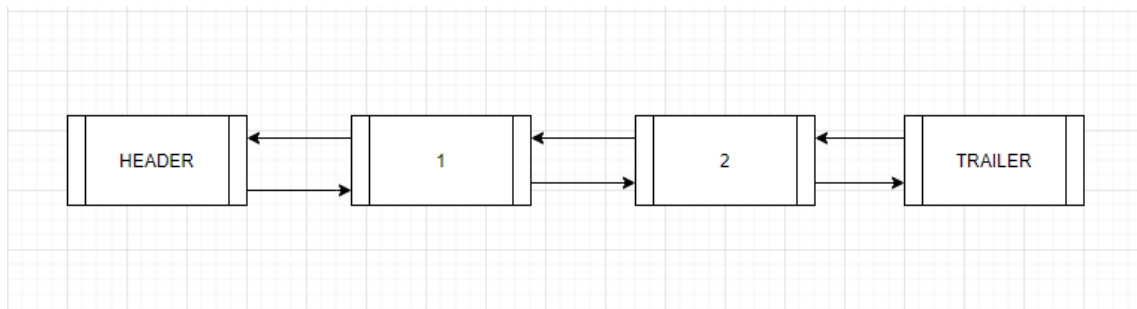
NÃO é permitido utilizar array [ ], nem ArrayList.

NÃO é permitido usar mais de duas coleções do mesmo tipo.

**Palíndromo:** uma palavra, frase ou qualquer outra sequência de letras que tem a mesma leitura se lida tanto da esquerda para a direita como da direita para a esquerda

## Solução

Para realizar o desafio proposto, foi utilizada uma Linked list, mais precisamente uma Double Linked List, abaixo um diagrama que demonstra seu funcionamento:



Uma Double Linked List possui nodos, que referenciam o próximo e no caso da Double, referenciam o anterior também, o header e o trailer, que

são respectivamente nulls, com isso podemos facilmente percorrer ela para acharmos um valor em específico ou para adicionarmos um valor, no caso desse programa, sua utilização foi bem simples, a cada char digitado, o mesmo era adicionado a lista, e quando o usuário finalizasse a inserção, a lista seria percorrida do início até a metade e do fim até a metade, ao mesmo tempo, comparando cada char, se o resultado de todas as comparações fosse true, então a sequência de letras é um palíndromo, para isso, foi criada uma classe Palíndromo que faz os processos necessários.

inserção é feita por um método void:

```
public void addList(char c){  
    c= Character.toUpperCase(c);  
    list.add(c);  
}
```

Para garantir que só entrasse letras maiúsculas foi usado um método de UpperCase, e para adicionar, foi utilizado o próprio .add da lista.

Para fins de demonstração, foi feito o método mostrar, que além de mostrar o conteúdo da lista, mostrava seu reverso para ficar visível se é ou não um palíndromo:

```
public void mostrar(){  
    System.out.println(Arrays.toString(list.toArray()));  
    int j= list.size()-1;  
    System.out.println("reverse:");  
    System.out.print("[");  
    for (int i = 0; i <list.size() ; i++) {  
        if(i== 0){  
            System.out.print(list.get(i)+",");  
        }  
        else if(i>0 && i <list.size()-1){  
            System.out.print(" "+list.get(i)+",");  
        }  
        else{  
            System.out.print(" "+list.get(i));  
        }  
        i--;  
    }  
    System.out.println("]");  
}
```

E por fim, o método que verifica se a sequência de letras é um palíndromo, que consiste em percorrer a lista do seu início até a metade, e ao mesmo tempo percorrer do seu fim até a metade, comparando char com char a cada troca de posição, é um método booleano que retorna true caso a sequência de letras for um palíndromo, e false caso não for:

```
public boolean ePalindromo(){
    int j=list.size()-1;
    boolean confere=false;
    for (int i = 0; i <list.size()/2 ; i++) {
        if(list.get(i) == list.get(j)){
            confere = true;
        }
        else{
            confere = false;
            return confere;
        }
        j--;
    }
    if(confere == true){
        return true;
    }
    return false;
}
```

Com isso, foram apenas adicionadas mais algumas linhas de código no main para melhor organizar o terminal e ficar mais claro o resultado.

Primeira parte do terminal:

```
=====
Digite os caracteres e pressione enter, após isso, digite 0 para ver se é um palindromo ou não
=====
hit a key: |
```

Após digitar a frase como exemplo: “SOCORRAM-ME SUBI NO ONIBUS EM MARROCOS”. Ele mostra cada caractere digitado e seu valor na tabela ASCII, por fim, quando o usuário digita 0, o programa finaliza mostrando a palavra e se seu reverso é um palíndromo.

## Segunda parte do terminal:

```
hit a key: 0
Tecla pressionada: 48
Tecla pressionada: 0
printando array:
[S, O, C, O, R, R, A, M, M, E, S, U, B, I, N, O, O, N, I, B, U, S, E, M, M, A, R, R, O, C, O, S]
reverse:
[S, O, C, O, R, R, A, M, M, E, S, U, B, I, N, O, O, N, I, B, U, S, E, M, M, A, R, R, O, C, O, S]

É palindromo? Sim!
```