

Prática de Laboratório 3

Objetivos:

- Introdução ao conceito de Tipos Abstratos de Dados - TADs.
- Implementação da TAD Lista Simplesmente Encadeada.

Classe nodo

```
1
2 package Node;
3
4 public class Nodo {
5
6     private String element;
7     private Nodo next;
8
9     public Nodo(String s, Nodo n) {
10         element = s;
11         next = n;
12     }
13
14     public String getElement() {
15         return element;
16     }
17
18     public Nodo getNext() {
19         return next;
20     }
21
22     public void setElement(String newElem) {
23         element = newElem;
24     }
25
26     public void setNext(Nodo newNext) {
27         next = newNext;
28     }
29
30 }
```

Classe Lista Encadeada

```
1 package ListaEncadeada;
2
3 import Node.Nodo;
4
5 public class ListaEncadeada {
6     protected Nodo cabeca; // cabeca é o primeiro nodo da lista
7     protected long tamanho; // Número de nodos na lista
8
9     public ListaEncadeada() {
10         this.cabeca = null;
11         this.tamanho = 0;
12     }
13
14     void insereNoInicio(Nodo n) {
15         n.setNext(this.cabeca);
16         this.cabeca = n;
17         this.tamanho++;
18     }
19
20     void imprimeLista() {
21         Nodo aux = this.cabeca;
22         while (aux != null) {
23             System.out.println(aux.getElement());
24             aux = aux.getNext();
25         }
26     }
27
28     // Inclua códigos aqui
29
30 }
```

Atividades

1. Implemente as classes Nodo e Lista Encadeada acima.
2. Crie uma classe principal onde no main são feitos os testes da Lista Encadeada.
3. Utilizando a classe Lista Encadeada, implemente as seguintes funções:
 - a. Excluir elemento
 - b. Incluir no final da lista
 - c. Incluir ordenado
4. Modifique as classes Nodo e Lista Encadeada de forma que seja possível criar uma lista de Jogadores, onde deve ser possível executar todas as funções implementadas em sala hoje. Considere a ordenação da lista pela pontuação do jogador.

Bom trabalho! 🖐