Prática de Laboratório 4

Objetivos:

- Tipos Abstratos de Dados - TADs.
- Implementação da TAD Lista Duplamente Encadeada.

Classe nodeD

```
public class NodeD {

	private String item;
	private NodeD next, prev;


	public NodeD(String item, NodeD next, NodeD prev) {
		super();
		this.item = item;
		this.next = next;
		this.prev = prev;
	}
	public String getItem() {
		return item;
	}
	public void setItem(String item) {
		this.item = item;
	}
	public NodeD getNext() {
		return next;
	}
	public void setNext(NodeD next) {
		this.next = next;
	}
	public NodeD getPrev() {
		return prev;
	}
	public void setPrev(NodeD prev) {
		this.prev = prev;
	}

}
```

Classe Lista Duplamente Encadeada

```
public class ListaD {

	private int tamanho;
	private NodeD NodoCabeca, NodoFinal;

	public ListaD() {
		super();
		//Iniciliza lista vazia
		this.tamanho = 0;
		//Define novo nodo pra Cabeça da lista
		this.NodoCabeca = new NodeD("", null, null);
		//Define novo nodo Final da Lista, seta o prev dele como nó cabeça
		this.NodoFinal = new NodeD("", null, this.NodoCabeca);
		//Atualiza o próximo do nodo Cabeça o nodo Final
		this.NodoCabeca.setNext(this.NodoFinal);
	}
	public int getTamanho() {
		return tamanho;
	}
	public NodeD getNodoCabeca() {
		return NodoCabeca;
	}
	public void setNodoCabeca(NodeD nodoCabeca) {
		NodoCabeca = nodoCabeca;
	}
	public NodeD getNodoFinal() {
		return NodoFinal;
	}
	public void setNodoFinal(NodeD nodoFinal) {
		NodoFinal = nodoFinal;
	}

	/*Implementem aqui as seguintes funções:
	 * - Verifica se lista está vazia
	 * - Busca o primeiro elemento da lista (não o cabeça)
	 * - Busca o último elemento da lista (não o final)
	 * - Insere antes de um nodo
	 * - Insere depois de um nodo
	 * - Insere ordenado
	 * - Adiciona no início
	 * - Adiciona no final
	 * - Busca na lista
	 * - Remove (passe o Nodo que será excluído)
	 * */
}
```

Atividades

1. Implemente na classe Lista Duplamente Encadeada (ListD), os seguintes métodos:
   - `- Verifica se lista está vazia`
   - `- Busca o primeiro elemento da lista (não o cabeça)`
   - `- Busca o último elemento da lista (não o final)`
   - `- Insere antes de um nodo`
   - `- Insere depois de um nodo`
   - `- Insere ordenado`
   - `- Adiciona no início`
   - `- Adiciona no final`
   - `- Busca na lista`
   - `- Remove (passe o Nodo que será excluído)`

2. Crie uma classe principal onde no main são feitos os testes da Lista Duplamente Encadeada.

3. Modifique as classes Nodo e Lista Duplamente Encadeada de forma que seja possível criar uma lista de Alunos.
   a) Implemente uma classe Aluno com os seguintes dados:
      Nome
      Matrícula
      Data nascimento
   b) Modifique a lista duplamente encadeada para que esta seja uma lista de Alunos
   c) Crie no main as seguintes opções:
      a. Criar novo aluno
      b. Inserir aluno na lista
      c. Buscar aluno por matrícula
      d. Remover aluno da lista
      e. Imprimir alunos da lista

Bom trabalho!