

**Instruções para a entrega:** fazer os exercícios a seguir e mostrar para o professor no computador na aula do dia 05/abr. A entrega pode ser em dupla, alunos ausentes não terão a nota considerada.

Considere as classes a seguir nos exercícios. A classe **Lista** implementa uma lista **Duplamente Encadeada** e a classe **ListaCircular** implementa uma lista **Circular Duplamente Encadeada**.

```
public class No {
    int conteudo;
    No anterior, proximo;
}
```

```
public class Lista {
    No inicio;

    void add(int nro) {
        No no = new No();
        no.conteudo = nro;
        no.anterior = null;
        no.proximo = null;
        if( inicio == null ){
            inicio = no;
        }
        else{
            No ultimo = inicio;
            while( ultimo.proximo != null ){
                ultimo = ultimo.proximo;
            }
            ultimo.proximo = no;
            no.anterior = ultimo;
        }
    }

    void print() {
        if( inicio == null ){
            System.out.println("\nLista vazia");
        }
        else{
            /* quebra de linha na tela */
            System.out.println();
            No ultimo = inicio;
            while( ultimo != null ){
                System.out.print(
                    ultimo.conteudo + " ");
                ultimo = ultimo.proximo;
            }
        }
    }
}
```

**Exercício 1** – Programar na classe **Lista** um método de nome **addInicio(nro:int)** para inserir os elementos no início da lista, desta forma, o início da lista será sempre deslocado para o nó inserido.

**Exercício 2** – Programar na classe **Lista** um método de nome **addOrder(nro:int)** para inserir os elementos na lista em ordem crescente de valor, desta forma, a lista sempre estará ordenada.

**Exercício 3** – Programar na classe **Lista** o método **split(nro:int):Lista**. Ele quebra uma lista em duas no elemento que tiver o conteúdo passado como parâmetro.

```
public class ListaCircular {
    No inicio;

    void add(int nro) {
        No no = new No();
        no.conteudo = nro;
        no.anterior = null;
        no.proximo = null;
        if( inicio == null ){
            inicio = no;
            no.proximo = no;
            no.anterior = no;
        }
        else{
            No ultimo = inicio;
            while( ultimo.proximo != inicio ){
                ultimo = ultimo.proximo;
            }
            ultimo.proximo.anterior = no;
            no.proximo = ultimo.proximo;
            no.anterior = ultimo;
            ultimo.proximo = no;
        }
    }

    void print() {
        if( inicio == null ){
            System.out.println("\nlista vazia");
        }
        else{
            System.out.println();
            No ultimo = inicio;
            do{
                System.out.print(
                    ultimo.conteudo + " ");
                ultimo = ultimo.proximo;
            }while( ultimo != inicio );
        }
    }
}
```

**Exercício 4** – Programar na classe **ListaCircular** um método de nome **addInicio(nro:int)** para inserir os

elementos no início da lista, desta forma, o início da lista será sempre deslocado para o nó inserido.

**Exercício 5** – Programar na classe ***ListaCircular*** o método ***split(nro:int):ListaCircular***. Ele quebra uma lista em duas no elemento que tiver o conteúdo passado como parâmetro.