



O objetivo deste exercício é implementar um TAD Lista através de uma lista simplesmente encadeada. Para tal, você deverá construir o seguinte (use **copy** e **paste** para copiar o código):

Pacote Lista

```
class No
{
    private Object elem;
    private No prox;

    public No(Object e, No p)
    {
        elem=e;
        prox=p;
    }
    public Object getElem()
    {
        //insira o código da função
    }
    public No getProx()
    {
        //insira o código da função
    }
    public void setProx(No o)
    {
        //insira o código da função
    }
}
```

```
public class Lista
{
    private int tam=0;
    private No ini=null;
    private No fin=null;
    private No corr=null;

    public boolean vazio()
    {
        //insira o código da função
    }
    public boolean insIni(Object x)
    {
        //insira o código da função
    }
    public boolean insFin(Object x)
    {
        //insira o código da função
    }
}
```



```
    }  
    public Object retIni()  
    {  
        //insira o código da função  
    }  
    public Object retFin()  
    {  
        //insira o código da função  
    }  
    public void posIni()  
    {  
        //insira o código da função  
    }  
    public Object prox()  
    {  
        //insira o código da função  
    }  
}
```

Operações da Classe Lista

public boolean vazio() – retorna **true** se a lista estiver vazia e **false** caso contrário.

public boolean insIni(Object x) – insere o objeto **x** no início da lista. Caso a inclusão seja feita corretamente o método deve retornar **true**, caso contrário deve retornar **false**.

public boolean insFin(Object x) – insere o objeto **x** no final da lista. Caso a inclusão seja feita corretamente o método deve retornar **true**, caso contrário deve retornar **false**.

public Object retIni() – retira o primeiro elemento da lista e retorna uma referência o mesmo. Caso a lista esteja vazia o método deve retornar **null**.

public Object retFin() – retira o último elemento da lista e retorna uma referência o mesmo. Caso a lista esteja vazia o método deve retornar **null**.

public void posIni() – o primeiro elemento da lista passa a ser o elemento corrente (variável **corr**). Caso a lista esteja vazia **corr** irá conter o valor **null** após a execução desta operação.

public Object prox() – retorna o objeto armazenado no nó corrente (variável **corr**) e faz com que a variável **corr** referencie o próximo nó da lista. Caso a variável **corr** contenha o valor **null** esta operação deverá retornar o valor **null**.

Variáveis de Instância da Classe Lista

private int tam – guarda a quantidade de elementos armazenados na lista.



private No ini – guarda uma referência para o primeiro nó da lista. Se a lista estiver vazia **ini** será igual a **null**.

private No fin – guarda uma referência para o último nó da lista. Se a lista estiver vazia **fin** será igual a **null**.

private No corr – guarda uma referência para o nó corrente da lista. Se a lista estiver vazia **corr** será igual a **null**. Antes que os métodos **prox()** e **ant()** possam ser usados pela primeira vez é necessário que o nó corrente seja *inicializado* através dos métodos **posIni()** ou **posFin()**.

Classe Principal

Para testar o TAD Lista, utilize a classe **Vetor**, elaborada no exercício anterior. Insira elementos no início e no final da lista, posicione o cursor no início da mesma (**Lista.posIni()**) e percorra a lista até o final (**Lista.prox()**), imprimindo as coordenadas dos mesmos.

Observação: tente criar um objeto da classe **No** no interior do método **main()**.