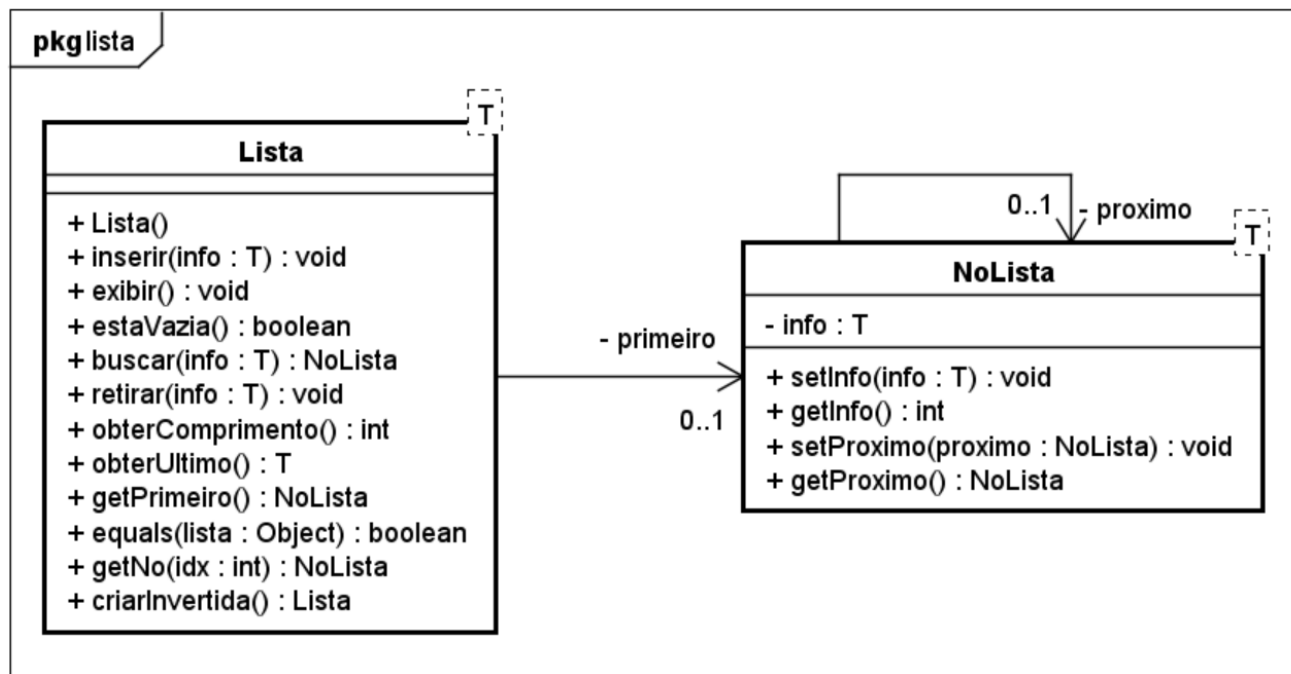


## Lista de Exercícios 02

Partindo da solução da lista de exercícios 01, adapte as classes NoLista e Lista, tornando a implementação de listas encadeadas apta para lidar com quaisquer objetos. Utilize o diagrama abaixo como guia, para implementar os novos métodos.



As duas classes devem compor o pacote Java chamado lista. A descrição dos métodos da classe Lista a serem implementados consta abaixo:

- `getPrimeiro()`: método *getter* da variável `primeiro`;
- `equals()`: Este método deve receber como parâmetro um objeto da classe `Lista` e deve retornar `true` quando a lista fornecida como parâmetro e a lista corrente são iguais, isto é, armazenam os mesmos dados e na mesma ordem. Como o método `equals()` deve sobrescrever o método `equals()` da classe `Object`, você deve declará-lo de forma que receba como parâmetro um objeto da classe `Object`.
- `getNo()`: o método deverá retornar o nó que está na posição fornecida como argumento. Considere que o primeiro nó tem posição igual à 0 e que o último nó (aquele que está na extremidade oposta ao primeiro), está na posição *Comprimento-1*. Caso o argumento fornecido ao método `getNo()` seja negativo ou maior que o comprimento da lista, deverá ser lançada a exceção `java.lang.IndexOutOfBoundsException`.
- `criarInvertida()`: Este método deve criar uma nova lista com os mesmos dados armazenados na lista fornecida como argumento, porém os nós deverão ser inseridos em ordem invertida na nova lista. Isto é, o primeiro nó da lista fornecida como argumento deve ser o último nó da nova lista criada. A nova lista criada deve ser retornada à rotina chamadora.

Após implementar a lista encadeada, crie uma nova classe que contenha o método estático `main()` para demonstrar o funcionamento dos novos métodos da lista encadeada.