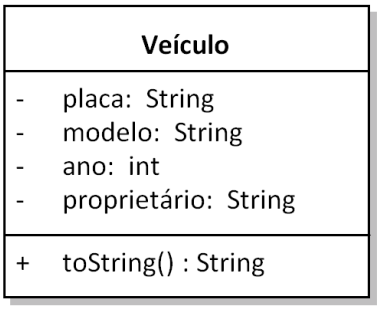
1) Crie um programa para armazenar em `LinkedList`, instâncias da classe `Veiculo`.



Efetue as seguintes operações no seu programa:

a) Inclua 10 instâncias na coleção;
b) Implemente o método toString() para retornar a seguinte frase:
   Veículo <modelo>, placa <placa>, ano <ano>, de <proprietário>
c) Remova instâncias da coleção:
   a. Pelo objeto;
   b. Pela posição: sétimo elemento;
   c. Pelo iterador: antepenúltimo.
d) Defina a "ordem natural" dos objetos como sendo a placa do veículo.
e) Exiba os veículos ordenando-os por número de placa
f) Exiba os veículos ordenando-os por ano e em seguida por placa
g) Exiba os veículos ordenando-os por modelo, em seguida por ano e por último por placa

2) Monte um quadro comparativo entre as coleções e mapas a seguir, que inclua informações sobre:

a) Estrutura de dados de suporte (qual estrutura está implementada internamente?)
b) Duplicação de objetos (permite duplicar objetos?)
c) Valores nulos (permite inserir null?)
d) Métodos de inserção (indicando a posição: no início; qualquer posição; ao final)
e) Métodos de remoção (indicando a posição: no início; qualquer posição; ao final)
f) Métodos de busca de elementos (por posição, chave, objeto)
g) Interfaces JCF implementadas

Uma grande fonte de consulta é a própria documentação do Java (https://docs.oracle.com/en/java/javase/11/).

I. ArrayDeque
II. ArrayList
III. HashSet
IV. LinkedHashSet
V. LinkedList
VI. PriorityQueue
VII. TreeSet
VIII. Vector
IX. HashMap
X. LinkedHashMap
XI. TreeMap

3) Crie um *array* com 100.000 valores inteiros entre 0 e 10.000.000 aleatoriamente. Pode ocorrer repetições. O último valor do *array* deve ser 10.000.001.
   Efetue as seguintes operações no seu programa, manipulando as estruturas como `Collection`:
   a) Inclua os valores do *array* em cada uma das seguintes estruturas:
      a. LinkedList;
      b. ArrayList;
      c. Vector;
      d. HashSet;
      e. LinkedHashSet;
      f. TreeSet;
      g. PriorityQueue;
      h. ArrayDeque.
   b) Exiba o tempo de execução da inclusão dos elementos nas estruturas, juntamente com a quantidade de elementos incluídos em cada uma.
   c) Verifique através do `Iterator` se o valor 10.000.001 está presente em cada uma das estruturas, exibindo o tempo de execução.
   d) Verifique através do método `contains` se o valor 99.000.000 está presente em cada uma das estruturas, exibindo o tempo de execução.
   e) Remova 1 a cada 2 objetos das estruturas, usando `Iterator`.
   f) Exiba o tempo de execução da exclusão dos elementos nas estruturas, juntamente com a quantidade de elementos restantes em cada uma.