Alphabet.java

1. Class - Alphabet is class of the code
   This class represents an abstract concept of alphabets used in string processing.

2. Data Types:
   a. char[] alphabet - to store all the characters in the alphabet.
   b. Int[] inverse- to store the indices
   c. Int R- to store the radix of the alphabet
   d. boolean[] unicode - to store characters max value
   e. String alpha - declaring variable alpha
3. Variables
   a. Static Variables
      i. Examples:
         1. public static final Alphabet BINARY = new Alphabet("01");

      Purpose: Represents the binary alphabet and is shared across all instances.
         2. public static final Alphabet OCTAL = new Alphabet("01234567");
            Purpose: Represents the octal alphabet and is shared across all instances.
         3. public static final Alphabet DECIMAL = new Alphabet("0123456789");
            Purpose: Represents the decimal alphabet and is shared across all instances.
         4. public static final Alphabet BASE64 = new Alphabet("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/");P
            Purpose: Represents the Base64 alphabet and is shared across all instances.
            ……. So on

      Each public static final Alphabet variable represents a commonly used or predefined alphabet (e.g., binary, octal, decimal, etc.).

      static: shared by all instances of Alphabet
      final: immutable
      public: can be accessed outside class

   b. Instance Variables
      i. Examples:
         1. private char[] alphabet
            Purpose: Stores the characters of the specific alphabet instance.
         2. private int[] inverse

Purpose: Stores indices

3. private final int R
   Purpose: radix of the alphabet

4. Methods
   1. public boolean contains(char c)
      Parameters - char c
      Return type- boolean
      Functionality- checks whether the character c is included in alphabet
   2. public int radix()
      Parameters - none
      Return - int
      Functionality- returns size of the alphabets
   3. public int lgR()
      Parameters- none
      Return- int
      Functionality- Returns the binary logarithm of the number of characters in this alphabet.
   4. public int toIndex(char c)
      Params - char c
      Return - int
      Functionality - Returns the index corresponding to the argument character.
   5. public int[] toIndices(String s)
      Params - String s
      Returns - int[]
      Func - Returns the indices corresponding to the argument characters.
   6. public char toChar(int index)
      Params - int index
      Returns - char
      Func - Returns the character corresponding to the argument index.
   7. public String toChars(int[] indices)
      Params - int[] indices
      Returns - String
      Func - returns the characters corresponding to the argument indices.

5. Objects
   1. Alphabet.BINARY
      Purpose: Represents the binary alphabet ("01") for handling two-character encodings (0 and 1).
   2. Alphabet.OCTAL
      Purpose: Represents the octal alphabet ("01234567") for base-8 encoding or decoding tasks.
   3. Alphabet.DECIMAL
      Purpose: Represents the decimal alphabet ("0123456789") for base-10 operations.
   4. Alphabet.HEXADECIMAL

Purpose: Represents the hexadecimal alphabet ("0123456789ABCDEF") commonly used in base-16 encodings.

5. Alphabet.DNA
   Purpose: Represents the DNA alphabet ("ACGT") often used in bioinformatics for handling nucleotide sequences.
6. Alphabet.LOWERCASE
   Purpose: Represents the lowercase English alphabet ("abcdefghijklmnopqrstuvwxyz").
7. Alphabet.UPPERCASE
   Purpose: Represents the uppercase English alphabet ("ABCDEFGHIJKLMNOPQRSTUVWXYZ").
8. Alphabet.PROTEIN
   Purpose: Represents the protein alphabet ("ACDEFGHIKLMNPQRSTVWY") for amino acid codes.
9. Alphabet.BASE64
   Purpose: Represents the Base64 alphabet ("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/") used for encoding and decoding data in a base-64 format.

6. Encapsulation

   private char[] alphabet;

   Encapsulates the underlying character array. By making alphabet private, the class ensures that it cannot be directly modified from outside. Instead, access is provided through public methods (toIndex, toChar, etc.), protecting its integrity and preventing unintended changes.

7. Abstraction

   public int[] toIndices(String s)
   public String toChars(int[] indices)

These methods abstract away the details of how characters map to indices internally (alphabet and inverse). Users of the Alphabet class can simply call toIndices or toChars without needing to understand the low-level indexing mechanism.

8. Constructor

   public Alphabet(String alpha)

   initializes a new Alphabet from a string of characters, ensuring there are no duplicates. It sets up the internal arrays (alphabet and inverse) so each character can be converted to an index and vice versa.

9. Static Blocks or Final Constants

public static final Alphabet DECIMAL = new Alphabet("0123456789");

Purpose:
This predefines a commonly used alphabet—digits 0–9—for efficient reuse. Because it is declared as public static final, it is:
public – accessible from anywhere
static – associated with the Alphabet class itself (not instances)
final – cannot be reassigned, guaranteeing a constant