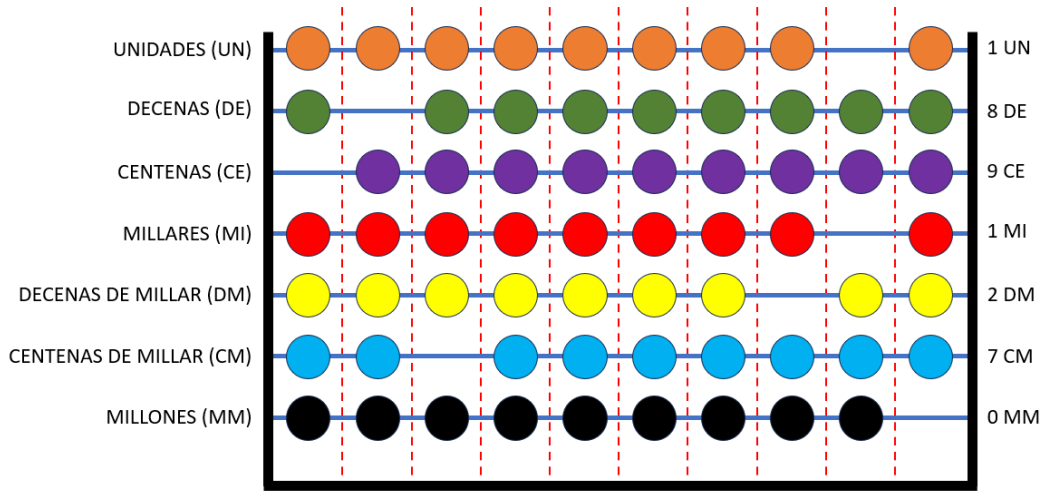
SUPUESTO 3 – PROGRAMACIÓN – ÁBACO	DNI.



Véase la figura que aparece arriba que representa un ábaco con las características que se describen a continuación.

Está compuesto por siete filas, que de arriba abajo representan las unidades, decenas, centenas, millares, decenas de millar, centenas de millar y millones. Cada fila tiene 10 posibles posiciones y 9 bolas. Una posición solo puede estar ocupada por una bola o estar vacia. En cada fila hay siempre un hueco vacio (sin bola) que es el que determina el número entre 0 y 9 que se está representando. Las bolas a la izquierda del hueco vacio no intervienen en la representación del número, sí lo hacen las bolas que hay a la derecha. Si el hueco está en el extremo izquierdo (como en el caso de las centenas en la figura superior), el número que se está representando es el nueve (porque todas las bolas están a la derecha del hueco). Si el hueco está en el extremo derecho (como en el caso de los millones en la figura superior), el número que se está representando es el cero (porque no hay ninguna bola a la derecha del hueco). Si el hueco está en una posición intermedia (como en el caso de las decenas de millar) la cifra que se representa es la cantidad de bolas que hay a la derecha del hueco (para las decenas de millar, hay 7 bolas a la izquierda del hueco, que no cuentan, y dos a la derecha, por lo que el número que se está representando es el dos). En la parte derecha de la figura se indica para cada fila, la cifra que está representando en la posición actual, que junto con la posición que ocupa que se encuentra a la izquierda se obtiene que la cifra que se está representando es: **721.981**

Se pide que analice la clase java que implementa el sistema y que:

1. Obtenga la cifra con la que se inicializa el ábaco (distinta a la que se representa en la figura).
2. Indique las operaciones (máximo siete) en las casillas (Fila, Nº de bolas y Desplazamiento) que le permitan llevar al ábaco de la posición de partida, a la posición final que es la que se representa en la figura superior *(Las columnas Millones, Centenas de millar, Decenas de millar, Millares, Centenas, Decenas y Unidades no es obligatorio rellenarlas, aunque le pueden servir para resolver adecuadamente el problema)*

Núm. Instrucción	Millones	Centenas de millar	Decenas de millar	Millares	Centenas	Decenas	Unidades	Fila	Nº de bolas	Desplazamiento
1										
2										
3										
4										
5										
6										
7										
8	**0**	**7**	**2**	**1**	**9**	**8**	**1**	-	-	-

```java
package examenA2;

import java.util.Scanner;

public class Abaco {
        private char[][] filas;
        public static final int OBJETIVO = 721981;
    public Abaco() {
        this.filas = new char[7][10];
        inicializarAbaco(); // Linea 12
    }
    private void inicializarAbaco() {
        int un = inicializarUnidades(); //Linea 27
        int de = inicializarDecenas(); //Linea 32
        int ce = inicializarCentenas(); //Linea 38
        int mi = inicializarMillares(); //Linea 42
        int dm = inicializarDecenasMillar(); //Linea 47
        int cm = inicializarCentenasMillar(); //Linea 60
        int mm = inicializarMillones(); //Linea 73
        int cifra = un + de + ce + mi + dm + cm + mm;
        for (int i = 0; i < 7; i++) {
            int valor = cifra % 10; // Obtener la última cifra
            cifra /= 10;
            colocarBolas(i, valor); //Linea 137
        }
    }
    private int inicializarUnidades()
    {
        int condicion = 15;
        return (condicion > 10) ? 9 : 2;
    }
    private int inicializarDecenas()
    {
        String texto = "#hashtag#";
        int k = texto.indexOf('#');
        return (k * 10);
     }
    private int inicializarCentenas()
    {
        return (100 * (int) Math.sqrt(49));
    }
    private int inicializarMillares()
    {
        int[] numeros = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
        return (1000 * numeros[1]);
    }
    private int inicializarDecenasMillar()
    {
        try {
                double importe = Double.parseDouble("12,34€");
        }
        catch (NumberFormatException e) {
                return 40000;
        }
        catch (NullPointerException e) {
                return 30000;
        }
        return 10000;
    }
    private int inicializarCentenasMillar()
    {
        String cadena = "Lorem Ipsum sit ex Vestibulum";
        String[] palabras = cadena.split(" ");
        String resultado = "";
        for (String palabra : palabras) {
            if (!Character.isUpperCase(palabra.charAt(0))) {
                resultado = resultado + palabra + " ";
            }
        }
        resultado = resultado.trim();
        return (100000 * resultado.length());
    }
```

```java
    private int inicializarMillones()
    {
        String cadena = "62981363A";
        String patron = "^[0-9]{8}[A-Za-z]$";
        if (cadena.matches(patron))
               return 1000000;
               else return 5000000;
    }
    public void cambiarCifra() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("\nIngrese la fila (UN, DE, CE, MI, DM, CM, MM): ");
        String fila = scanner.nextLine().toUpperCase();
        System.out.println("Ingrese la cantidad de bolas: ");
        int cantidad = scanner.nextInt();
        System.out.println("Ingrese el sentido del desplazamiento (D/I): ");
        char sentido = scanner.next().toUpperCase().charAt(0);
        actualizarCifra(fila, cantidad, sentido); //Linea 92
        mostrarEstadoAbaco(); //Linea 150
    }
    private void actualizarCifra(String fila, int cantidad, char sentido) {
        int indiceFila = obtenerIndiceFila(fila); //Linea 100
        if (sentido == 'D') {
            desplazarDerecha(indiceFila, cantidad); //Linea 114
        } else if (sentido == 'I') {
            desplazarIzquierda(indiceFila, cantidad); //Linea 122
        }
    }
    private int obtenerIndiceFila(String fila) {
        switch (fila) {
            case "UN": return 0;
            case "DE": return 1;
            case "CE": return 2;
            case "MI": return 3;
            case "DM": return 4;
            case "CM": return 5;
            case "MM": return 6;
            default:
                System.out.println("Fila no válida");
                return -1;
        }
    }
    private void desplazarDerecha(int indiceFila, int cantidad) {
        int separador = localizaSeparador(indiceFila); //Linea 130
        if (separador > cantidad - 1) {
               filas[indiceFila][separador] = 'O';
               filas[indiceFila][separador-cantidad] = '-';
        }
        else System.out.println("No hay tantas bolas a la izquierda");
    }
    private void desplazarIzquierda(int indiceFila, int cantidad) {
        int separador = localizaSeparador(indiceFila); //Linea 130
        if (separador + cantidad < 10) {
               filas[indiceFila][separador] = 'O';
               filas[indiceFila][separador+cantidad] = '-';
        }
        else System.out.println("No hay tantas bolas a la derecha");
    }
    private int localizaSeparador(int indiceFila) {
        for (int i = 0; i < 11 ; i++) {
               if (filas[indiceFila][i] == '-')
                              return i;
        }
        return -1;
    }
```

```java
    private void colocarBolas(int indiceFila, int cantidad) {
        // Segmento de las bolas de la izquierda
        for (int i = 0; i < (9-cantidad); i++) {
            filas[indiceFila][i] = 'O';
        };
        // Separador
        int posicionSeparador = 9 - cantidad;
        filas[indiceFila][posicionSeparador] = '-';
        // Segmento de las bolas de la derecha
        for (int i = 0; i < cantidad; i++) {
            filas[indiceFila][i + 10 - cantidad] = 'O';
        }
    }
    private void mostrarEstadoAbaco() {
        for (int i = 0; i < 7; i++) {
            imprimirFila(obtenerNombreFila(i), filas[i]); //Lineas 156 y 163
        }
        System.out.println();
    }
    private void imprimirFila(String fila, char[] segmentos) {
        System.out.print(fila + ": ");
        for (int i = 0; i < 10; i++) {
            System.out.print(segmentos[i]);
        }
        System.out.println();
    }
    private String obtenerNombreFila(int indiceFila) {
        switch (indiceFila) {
            case 0: return "UN";
            case 1: return "DE";
            case 2: return "CE";
            case 3: return "MI";
            case 4: return "DM";
            case 5: return "CM";
            case 6: return "MM";
            default: return "";
        }
    }
    public int obtenerCifra() {
        int cifra = 0;
        for (int i = 0; i < 7; i++) {
            int valorFila = 0;
            int separador = localizaSeparador(i); //Linea 130
            valorFila = 9 - separador;
            for (int j = 0; j < i; j++) {
                valorFila *= 10;
            }
            cifra = cifra + valorFila;
        }
        System.out.print("Número actual: " + cifra + " Número objetivo: " + OBJETIVO);
        return cifra;
    }
    public static void main(String[] args) {
        Abaco abaco = new Abaco();
        abaco.mostrarEstadoAbaco(); //Linea 150
        while (abaco.obtenerCifra() != OBJETIVO) { // Linea 175
            abaco.cambiarCifra(); //Linea 81
        }
        System.out.println("\n¡Felicidades! Has alcanzado la cifra objetivo.");
    }
}
```