

Práctica 3.La clase String

Objetivo: Aplicar los conceptos básicos de la clase String en java.

Introducción:

La clase String es una clase que proporciona el propio lenguaje para el manejo de cadenas de caracteres. Se trata de una clase especial. Puede utilizarse como cualquier otra clase para instanciar objetos que almacenen cadenas de caracteres de la siguiente manera:

```
String st = new String("Esto es una cadena");
```

Sin embargo, un código equivalente al anterior es el siguiente:

```
String st = "Esto es una cadena";
```

Dado que String es una clase especial, los dos códigos anteriores son equivalentes.

Otra particularidad de la clase String es que tiene el operador "+" sobrecargado, es decir, se pueden "sumar" objetos de tipo String dando como resultado una cadena nueva formada por ambas cadenas concatenadas. Por ejemplo:

```
import java.util.Scanner;

public class Clase3 {
    public static void main(String[] args) {

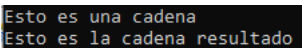
        String st = new String("Esto es una cadena");

        //String st = "Esto es una cadena";

        System.out.println(st);

        String string1 = "Esto es ";
        String string2 = "la cadena resultado";
        String string3 = string1 + string2;
        System.out.println(string3);

    }
}
```

La salida en pantalla sería: 

La clase String es el mecanismo que proporciona Java para la manipulación de cadenas de la misma manera que otros lenguajes no orientados a objetos lo hacen mediante librerías de subrutinas. A través de los métodos de la clase es posible realizar aquellas operaciones más comunes sobre las cadenas de caracteres. Destacan los siguientes métodos:

`length()`: Devuelve la longitud de una cadena. Nota: obsérvese que `length()` es un método de la clase String que no hay que confundir con el atributo `length` de los vectores.

`equals(String st)`: Devuelve un valor lógico indicando si la cadena `st` pasada como argumento es igual o no a aquella sobre la que se llama el método.

`charAt(int i)`: Devuelve el carácter (tipo `char`) correspondiente a la posición `i`, comenzando desde 0 como en las matrices.

Por ejemplo:

```
import java.util.Scanner;

public class Clase3Ejem2 {
    public static void main(String[] args) {

        String animal1 = "Elefante";
        String animal2 = "Leon";
        String animal3 = "Leon";

        System.out.println("El primer animal y el segundo: "+
            (animal1.equals(animal2)? "si":"no")+ " son iguales");

        System.out.println("El segundo animal y el tercero: "+
            (animal2.equals(animal3)? "si":"no")+ " son iguales");

        System.out.println("La tercera letra de " + animal1 + " es la"
            "\"" + animal1.charAt(2) + "\"");

        System.out.println("El numero de letras de \"" + animal2 + "\" es
            " + animal2.length());
    }
}
```

La salida en pantalla sería:

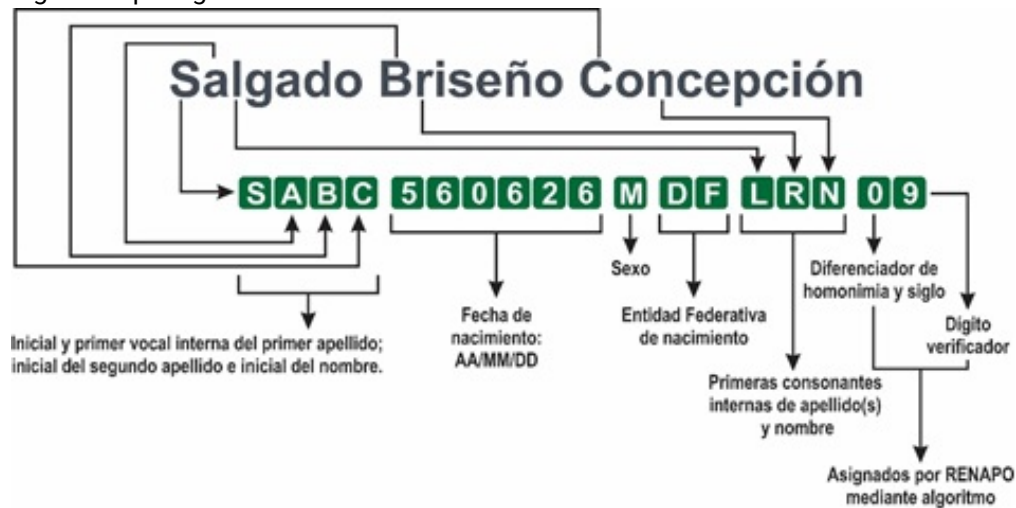
```
El primer animal y el segundo: no son iguales
El segundo animal y el tercero: si son iguales
La tercera letra de Elefante es la "e"
El numero de letras de "Leon" es 4
```

Práctica:

1. Crear una clase llamada "Persona" que contenga un nombre, CURP, y RFC.
2. Implementar un algoritmo para calcular el CURP y RFC de una persona física.
3. Investigar las reglas oficiales para calcular el CURP y RFC a partir de los datos de una persona tales como: Nombre(s), Apellido Paterno, Apellido Materno, Fecha de Nacimiento.

Reglas generales: Para mayor detalle véase el documento [IFAI 0610100135506 065 Algoritmo para generar el RFC con homoclave para personas físicas y morales.pdf](#)

Algoritmo para generar CURP:



4. Los últimos 3 caracteres del RFC correspondiente a la homoclave se generarán de manera aleatoria.
5. Los últimos 2 caracteres del CURP se generarán de manera aleatoria.
6. El programa deberá de solicitar los datos necesarios como nombre y fecha de nacimiento para poder generar el CURP y RFC.
7. El programa deberá crear y mostrar 3 objetos:
Para el objeto 1 solo se generará el CURP.
Para el objeto 2 solo se generará el RFC.
Para el objeto 3 se generarán ambos datos (CURP Y RFC).

Actividades:

- Hacer un reporte con portada de hoja completa que incluya el código fuente, el link de su código en GitHub y una captura de pantalla de la práctica funcionando.
- Si la práctica contiene menú, se agregará una captura de pantalla por cada opción implementada.
- Subir a blackboard código fuente y reporte.