

Nombre: _____ Cédula: _____

Observaciones

- La carpeta que contenga el examen se debe llamar con el nombre del estudiante.
- Salidas forzadas harán que se pierda el doble del valor del método.
- Carpetas dañadas o carpetas vacías harán que se pierda el valor del examen.
- Puede utilizar los métodos incluidos en la aplicación (esVacía(), toString(), longitudLista()) (**No se debe** invocar métodos como: insertar inicio, insertar final, etc., de las clases diseñadas en clase, toda la lógica se debe incluir en el método respectivo).
- Se incluyó el código **insertarInicio()** solamente por si no pudiera llenar la lista. En este caso utilice el método insertarInicio() para llenar la lista de uno en uno con el fin de poder probar los otros métodos. En este caso se perderían los puntos de llenarLista().
- Se atenderán consultas de enunciado vía WhatsApp al **83840300**.
- No se aceptarán exámenes después de la hora de cierre de la asignación.**
- Si se queda sin internet o se suspende el fluido eléctrico, debe notificarlo de inmediato a la profesora vía WhatsApp, no debe esperar al final para indicarlo.

Desarrollo

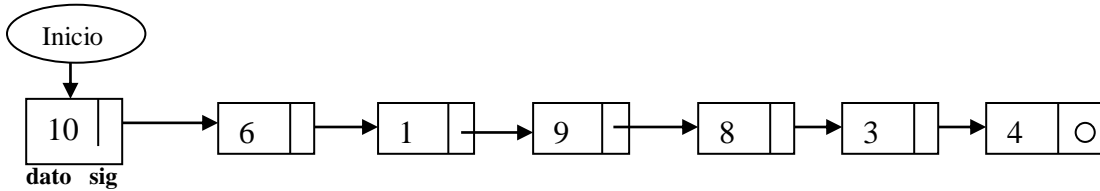
En la aplicación que se le entrega, incluya en las clases **ListaSimple** los métodos **a,b,c,y,d** que se describen a continuación; y en la clase **Recursividad**; el método **e**, según se describe más adelante. Dé la funcionalidad respectiva al frame **FrmListaSimple**.

The screenshot shows a Java Swing window titled 'FrmListaSimple'. It contains a text field with the label 'N' and the value '7'. To the right of the text field are five buttons: 'Llenar Lista', 'Mayor diferencia', 'BuscarNumero', 'Duplicar Nodos', and 'Suma Números (Recursivo)'. Below the text field is a text area with the label 'Lista' and the content '5 2 7 8 3 7 8'. At the bottom right of the window is a 'Salir' button.

- a. **void llenarLista():** Recibe como parámetro un número entero (**n**) que indicará la cantidad de nodos que contendrá la lista. Procede a crear e insertar en la lista **n** nodos de manera que dentro de la lista quede, en cada nodo, un número aleatorio entre 1 y 10.

Observe que cada nodo se debe ir insertando en forma consecutiva después del último nodo incluido. Todos los nodos se crean en una sola ejecución del método.

Ejemplo de la lista si **n = 7**



- b. **int mayorDiferencia():** Retorna la mayor diferencia **en valor absoluto** que existe entre el contenido de dos nodos consecutivos.

Ejemplo:

En la lista anterior se debe retornar un **8**, ya que esta es la mayor diferencia entre dos nodos consecutivos, es decir entre el valor 1 y el valor 9

- c. **int buscarNumero():** recibe como parámetro un número entero y retorna la posición donde se encuentra por primera vez el número dentro de la lista. En caso de no encontrarlo retorna -1. **Asuma que el primer nodo de la lista ocupa la posición cero.**

- d. **void duplicarNodosIntercalados():** este método duplica (multiplica por dos) el dato de cada nodo de la lista, pero de manera que se duplica el contenido del primero, el tercero, el quinto, etc. Es decir, la lista del ejemplo en el punto **a** quedaría:

Lista: 20 6 2 9 16 3 8

Recuerde que la lista puede tener longitud variable, según el tamaño creado en el método **a**

- e. **int sumaNumeros(int n):** En la clase **Recursividad** hacer el método recursivo que recibe como parámetro **n**, calcula y retorna la suma de todos los números que van desde 1 hasta **n**.

Ejemplo:

Suponga que n = 6

Sumar los números de 1 a 6 =

$6 + (5+4+3+2+1)$ -> Sumar 6 más la suma de todos los números de 5 a 1

$6 + 5 + (4 + 3 + 2 + 1)$ -> Sumar 5 más la suma de todos los números de 4 a 1

$6 + 5 + 4 + (3 + 2 + 1)$ -> Sumar 4 más la suma de todos los números de 3 a 1

Y así sucesivamente

Funcionalidad de la pantalla:

- 1- **Botón Llenar Lista:** Invoca al método **llenarLista()**, le envía como parámetro el valor digitado por el usuario en la caja de texto N. Muestra en el objeto **jTextArea**, el **toString()** de la lista. Cada vez que se presione dicho botón inicia una nueva lista. Una vez que muestra el **toString()** de la lista, el focus se debe colocar en la caja de texto N, y seleccionar el contenido.

- Al presionar **Enter** en la caja de texto N, se debe ejecutar el evento del botón “**Llenar Lista**”

- 2- Botón Mayor Diferencia:** Invoca al método **mayorDiferencia**. Muestra en el objeto `TextArea` el `toString()` de la lista y en la siguiente línea, el valor retornado por el método como la mayor diferencia. El focus debe regresar a la caja de texto N y el texto contenido en ella debe estar seleccionado.
- 3- Botón Buscar Número:** Lee el número ingresado en la caja de texto N y muestra en el objeto `TextArea`, el `toString()` de la lista y la posición dentro de la lista donde se encuentra el número o un mensaje indicando que no se encontró el número en la lista. El focus debe regresar a la caja de texto N y el texto contenido en ella debe estar seleccionado. Asuma que el primer nodo ocupa la posición 0.
- 4- Botón Duplicar Nodos:** Muestra el `toString()` de la lista, invoca al método y vuelve a mostrar el `toString()` de la lista (en el siguiente renglón) de manera que se observen los contenidos de los nodos duplicados. El focus debe regresar a la caja de texto N y el texto contenido en ella debe estar seleccionado.
- 5- Botón Suma Números:** Lee el número que el usuario digita en la caja de texto N, invoca al método recursivo `sumaNumeros(n)` y muestra en el `TextArea` el resultado correspondiente. El focus debe regresar a la caja de texto N y el texto contenido en ella debe estar seleccionado.

Tabla de calificación

| | Item | Puntos | Puntos Obtenidos |
|----|---------------------------------------|------------------|-------------------------|
| | <u>Ejercicio 1</u> | | |
| 1 | Compila | 2 | |
| 2 | Ejecuta | 5 | |
| | <u>Subtotal</u> | <u>7</u> | |
| | Capa Lógica | | |
| 3 | Método <code>llenarLista()</code> | 6 | |
| 4 | Método <code>mayorDiferencia()</code> | 6 | |
| 5 | Método <code>buscarNúmero()</code> | 6 | |
| 6 | Método <code>duplicarNodos()</code> | 6 | |
| 7 | Método <code>sumaNumeros()</code> | 6 | |
| | <u>Subtotal</u> | <u>30</u> | |
| | Capa Vista(Gráfica) | | |
| 8 | Botón Llenar Lista | 3 | |
| 9 | Botón mayor Diferencia | 2 | |
| 10 | Botón Buscar Número | 3 | |
| 11 | Botón Duplicar Nodos | 2 | |
| 12 | Botón sumarNumeros | 3 | |
| | <u>Subtotal</u> | <u>13</u> | |
| | TOTAL EXAMEN | <u>50</u> | |

“Tú eres una persona de éxito, piensa y actúa como tal”