Manual Técnico

Almacenamiento y modificación de vectores

Juan David Monroy

CONTENIDO

1. **Introducción………………………………………………….….3** 1. Objetivo………………………………………………….....3

2. Requerimientos……………………………………………..3

1. **Funcionamiento del aplicativo**
2. Ingreso de Información………………………………………(4-5)
3. Funcionalidades………………………………………………(6)
4. **INTRODUCCIÓN**
5. **Objetivo**

1. Crear un programa en cual se implemente un vector aleatorio de tamaño n, cuyo número será dado directamente por el usuario, además deberá preguntar al usuario si este desea llenar el vector en lugar de generarlo aleatoriamente, si la respuesta del usuario es si, se deberá permitirle agregar los números que el desee ingresar, y si el usuario no llena las n posiciones, deberá dar un aviso al usuario que le faltan por llenar ciertas posiciones, si el usuario responde que no las desea llenar, por defecto usted lo rellenara con 0.

2. Se le debe permitir al usuario cambiar elementos en el vector, es decir, el usuario puede dar un índice y un número, usted debe cambiar el numero en dicha posición por el dado por el usuario, además si el desea cambiar el número, también debe proporcionarle la opción de cambiar el numero en dicha posición por un numero aleatorio.

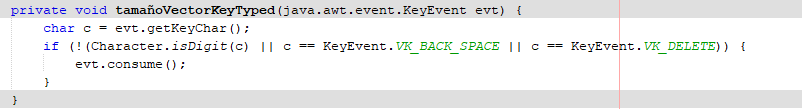
3. Crear un método que permita cambiar la posición de dos elementos del arreglo, es decir, el usuario te dará dos índices, el programa deberá cambiar la posición de ambos elementos. 4. Crear un método que permita invertir el orden del vector, es decir, en el orden en que fueron ingresados, usted deberá ponerlos de manera inversa, vale aclarar que invertir el orden no significa que deba ordenarlos. (Un ejemplo es, el elemento de la posición 1 pasara a la posición n, el elemento de la posición 2 pasara a la posición n-1, y así sucesivamente)

1. **Requerimientos**

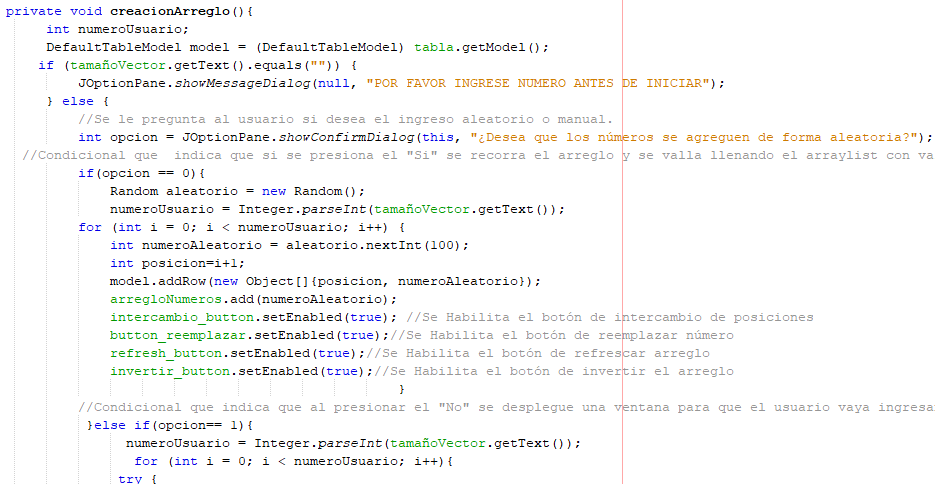
* Netbeans IDE 8.2
* Java.

1. **Funcionamiento del aplicativo**
2. **Ingreso de Información**

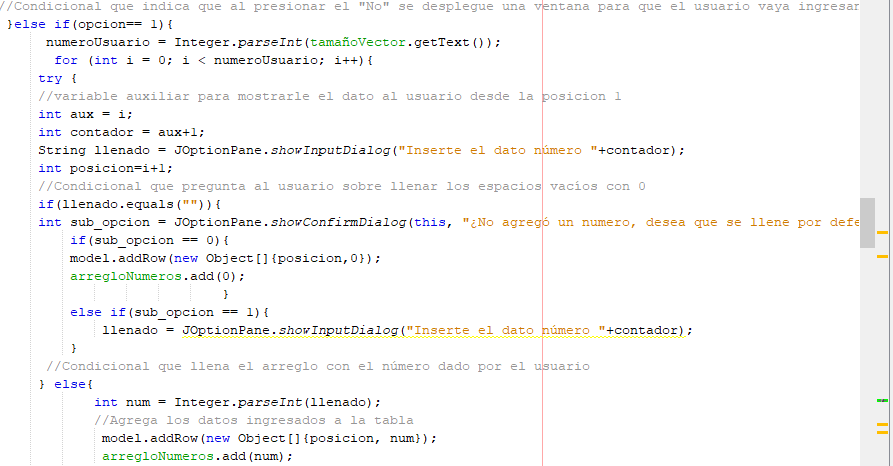
El ingreso de la información del usuario parte de que él ingrese el tamaño del arreglo por medio de un textfield, el dato ingresado es validado por medio de un condicional primero verificando que el campo de texto no esté vacío. El usuario no tiene posibilidad de ingresar caracteres no numéricos en este campo debido a la definición del siguiente método:



El cual apenas detecta el evento de al ser digitado un carácter e inmediatamente lo borra.

Después de validar que ingresó un número, por medio de un JOptionPane se le pregunta al usuario si desea el ingreso de valores aleatorios para llenar el arreglo y por medio de un condicional se valida que si presionó que si (internamente es el valor numérico 0) se genere un nuevo objeto de la clase Random y se empiece a recorrer el arreglo hasta el tamaño dado por el usuario por medio de un ciclo for , añadiéndolos al arreglo y mostrándolos en la tabla de la GUI. Al iniciarse la ejecución del ciclo los botones previamente desactivados para el usuario se activan, debido a que ya se tiene el arreglo con el cual desarrollar las funcionalides.

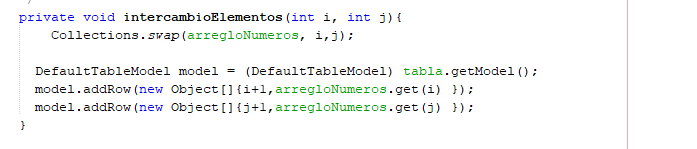
Al usuario presionar el botón no (internamente con el valor numérico 1) se ejecuta igualmente un ciclo pero adicionalmente se declara una variable auxiliar la cual tendrá el valor que se le mostrará al usuario con la posición del arreglo mas 1 para mostrárselo en un JoptionPane en el momento que esté ingresando la información, cada dato será ingresado uno a la vez a la GUI. En el caso que el campo de texto quede vacío se le preguntará al usuario si desea que se le agregue el valor por defecto (valor 0) , y con el uso de un condicional de función parecida al anterior para decidir si el ingreso es aleatorio o no, se procede a llenar la posición con 0 o se vuelve a abrir el menú para que el usuario ingrese el número.



1. **Funcionalidades**

**Intercambio Elementos:**

Para el intercambio de elementos se hace el uso de dos parámetros como ingreso, los cuales representan el índice o la posición del arreglo de las cuales se desea hacer el cambio. En la ejecución del método se utiliza la función Swap de la librería Java Util , que permite dado un arraylist y las dos posiciones hacer el intercambio.



En el instante que el usuario presiona el botón de intercambio se le hace el pedido de ambos índices a cambiar por medio de un JOptionPane, por medio de variables auxiliares se procede a restarle 1 unidad para que coincidan con el tamaño del arraylist, con un condicional se valida que los índices ingresados estén presentes en el arreglo y se procede a realizar el cambio y a informárselo al usuario, por medio de un mensaje como mostrando la nueva información en la tabla.

**Reemplazar Elementos:**

En este método se le solicita al usuario inicialmente el número del índice al cual desea cambiar el valor , se valida que el índice esté en el tamaño del arreglo de una forma similar al del método Intercambiar Elementos y se le pregunta al usuario si desea el ingreso de un valor aleatorio también de una forma similar a la del método anterior. Después de que el número nuevo es ingresado, en la tabla de la interfaz gráfica se muestra el valor con la posición que el usuario solicitó. El método se ejecuta en el momento que el usuario da click al botón “Reemplazar”

**Invertir Array:**

Este método se ejecuta en el instante que el usuario da click en el botón “Invertir valores”. Él método lo que hace es recorrer el arreglo de forma inversa con ayuda de un ciclo for, tomando como valor inicial el tamaño total del arreglo, y validando que mientras este sea mayor que 0 , se vaya restando 1 al valor de i y de forma simultanea se vayan imprimiendo en la tabla los valores que presentes en el arreglo.

