UNIVERSIDAD DEL VALLE DE ORIZABA

UniVO®

Universidad del Valle de Orizaba

Documentación

Lista Enteros

Miguel Angel Tress Ruiz

Docente. Flor Denisse Arenas Cortes

Licenciatura en Ingeniería en Sistemas computacionales

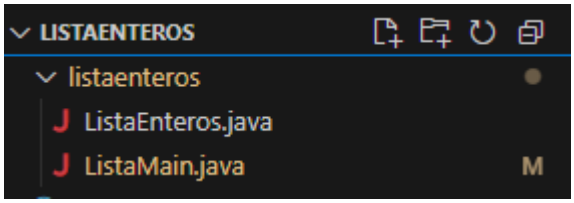
Asignatura. Estructura de Datos

UNIVERSIDAD DEL VALLE DE ORIZABA

Introducción

El sistema ListaEnteros proporciona una implementación encapsulada de una lista de números enteros en Java, utilizando internamente un ArrayList<Integer>. Ofrece operaciones básicas para la manipulación de listas con un enfoque en seguridad y simplicidad.

1. Estructura del paquete



Esta será la organización de nuestro trabajo del cual este se realizará en VSC, las clases se encontrarán en una carpeta llamada “listaenteros” y en este se encuentra la clase “ListaEnteros.java” de la cual en esta se encontrarán todos los métodos básicos y “ListaMain.java” será la clase que servirá como el punto de entrada para la ejecución del programa

2. Clase ListaEnteros

Aquí se proporciona una estructura de datos tipo lista especializada para almacenar exclusivamente valores enteros, con operaciones seguras y métodos convenientes.

```java
package listaenteros;

import java.util.ArrayList;

public class ListaEnteros {
    private ArrayList<Integer> lista;
```

En esta clase se agrega el paquete con el mismo nombre de nuestra carpeta ya que esta ayuda a agrupar a elementos relacionados, se importa la case ArrayList del paquete java.util, se define la clase ListaEnteros para que esta sea publica de igual forma en otras clases

UNIVERSIDAD DEL VALLE DE ORIZABA

3. Constructor

```
public ListaEnteros() {
    lista = new ArrayList<>();
}
```

Se crea un constructor vacío de enteros es necesario para la inicialización de objetos de la clase

4. Métodos públicos

```
// Agrega un elemento al final de la lista
public void agregar(int elemento) {
    lista.add(elemento);
}
```

Como se menciona este método lo que hace es agregar un elemento al final de la lista, nuestro parámetro en este método es “metodo” se agrega int ya que en este caso solo se van almacenar cantidades enteras

```
// Elimina la primera aparición del elemento
public boolean eliminar(int elemento) {
    return lista.remove((Integer) elemento);
}
```

En este metodo se encarga de eliminar la primera ocurrencia del elemento especificado

De igual forma “elemento” será nuestro parámetro ya que esta es una variable que utiliza los datos para pasarlos al metodo cuando sean llamados y se utiliza boolean para verificar que este elemento se encuentre en la lista. En este caso devuelve la lista removiendo el elemento en caso de que se encuentre en esta

```
// Verifica si la lista contiene el elemento
public boolean contiene(int elemento) {
    return lista.contains(elemento);
}
```

Se agrega un metodo en el cual este se encargue de verificar que en la lista si contenga un elemento, nuestro parámetro es el mismo. En este caso nos devolverá una respuesta como false o true en caso de que el elemento que pidamos este, esto es debido que se utiliza boolean

```java
// Obtiene el elemento en el índice dado
public int obtener(int indice) {
    if (indice >= 0 && indice < lista.size()) {
        return lista.get(indice);
    } else {
        throw new IndexOutOfBoundsException(s:"Índice fuera de rango.");
    }
}
```

De igual forma se implementó un metodo de cual este se encarga de mostrar el elemento que se encuentra en el índice que se pida, nuestro parámetro cambia a “índice” ya que este se utiliza para referenciar la posición de un elemento. En este caso debemos agregar una condición en la cual nos devolverá el valor entero en la posición indicada, en caso de que no se agrega otra condición en la cual nos mostrara una leyenda de que el índice esta fuera del rango o es invalido

```java
// Devuelve la cantidad de elementos en la lista
public int tamaño() {
    return lista.size();
}
```

En este metodo se devuelven la cantidad de elementos que hay en la lista, no hay mucho que decir en este caso

```java
// Muestra todos los elementos de la lista
public void mostrar() {
    System.out.println(lista);
}
```

En este metodo se encarga de mostrar todos los elementos que se encuetran en la lista, esto en la salida estándar, con un formato de [elemento1, elemento2, ..., elementoN]

UNIVERSIDAD DEL VALLE DE ORIZABA

5. Clase ListaMain

Aquí se muestra de forma ilustrativa el uso básico de la clase ListaEnteros

6. Flujo de ejecución

1. Crea una lista vacía
2. Agrega los elementos 10, 20 y 30
3. Muestra el contenido inicial
4. Elimina el elemento 20
5. Muestra el contenido actualizado
6. Verifica la existencia del elemento 30
7. Obtiene el elemento en la posición 1
8. Muestra el tamaño final de la lista

```java
package listaenteros;

public class ListaMain {
    Run | Debug
    public static void main(String[] args) {
        ListaEnteros miLista = new ListaEnteros();

        miLista.agregar(elemento:10);
        miLista.agregar(elemento:20);
        miLista.agregar(elemento:30);


        miLista.mostrar(); // [10, 20, 30]

        miLista.eliminar(elemento:20);
        miLista.mostrar(); // [10, 30]

        System.out.println("Contiene 30? " + miLista.contiene(elemento:30)); // true
        System.out.println("Elemento en índice 1: " + miLista.obtener(índice:1)); // 30
        System.out.println("Tamaño: " + miLista.tamaño()); // 2
    }
}
```

Al ejecutar esto es lo que nos mostraría en pantalla:

```
[10, 20, 30]
[10, 30]
Contiene 30? true
Elemento en índice 1: 30
Tamaño: 2
PS C:\Users\Miguel\Listaenteros>
```

Conclusión

El sistema ListaEnteros cumple su propósito al ofrecer una forma sencilla y segura de trabajar con listas de enteros en Java. Gracias a su diseño encapsulado, protege los datos internos y expone solo las operaciones necesarias, siguiendo buenas prácticas de POO.