Operaciones básicas en un Array dinámico

¡Comencemos!

¿Qué aprenderás?

- Comprender la documentación de la clase ArrayList para hacer uso de sus métodos y codificar de manera rápida.
- Aplicar operaciones de un array dinámico para “agregar”, “eliminar”, “ordenar”, “contar” para conocer los métodos esenciales de un ArrayList.

Introducción

Los arreglos tienen muchos métodos que nos permiten realizar operaciones básicas como, por ejemplo, saber si un elemento está dentro de un arreglo, agregar elementos, borrar elementos y contar elementos. Por lo mismo es importante saber leer e interpretar su documentación.

Los arreglos dinámicos nos ayudan a tener una imagen a grandes rasgos de cómo es el comportamiento de una base de datos.

Para tener más información se recomienda acceder a Arreglos dinámicos (ArrayList) ubicado en “Material Complementario”.

¡Vamos con todo!

Comprender e interpretar la documentación de la clase ArrayList

Primero, vemos a qué módulo y paquete pertenece la clase ArrayList, cómo se define una descripción, entre otros.

Luego, vienen las definiciones de los constructores y todos los métodos que tiene la clase.

A continuación se detalla cada uno de sus constructores y métodos:

Agregar un elemento

Modificador y Tipo	Método y descripción.
boolean	add(E e) Agrega un elemento específico al final de la lista.
void	add(int index, E element) Inserta un elemento específico en la posición indicada.

Tabla 1. Agrega elementos a un ArrayList.
Fuente: Desafío Latam.

El método principal para agregar elementos a una Arraylist es `add`, a medida que se vayan agregando, el índice va aumentando de forma secuencial, partiendo como valor inicial en cero.

```
ArrayList <String> a = new ArrayList <String> ();
a.add(1);
a.add(2);
a.add(3);
```

```
System.out.println(a);
```

A diferencia de los array estáticos, el `ArrayList` permite mostrar todos sus elementos sin recorrer.

Salida del ArrayList

```
[1,2,3]
```

Ejercicio guiado: Agregar elementos a un arreglo

Dado un arreglo llamado "ingredientes" se nos pide crear un programa donde el usuario pueda consultar si un ingrediente existe en la pizza, y si no existe debe ser añadido a la lista de ingredientes.

```
ingredientes // [piña, jamón, salsa, queso]
```

- ***Paso 1:*** Importar la clase ArrayList

```
import java.util.ArrayList;
```

- ***Paso 2:*** Crear un ArrayList de tipo String llamado "ingredientes" el cual se le agrega los valores por defectos citados en el ejercicio.

```
ArrayList<String> ingredientes = new ArrayList<String> ();
ingredientes.add("piña");
ingredientes.add("jamón");
ingredientes.add("salsa");
ingredientes.add("queso");
```

- ***Paso 3:*** Crear un objeto Scanner el cual nos permitirá leer los datos ingresados por consola.

```
Scanner sc = new Scanner(System.in);
String ingrediente = sc.nextLine();
```

- ***Paso 4:*** En la condición if, preguntar si el ingrediente ingresado está en el arreglo, si esto existe, mostramos por consola el mensaje correspondiente.

```
if(ingredientes.contains(ingrediente)){
 System.out.printf("El ingrediente ya se encuentra dentro de la
pizza\n");
}
```

- ***Paso 5:*** Si el ingrediente no existe dentro del arreglo se mostrará el mensaje correspondiente más el arreglo completo.

```
else {
 ingredientes.add(ingrediente);
 System.out.printf("El ingrediente %s fue
agregado\n",ingrediente);
}
System.out.println(ingredientes);
```

Solución completa

```
// Paso 1
import java.util.ArrayList;
// Paso 2
ArrayList<String> ingredientes = new ArrayList<String> ();
ingredientes.add("piña");
ingredientes.add("jamón");
ingredientes.add("salsa");
ingredientes.add("queso");
// Paso 3
Scanner sc = new Scanner(System.in);
String ingrediente = sc.nextLine();
// Paso 4
if(ingredientes.contains(ingrediente)){
 System.out.printf("El ingrediente ya se encuentra dentro de la
pizza\n");
}
// Paso 5
else {
 ingredientes.add(ingrediente);
 System.out.printf("El ingrediente %s fue
agregado\n",ingrediente);
}
System.out.println(ingredientes);
```

Si agregamos el elemento "champiñon", esta sería la salida:

```
El ingrediente champiñón fue agregado
[piña, jamón, salsa, queso, champiñón]
```

Remover elementos

Existen muchas formas de eliminar uno o varios elementos de un Arraylist.

Eliminar todos los elementos de un arreglo

Si queremos eliminar todos los elementos del arreglo, usaremos el método clear();

```
// Crear arreglo nombres
ArrayList<String> nombres = new ArrayList<String> ();

// Añadir “Juan” al arreglo nombres
nombres.add("Juan");

// Eliminar el arreglo nombres
nombres.clear();

//Imprimir mensaje de salida
System.out.println(" Valores en el arreglo" + nombres);
```

```
Valores en el arreglo []
```

Eliminar elemento según índice

El método `remove(int index)` es el encargado de eliminar elementos dentro del ArrayList según su índice como parámetro de entrada.

```
ArrayList<String> nombres = new ArrayList <String>();
nombres.add("Juan");
nombres.add("Pedro");
nombres.add("Luis");
nombres.remove(1); // "Pedro"
System.out.println(nombres);
```

Cuando realizamos esta operación, el valor de retorno del método es el valor eliminado, por lo que podríamos hacer lo siguiente:

Eliminar el elemento que coincida con el valor entregado.

¿Qué pasa ahora si entregamos el valor del elemento que queremos eliminar?

Si queremos eliminar un elemento, por ejemplo, el valor "a".

En este caso, el método `remove()` nos retornará `true` o `false`, si se hizo o no la eliminación.

```
ArrayList<String> a = new ArrayList <String>();
a.add("a");
a.add("b");
a.add("c");
a.add("d");
System.out.println(a);
String borrado = a.remove(1); // "b"
System.out.println(a);
System.out.println("Elemento borrado: " + borrado);
```

```
[a, b, c, d]
[a, c, d]
```

```
Elemento borrado: b
```

Como dato importante, antes de eliminar un elemento necesitas validar si existen elementos dentro del arreglo, esto para que el índice que se desea borrar al menos exista y el programa no se caiga .

Eliminar elemento que coincida con el valor entregado

```
ArrayList<String> a = new ArrayList <String>();
a.add("a");
a.add("b");
a.add("c");
a.add("d");
System.out.println(a); //[a,b,c,d]
a.remove("a");
System.out.println(a); //[b,c,d]
```

```
[a, b, c, d]
[b, c, d]
```

En este caso, el método `remove()` nos retornará `true` o `false`, dependiendo si se realiza o no la eliminación.

¿Qué pasa si tenemos más de un elemento con el mismo valor?

```
ArrayList<String> a = new ArrayList <String>();
a.add("a");
a.add("b");
a.add("c");
a.add("c");
a.add("c");
a.add("c");
a.add("a");
System.out.println(a); //[a, b, c, c, c, c, a]
a.remove("a");
System.out.println(a); //[b, c, c, c, c, a]
```

```
[a, b, c, c, c, c, a]
[b, c, c, c, c, a]
```

Va a eliminar solo la primera ocurrencia de este.

Eliminar todos los elementos dentro de una colección

Si queremos eliminar todos los elementos que coincidan con "a" del ejemplo anterior, tenemos el método `public boolean removeAll(Collection c);` que recibe como parámetro una colección.

Por lo que podemos crear un ArrayList con los elementos que queremos eliminar de nuestro arreglo:

```
ArrayList<String> a = new ArrayList <String>();
a.add("a");
a.add("b");
a.add("c");
a.add("c");
a.add("c");
a.add("c");
a.add("a");
a.add("d");
System.out.println(a); //[a, b, c, c, c, c, a, d]
ArrayList<String> elementosABorrar = new ArrayList<String>();
elementosABorrar.add("a");
elementosABorrar.add("c");
a.removeAll(elementosABorrar);
System.out.println(a); //[b, d]
```

```
[a, b, c, c, c, c, a, d]
[b, d]
```

Ejercicio guiado: Agregar número par

Crear un método que permita agregar solo números pares a un ArrayList y mostrar el o los elementos del ArrayList.

Paso 1: Se crea un método llamado `agregarNumeroPar` que recibe como parámetro de entrada un número de tipo entero.

Paso 2: Crear una variable local de tipo ArrayList llamada números.

Paso 3: Realizamos la condición if para validar si el número ingresado es un número par.

Paso 4: Si la condición se cumple, agregamos el elemento al ArrayList.

Paso 5: Mostramos el resultado con la sentencia `System.out.println`.

```
public static void main(String[] args) {
            agregarNumeroPar(3);
      }

public static  void agregarNumeroPar(int numero) {
            ArrayList<Integer>  numeros = new ArrayList<Integer>();
            if(numero%2 == 0) {
                  numeros.add(numero);
            }
            System.out.println(numeros);
      }
```