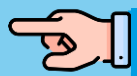
Arreglos y archivos

Otros métodos utilizados en ArrayList

{desafío} latam_

Utilizar la sintaxis básica del lenguaje Java para la construcción de programas que resuelven un problema de baja complejidad

- Unidad 1: Flujo, ciclos y métodos
- Unidad 2: Arreglos y archivos
- Unidad 3: Programación orientada a objetos
- Unidad 4: Pruebas unitarias y TDD



Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

- *Aplicar métodos importantes como size(), sort(), entre otros, para manejar fácilmente volúmenes de información.*

¿Qué otros métodos de ArrayList pueden existir?

/* Otros métodos utilizados en ArrayList */

ArrayList

Algunos métodos

- Declaración del objeto ArrayList
  ```
  private ArrayList<TipoDeObjetosEnLaColección> NombreDelArrayList;
  ```
- Creación de un objeto
  ```
  NombreDeObjeto = new ArrayList<TipodeObjetosEnLaColección>();
  ```
- Reemplazar objeto existente
  ```
  NombreDelArrayList.set (int índice, E elemento);
  ```
- Añadir al final
  ```
  NombreDelArrayList.add (objeto_a_añadir);
  ```
- Obtener el número de objetos en la lista
  ```
  NombreDelArrayList.size();
  ```
- Extraer un objeto de cierta posición
  ```
  NombreDelArrayList.get (posición);
  ```

Reemplazar elementos según índice

Método `set(int index, E element);`

```
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.set(1, "k");
System.out.println(a); //[a, k, c, d]
```

```
String elementoCambiado = a.set(0,"j");
System.out.println("elemento cambiado" + elementoCambiado);
elemento cambiado a
```

Contar elementos

Método `size();`

```
ArrayList<String> a = new ArrayList <String>();
a.add("a");
a.add("b");
a.add("c");
a.add("d");
System.out.println(a.size()); //4
```

Importar ArrayList

En Java podemos aplicar otros métodos sobre los ArrayList que permiten facilitar el trabajo al momento de operar sobre los elementos.

Para estas operaciones utilizaremos la librería.

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

Ordenar los elementos

Método `sort();`

```
ArrayList<String> paises = new ArrayList<String>();
paises.add("Chile");
paises.add("Argentina");
paises.add("Colombia");
paises.add("Perú");
paises.add("Venezuela");
Collections.sort(paises);
System.out.println(paises); //[Argentina, Chile, Colombia, Perú,Venezuela]
```

```
[Argentina, Chile, Colombia, Perú, Venezuela]
```

Ordenar los elementos

Método `sort();`

Agregaremos un nuevo elemento a la lista, pero con minúscula la primera letra y llamaremos al método sort().

```
paises.add("chile");
Collections.sort(paises);
System.out.println(paises);
```

```
[Argentina, Chile, Colombia, Perú, Venezuela, chile]
```

Ordenar los elementos

Método `sort();`

Podemos ver que “chile”, luego de ordenar el arreglo, queda al final cuando debería haber quedado en la posición 1.

Para solucionar esto escribiremos lo siguiente:

```
Collections.sort(paises,String.CASE_INSENSITIVE_ORDER);
System.out.println(paises);
```

```
[Argentina, Chile, chile, Colombia, Perú, Venezuela]
```

¿Y si ahora queremos tener el orden descendente?

Invertir los elementos

Método `reverse();`

```
Collections.reverse(paises);
System.out.println(paises);
```

```
[Venezuela, Perú, Colombia, chile, Chile, Argentina]
```

Obtener el mínimo y máximo valor

- **min ():** Retorna el valor mínimo dentro del ArrayList.
- **max():** Retorna el valor máximo dentro del ArrayList.

```
ArrayList<Integer> numeros = new ArrayList<Integer>();
numeros.add(5);
numeros.add(1);
numeros.add(4);
numeros.add(1);
numeros.add(2);
numeros.add(6);
System.out.println(Collections.min(numeros)); //1
System.out.println(Collections.max(numeros)); //6
```

Obtener la frecuencia de un elemento

Método `frequency();`

```
System.out.println(Collections.frequency(numeros, 1)); //2
```

Ejercicio guiado

Lista de platos

Crear un método llamado “ordenar” que nos permita ordenar alfabéticamente una lista de platos de un restaurante, también se debe mostrar lista ordenada por pantalla.

Esta lista cuenta con los siguientes datos:

- Cazuela.
- Porotos.
- Pastel de Choclo.
- Ají de gallina.
- Ceviche.
- Arepas.

Lista de platos

PASO 1:
Creamos el método estático llamado ordenar.

```
public static void ordenar() {
}
```

Lista de platos

PASO 2:
Se crea una variable local de tipo ArrayList llamada lista.

```
public static void ordenar() {
    ArrayList<String> lista = new ArrayList<String>();
}
```

Lista de platos

PASO 3:

Agregamos cada elemento al arreglo.

```
public static void ordenar() {
    ArrayList<String> lista = new ArrayList<String>();
    lista.add("Cazuela");
    lista.add("Porotos");
    lista.add("Pastel de Choclo");
    lista.add("Ají de Gallina");
    lista.add("Ceviche");
    lista.add("Arepas");
}
```

Lista de platos

PASO 4:
Utilizamos el método sort para ordenar la lista.

```
Collections.sort(lista);
```

Lista de platos

PASO 5:
Se muestra por consola la lista.

```
System.out.println("La lista de comida es " + lista);
```

Lista de platos

Solución completa

```
public static void main(String[] args) {
      ordenar();
}
// Paso 1
public static void ordenar() {
      // Paso 2
      ArrayList<String> lista = new ArrayList<String>();
      // Paso 3
      lista.add("Cazuela");
      lista.add("Porotos");
      lista.add("Pastel de Choclo");
      lista.add("Aji de Gallina");
      lista.add("Ceviche");
      lista.add("Arepas");
      // Paso 4
      Collections.sort(lista);
      // Paso 5
      System.out.println("La lista de comida es " + lista);
}
```

¿Qué hace el método `sort()`?

Con el método `size()` de un Arraylist

¿Cuál es el tamaño de la siguiente declaración?

```
ArrayList<String> números= new ArrayList <String>();
números.add("Uno");
números.add("Dos");
números.add("Tres");
números.add("Cuatro");
números.add("Cinco");
```

Próxima sesión...

- *Comprender la interfaz Iterator y sus principales métodos para tener una mejor claridad del uso y lo que nos facilita como programador.*
- *Aplicar las distintas operaciones de un arreglo para “buscar”, “agregar”, “eliminar” y “mostrar” agilizando el manejo y dando utilidad al interfaz Iterator.*

{desafío}
latam_

Academia de
talentos digitales