

Ev_4_Teoria de colecciones

Kevin Andrés Castrillón Gutiérrez

Sena Complejo Sur Calatrava

Análisis y Desarrollo de Software

2694526

Edwar Velasquez

Fecha 02/11/2023



1. ¿Qué son las colecciones en Java?

Una colección representa a un grupo de objetos, estos objetos serán conocidos como elementos, gracias a esta podemos almacenar cualquier tipo de objeto y utilizar los métodos comunes (Añadir, eliminar, obtener el tamaño de la colección, etc)

2. Realice una tabla con los tipos de datos usados en colecciones

Tipos de datos	
Primitivos	Objetos
Int	Integer
double	Double
long	Long
char	Character
boolean	Boolean
String ya es un objeto, lo cual no lo hace un dato tipo primitivo.	

3. ¿Cuál es la diferencia entre un arreglo y una colección?

La principal diferencia es que las colecciones crecen de manera dinámica a medida que se van agregando campos

4. ¿Qué es un framework?

Es un marco de trabajo el cual va a contener un conjunto estandarizado de conceptos y prácticos, lo cual nos va a ayudar a tener un código más estructurado y organizado

5. ¿Cuáles son las clases de JAVA COLLECTIONS?

- ArrayList
- Linkedlist
- Hashset
- Tresset
- Linkedhashset

6. ¿En qué colecciones nos vamos a centrar en esta guía?

Nos vamos a centrar en las listas, conjuntos y mapas.

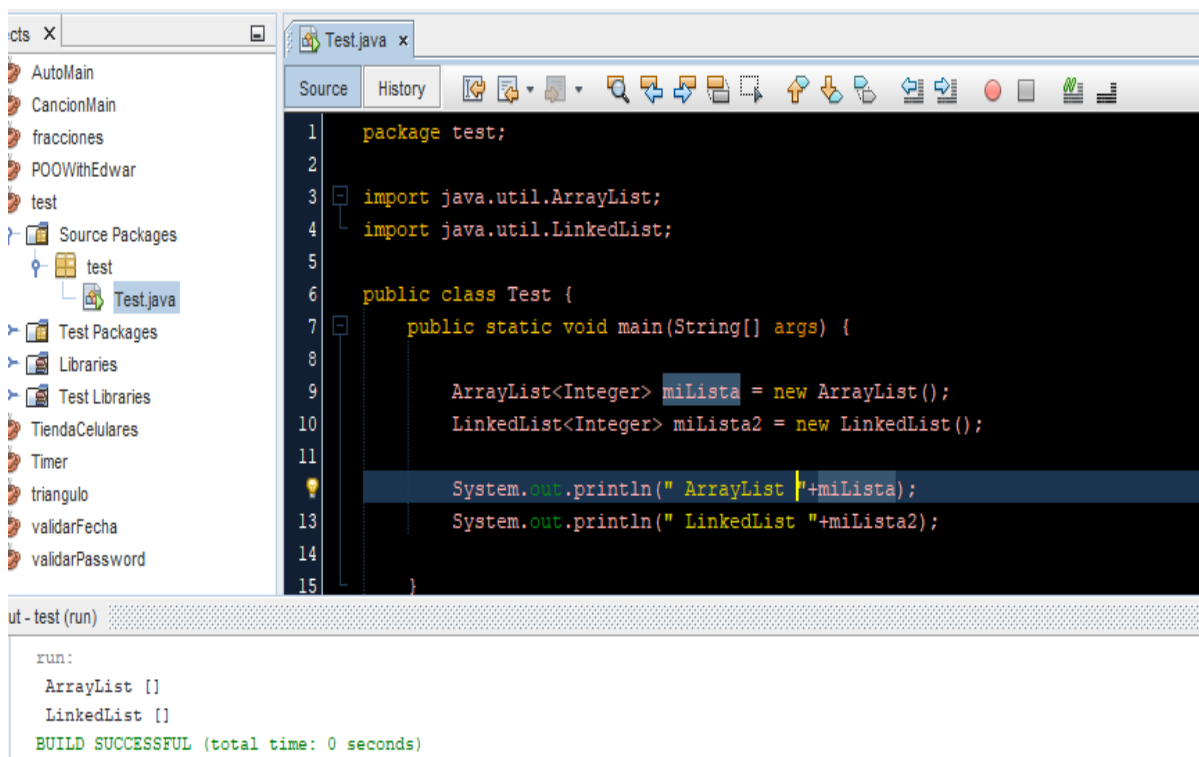
7. Liste las características de las LISTAS.

- Puede guardar elementos duplicados.
- Elegir en qué lugar se va a colocar cada elemento.
- Un control más preciso.

8. ¿Qué es una lista de doble enlace?

Es mucho más rápida cuando debe buscar un elemento de atrás hacia adelante, pero se vuelve más lenta cuando se utiliza el get o set.

9. Realice el ejemplo de la página 5 en Netbeans, le tomas un pantallazo.



```

1 package test;
2
3 import java.util.ArrayList;
4 import java.util.LinkedList;
5
6 public class Test {
7     public static void main(String[] args) {
8
9         ArrayList<Integer> miLista = new ArrayList();
10        LinkedList<Integer> miLista2 = new LinkedList();
11
12        System.out.println(" ArrayList "+miLista);
13        System.out.println(" LinkedList "+miLista2);
14
15    }
16 }

```

run:

```

ArrayList []
LinkedList []
BUILD SUCCESSFUL (total time: 0 seconds)

```

10. ¿Qué son los conjuntos o set en collections?

La colección o conocido como (set), son una collections de objetos donde los elementos solamente se ven una sola vez, no pueden ser duplicados a diferencia de las listas

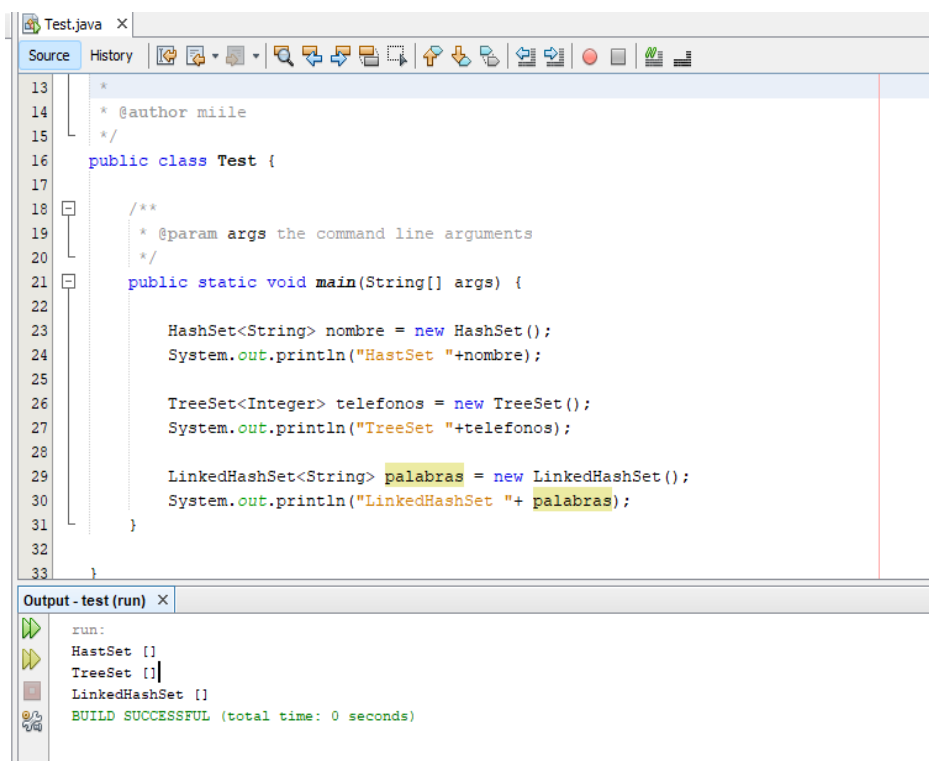
11. ¿Qué es un Hash?

Este se conoce como un algoritmo matemático donde convierte un bloque arbitrario de datos (una lista que aumenta constantemente, también conocida como blockchain), convirtiendo se en caracteres alfanuméricos (letras y números combinados), sin importar la longitud de sus datos.

12. Diferencia entre HashSet y TreeSet

Diferencias	
HashSet	TreeSet
No se mantienen sus elementos de manera ordenada	Se mantienen todos sus elementos de manera ordenada
Sus métodos de agregar, eliminar son mucho más rápidos y eficientes	Sus métodos son mucho más lentos a la hora de utilizarlo, debido a que los elementos los organiza de modo ordenado
Transforma cualquier bloque arbitrario en una serie de caracteres alfanuméricos	Su estructura es implementada en forma de árbol

13. Realice el ejemplo de la página 6 en Netbeans, le tomas un pantallazo



```

13  *
14  * @author miile
15  */
16  public class Test {
17
18      /**
19       * @param args the command line arguments
20       */
21      public static void main(String[] args) {
22
23          HashSet<String> nombre = new HashSet();
24          System.out.println("HastSet "+nombre);
25
26          TreeSet<Integer> telefonos = new TreeSet();
27          System.out.println("TreeSet "+telefonos);
28
29          LinkedHashSet<String> palabras = new LinkedHashSet();
30          System.out.println("LinkedHashSet "+ palabras);
31      }
32
33  }

```

Output - test (run) ×

```

run:
HastSet []
TreeSet []
LinkedHashSet []
BUILD SUCCESSFUL (total time: 0 seconds)

```

14. ¿Qué son los mapas?

Los mapas son aquellos que modelan a un objeto a través de una llave o un valor, es decir, con la llave única vamos a poder identificar qué representa ese valor (las llaves no pueden repetirse), pero sus valores sí, al crear el mapa sabemos que el primer dato es la llave y el segundo el valor.

15. Diferencia entre TreeMap y LinkedHashMap

El TreeMap ordena los elementos de manera ascendente a través de las llaves, en cambio, el LinkedHashMap conserva el orden de inserción.

16. Digita en netbeans el ejemplo de sintaxis de la página 7 y realiza la detección de errores también.

```

    * To change this template file, choose Tools | Templates
    * and open the template in the editor.
    */
package test;

import java.util.HashMap;
import java.util.LinkedHashMap;
import java.util.TreeMap;

public class Test {

    public static void main(String[] args) {
        HashMap<Integer, String> personaA = new HashMap();

        TreeMap<Integer, String> personaB = new TreeMap();

        LinkedHashMap<Integer, String> PersonaC = new LinkedHashMap();

import java.util.ArrayList;
import java.util.HashMap;
import java.util.TreeSet;

public class Test {

    public static void main(String[] args) {

        ArrayList<Integer> listado = new ArrayList();

        TreeSet<String> nombres = new TreeSet();

        HashMap<Integer, String> personas = new HashMap();

    }

```

17. ¿Cómo añadimos elementos a una colección?

Las colecciones cuentan con funciones distintas para realizar operaciones.

Primera forma: Si queremos añadir elementos a las listas o a conjuntos se utiliza:

`add(T).`

Segunda forma: Si solamente queremos añadirlo en los mapas se utiliza:

`put(llave, valor).`

18. Realiza el ejemplo de la página 8 (Añadir elementos a una lista, conjunto y mapa) toma el pantallazo

```
import java.util.HashSet;

public class Test {

    public static void main(String[] args) {

        ArrayList<Integer> numerosA = new ArrayList();
        int num1 = 12;
        numerosA.add(num1);

        HashSet<Integer> numerosB = new HashSet();
        int ident = 1011;
        numerosB.add(ident);

        HashMap<Integer, String> personas = new HashMap();
        int dni = 1011590183;
        String nombreAlumno = "Babano";

        personas.put(dni, nombreAlumno);
    }
}
```