**NOMBRES Y APELLIDOS DE LOS ESTUDIANTES DE CADA EQUIPO**

Consulta:

1. En Java, ¿Qué son los conjuntos?
2. Existen dos (2) tipos de implementaciones para los conjuntos en Java, ¿Cuáles son?, y de una pequeña explicación y ejemplo.
3. Ya conocemos la clase “Set” pero sabes, ¿qué existe una interfaz “Set” ?, describe con tus palabras todo lo que entiendas de esta.
4. ¿Qué operaciones de conjuntos existen que se puedan realizar con los ArrayList? y ¿con qué clases se utilizan?
5. ¿Qué hacen los métodos “equals()”, “removeAll(Collection c)” en la clase “Set”?
6. Desarrolle los siguientes Ejemplos, de una breve explicación de **cada línea** de que hacen cada uno, Documente las líneas que agregue.
7. Imprima pantallazos de la ejecución de cada uno de los siguientes ejemplos

Ejemplo-1

import java.util.List;

import java.util.Arrays;

import java.util.HashSet;

import java.util.Set;

import java.util.Collection;

public class PruebaSet {

private static final String colores[] = {"rojo", "blanco", "azul", "verde", "gris", "naranja", "carne", "blanco", "cyan", "gris", " naranja"};

public PruebaSet() {

List<String> lista = Arrays.asList(colores);

System.out.printf("ArrayList: %s \n", lista);

imprimirSinDuplicados(lista);

}

private void imprimirSinDuplicados(Collection<String> coleccion) {

Set<String> conjunto = new HashSet<String>(coleccion);

System.out.println("\nLos valores sin duplicados son: ");

for (String s: conjunto) {

System.out.printf ("%s", s);

}

System.out.println();

}

public static void main(String args[]) {

new PruebaSet();

}

}

Ejemplo-2

import java.util.Arrays;

import java.util.SortedSet;

import java.util.TreeSet;

public class PruebaSortedSet {

private static final String nombres[] = {"amarillo", "verde", "Negro", "carne", "gris", "blanco", "naranja", "rojo", "verde"};

public PruebaSortedSet() {

SortedSet<String> arbol = new TreeSet<String>(Arrays.asList(nombres));

System.out.println("conjunto ordenado:");

imprimirConjunt(arbol);

System.out.print("\nheadSet(\"naranja\"):");

imprimirConjunto(arbol.headSet("naranja"));

System.out.print("tailSet(\"naranja\"):");

imprimirConjunto(arbol.tailSet("naranja"));

System.out.printf("primero: %s \n", arbol.first());

System.out.printf("ultimo: %s \n", arbol.last());

}

private void imprimirConjunto(SortedSet<String> conjunto) {

for (String s: conjunto) {

System.out.print(s + "");

}

System.out.println ();

}

public static void main(String args[]) {

new PruebaSortedSet();

}

}

EJEMPLO 3

/\* Ejemplo Interface Set y SortedSet, clase HashSet y TreeSet \*/

import java.util.Objects;  
  
public class **Persona** implements **Comparable**<Persona>{  
    private int idPersona;       private String nombre;       private int altura;  
  
    public **Persona**(int idPersona, String nombre, int altura) {  
        this.idPersona = idPersona;   this.nombre = nombre;    this.altura = altura;}  
  
    @Override  
    public String **toString**() { return "Persona-> ID: "+idPersona+" Nombre: "+nombre+" Altura: "+altura+"\n";  }  
  
    @Override  
    public int **compareTo**(Persona o) { return this.altura-o.altura; }  
  
    @Override  
    public int **hashCode**() { return altura + nombre.hashCode() + idPersona; }  
  
    @Override  
    public boolean **equals**(Object obj) {  
        if (obj == null) {   return false;  }  
        if (getClass() != obj.getClass()) {  return false;   }  
        final Persona other = (Persona) obj;  
        if (this.idPersona != other.idPersona) {  return false;    }  
        if (!Objects.equals(this.nombre, other.nombre)) {    return false;     }  
        if (this.altura != other.altura) {     return false;   }  
        return true;  
    }  
}

EJEMPLO 4

/\* Ejemplo Interface Set y SortedSet, clase HashSet y TreeSet \*/

import java.util.HashSet;

import java.util.Set;

import java.util.SortedSet;

import java.util.TreeSet;

public class **Programa**{

    public static void **main** (String []args)      {

        Set<Persona> cjsp = new HashSet<Persona>();

        SortedSet<Persona> cjssp = new TreeSet<Persona>();

        Persona p = new Persona(1,"Marta",165);

        cjsp.add(p);

        cjssp.add(p);

        p = new Persona(2,"Elena",155);

        cjsp.add(p);

        cjssp.add(p);

        p = new Persona(3,"María",170);

        cjsp.add(p);

        cjssp.add(p);

        System.out.println("Personas en el conjunto: "+cjsp);

        System.out.println("Personas en el conjunto ordenado: "+cjssp);

    }

}