

Estructura de Datos

Amy Cárdenas Silva

GRupo: 1360

Carrera: ingenieria en computacion

Tarea 2, muestra de funcionamiento

```
import java.util.List;

public class Main {
    public static void main(String[] args) {
        /*
        System.out.println("hola mundo");

        ArrayADT nuevo = new ArrayADT(5);

        nuevo.establecerElemento(0, 100);

        System.out.println(nuevo.obtenerElemento(0));

        System.out.println(nuevo.longitud());

        nuevo.imprimir();

        nuevo.toString();
        */

        ConjuntoADT setA = new ConjuntoADT();
        ConjuntoADT setB = new ConjuntoADT();

        List<Integer> listaA = List.of(6,6,1,2,3,4,5,1,5,3,2,4,12,56,78);
        for (Integer item: listaA){
            setA.agregar(item);
        }
        System.out.println("el conjunto a es = " + setA.toString());

        List<Integer> listaB = List.of(7,8,6,9,5,5,5,7,4,4,8,9,5,70,50,30);
        for (Integer item: listaB){
            setB.agregar(item);
        }
        System.out.println("el conjunto b es = " + setB.toString());

        System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*");

        List<Integer> listaA2 = List.of(12,11,56,12,56,78);
        for (Integer item: listaA2){
            setA.eliminar(item);
        }
    }
}
```

```

System.out.println("el conjunto a es = " + setA.toString());

List<Integer> listaB2 = List.of(20,50,30,70,50,30);
for (Integer item: listaB2){
    setB.eliminar(item);
}
System.out.println("el conjunto b es = " + setB.toString());

System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*");

System.out.print("el conjunto a tiene " + setA.longitud() + " de
largo");
System.out.print(" y el conjunto b tiene " + setB.longitud() + " de
largo \n");

System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*");

int a = 3;
int b = 7;

if (setA.contiene(a)) {
    System.out.println("el conjunto a contiene " + a);
}else{
    System.out.println("el conjunto a NO contiene " + a);
}

if (setA.contiene(b)) {
    System.out.println("el conjunto a contiene " + b);
}else {
    System.out.println("el conjunto a NO contiene " + b);
}

if (setB.contiene(a)) {
    System.out.println("el conjunto b contiene " + a);
}else{
    System.out.println("el conjunto b NO contiene " + a);
}

if (setB.contiene(b)) {
    System.out.println("el conjunto b contiene " + b);
}else{

```

```

        System.out.println("el conjunto b NO contiene " + b);
    }

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    if (setA.equals(setB)) {
        System.out.println("el conjunto a es igual al conjunto b");
    } else {
        System.out.println("el conjunto a NO es igual al conjunto b");
    }

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    if (setA.esSubConjunto(setB)) {
        System.out.println("el conjunto a es subconjunto de b");
    } else {
        System.out.println("el conjunto a NO es subconjunto de b");
    }

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    if (setB.esSubConjunto(setA)) {
        System.out.println("el conjunto b es subconjunto de a");
    } else {
        System.out.println("el conjunto b NO es subconjunto de a");
    }

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    System.out.println("la union de a y b es = " +
setA.union(setB).toString());

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    System.out.println("la interseccion de a y b es = " +
setA.interseccion(setB).toString());

    System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*-*-*");

    System.out.println("la diferencia de a y b es = " +
setA.diferencia(setB).toString());

```

```

        System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*");

        System.out.println("la diferencia de b y a es = " +
setB.diferencia(setA).toString());

        System.out.println("*-*-*-*-*-*-*-*-*-*-*-*-*");
    }
}

```

```

import java.util.Set;
import java.util.HashSet;
import java.util.List;

public class ConjuntoADT<T> {

    private Set<T> conjunto;

    public ConjuntoADT() {

        conjunto = new HashSet<>();

    }

    public int longitud() {

        return conjunto.size();

    }

    public boolean contiene(T elemento) {

        return conjunto.contains(elemento);

    }

    public void agregar(T elemento) {

        //System.out.println("agregando " + elemento);

        if(conjunto.contains(elemento) == false) {

```

```
        conjunto.add(elemento);
        //System.out.println(elemento + " ah sido agregado");

    }else{

        System.out.println("el conjunto ya contiene " + elemento);

    }
}

public void eliminar(T elemento){

    //System.out.println("eliminando " + elemento);

    if (conjunto.contains(elemento) == true){

        conjunto.remove(elemento);
        //System.out.println(elemento + " ah sido eliminado");

    }else{

        System.out.println("el conjunto no contiene " + elemento);

    }
}

public boolean equals(ConjuntoADT<T> otroConjunto){

    return conjunto.equals(otroConjunto.conjunto);

}

public boolean esSubConjunto(ConjuntoADT<T> otroConjunto) {

    return otroConjunto.conjunto.containsAll(conjunto);

}

public ConjuntoADT<T> union(ConjuntoADT<T> otroConjunto){
```

```

        ConjuntoADT union = new ConjuntoADT<>();

        union.conjunto.addAll(this.conjunto);
        union.conjunto.addAll(otroConjunto.conjunto);

        return union;
    }

    public ConjuntoADT<T> interseccion(ConjuntoADT<T> otroConjunto) {

        ConjuntoADT<T> interseccion = new ConjuntoADT<>();

        interseccion.conjunto.addAll(this.conjunto);
        interseccion.conjunto.retainAll(otroConjunto.conjunto);

        return interseccion;
    }

    public ConjuntoADT<T> diferencia(ConjuntoADT<T> otroConjunto) {

        ConjuntoADT<T> diferencia = new ConjuntoADT<>();

        diferencia.conjunto.addAll(this.conjunto);
        diferencia.conjunto.removeAll(otroConjunto.conjunto);

        return diferencia;
    }

    @Override
    public String toString(){

        return conjunto.toString();
    }
}

```

```
el conjunto ya contiene 6
el conjunto ya contiene 1
el conjunto ya contiene 5
el conjunto ya contiene 3
el conjunto ya contiene 2
el conjunto ya contiene 4
el conjunto a es = [1, 2, 3, 4, 5, 6, 56, 12, 78]
el conjunto ya contiene 5
el conjunto ya contiene 5
el conjunto ya contiene 7
el conjunto ya contiene 4
el conjunto ya contiene 8
el conjunto ya contiene 9
el conjunto ya contiene 5
el conjunto b es = [50, 4, 5, 6, 70, 7, 8, 9, 30]
*-*-*-*-*
el conjunto no contiene 11
el conjunto no contiene 12
el conjunto no contiene 56
el conjunto a es = [1, 2, 3, 4, 5, 6]
el conjunto no contiene 20
el conjunto no contiene 50
el conjunto no contiene 30
el conjunto b es = [4, 5, 6, 7, 8, 9]
*-*-*-*-*
el conjunto a tiene 6 de largo y el conjunto b tiene 6 de largo
*-*-*-*-*
```


el conjunto a tiene 6 de largo y el conjunto b tiene 6 de largo

--*-*-*

el conjunto a contiene 3

el conjunto a N0 contiene 7

el conjunto b NO contiene 3

el conjunto b contiene 7

--*-*-*

el conjunto a NO es igual al conjunto b

--*-*-*

el conjunto $a \cap b$ es subconjunto de b

--*-*-*

el conjunto $b \cap A$ es subconjunto de a

--*-*-*-*-*-*-*-*-*-*-*-*-*

la union de a y b es = [1, 2, 3, 4, 5, 6, 7, 8, 9]

--*-*-*

```
la interseccion de a y b es = [4, 5, 6]
```

--*-*-*-*-*-*-*-*-*-*-*-*-*

la diferencia de a y b es = [1, 2, 3]

--*-*-*-*-*-*-*-*-*-*-*-*-*

la diferencia de b y a es = [7, 8, 9]

--*-*-*

```
Process finished with exit code 0
```