



Hernández Rubio Aarón Isaí

Estructura de Datos

Grupo:1360(2025-1)

Tarea 2: Implementación de conjuntoADT

23 de agosto de 2024

The screenshot shows the Visual Studio Code interface with a file named `main.js` open. The code defines a `Set` class and demonstrates its usage:

```
// main.js // Language Support for JavaScript by Red Hat > main.js < mainString()>
1 public class main {
2     Run main (Debug main) / Run (Debug
3     public static void main(String[] args) {
4         conjuntoDTA<Integer>, conjuntoDB = new conjuntoADT<>();
5         conjuntoDTA<Integer>, conjuntoDB = new conjuntoADT<>();
6
7         //Agregar los elementos//
8         conjuntoDA.agregar(elemento:00);
9         conjuntoDB.agregar(elemento:00);
10        conjuntoDA.imprimirSet();
11        conjuntoDA.nuevoElemento(00);
12        conjuntoDA.imprimirSet();
13    }
14 }
```

Below the editor, the output console shows the execution results:

```
PS C:\javascript> .\C:\Program Files\Java\jdk-22\bin\java.exe --enable-preview -Xmx4G %cd%\src\mainString.js
[0] elemento ya esta dentro del set
[0] set no tiene elementos
PS C:\javascript>
```

[illegible][illegible]

[illegible]

```
PS C:\javascrpt > javac conjuntoAOT.java && java conjuntoAOT
SetADT > ./main.java > Language Support for Java(TM) by Red Hat > %> main > @ mainClass()
1 public class main {
2     public static void main(String[] args) {
3         //
4         System.out.println("Los conjuntos son iguales");
5         //
6         //
7         System.out.println("Los conjuntos no son iguales");
8         //
9         //
10        //
11        //
12        //
13        //
14        //
15        //
16        //
17        //
18        //
19        //
20        //
21        //
22        //
23        //
24        //
25        //
26        //
27        //
28        //
29        //
30        //
31        //
32        //
33        //
34        //
35        //
36        //
37        //
38        //
39        //
40        //
41        //
42        //
43        //
44        //
45        //
46        //
47        //
48        //
49        //
50        //
51        //
52        //
53        //
54        //
55        //
56        //
57        //
58        //
59        //
60        //
61        //
62        //
63        //
64        //
65        //
66        //
67        //
68        //
69        //
70        //
71        //
72        //
73        //
74        //
75        //
76        //
77        //
78        //
79        //
80        //
81        //
82        //
83        //
84        //
85        //
86        //
87        //
88        //
89        //
90        //
91        //
92        //
93        //
94        //
95        //
96        //
97        //
98        //
99        //
100       //
101       //
102       //
103       //
104       //
105       //
106       //
107       //
108       //
109       //
110       //
111       //
112       //
113       //
114       //
115       //
116       //
117       //
118       //
119       //
120       //
121       //
122       //
123       //
124       //
125       //
126       //
127       //
128       //
129       //
130       //
131       //
132       //
133       //
134       //
135       //
136       //
137       //
138       //
139       //
140       //
141       //
142       //
143       //
144       //
145       //
146       //
147       //
148       //
149       //
150       //
151       //
152       //
153       //
154       //
155       //
156       //
157       //
158       //
159       //
160       //
161       //
162       //
163       //
164       //
165       //
166       //
167       //
168       //
169       //
170       //
171       //
172       //
173       //
174       //
175       //
176       //
177       //
178       //
179       //
180       //
181       //
182       //
183       //
184       //
185       //
186       //
187       //
188       //
189       //
190       //
191       //
192       //
193       //
194       //
195       //
196       //
197       //
198       //
199       //
200       //
201       //
202       //
203       //
204       //
205       //
206       //
207       //
208       //
209       //
210       //
211       //
212       //
213       //
214       //
215       //
216       //
217       //
218       //
219       //
220       //
221       //
222       //
223       //
224       //
225       //
226       //
227       //
228       //
229       //
230       //
231       //
232       //
233       //
234       //
235       //
236       //
237       //
238       //
239       //
240       //
241       //
242       //
243       //
244       //
245       //
246       //
247       //
248       //
249       //
250       //
251       //
252       //
253       //
254       //
255       //
256       //
257       //
258       //
259       //
260       //
261       //
262       //
263       //
264       //
265       //
266       //
267       //
268       //
269       //
270       //
271       //
272       //
273       //
274       //
275       //
276       //
277       //
278       //
279       //
280       //
281       //
282       //
283       //
284       //
285       //
286       //
287       //
288       //
289       //
290       //
291       //
292       //
293       //
294       //
295       //
296       //
297       //
298       //
299       //
300       //
301       //
302       //
303       //
304       //
305       //
306       //
307       //
308       //
309       //
310       //
311       //
312       //
313       //
314       //
315       //
316       //
317       //
318       //
319       //
320       //
321       //
322       //
323       //
324       //
325       //
326       //
327       //
328       //
329       //
330       //
331       //
332       //
333       //
334       //
335       //
336       //
337       //
338       //
339       //
340       //
341       //
342       //
343       //
344       //
345       //
346       //
347       //
348       //
349       //
350       //
351       //
352       //
353       //
354       //
355       //
356       //
357       //
358       //
359       //
360       //
361       //
362       //
363       //
364       //
365       //
366       //
367       //
368       //
369       //
370       //
371       //
372       //
373       //
374       //
375       //
376       //
377       //
378       //
379       //
380       //
381       //
382       //
383       //
384       //
385       //
386       //
387       //
388       //
389       //
390       //
391       //
392       //
393       //
394       //
395       //
396       //
397       //
398       //
399       //
400       //
401       //
402       //
403       //
404       //
405       //
406       //
407       //
408       //
409       //
410       //
411       //
412       //
413       //
414       //
415       //
416       //
417       //
418       //
419       //
420       //
421       //
422       //
423       //
424       //
425       //
426       //
427       //
428       //
429       //
430       //
431       //
432       //
433       //
434       //
435       //
436       //
437       //
438       //
439       //
440       //
441       //
442       //
443       //
444       //
445       //
446       //
447       //
448       //
449       //
450       //
451       //
452       //
453       //
454       //
455       //
456       //
457       //
458       //
459       //
460       //
461       //
462       //
463       //
464       //
465       //
466       //
467       //
468       //
469       //
470       //
471       //
472       //
473       //
474       //
475       //
476       //
477       //
478       //
479       //
480       //
481       //
482       //
483       //
484       //
485       //
486       //
487       //
488       //
489       //
490       //
491       //
492       //
493       //
494       //
495       //
496       //
497       //
498       //
499       //
500       //
501       //
502       //
503       //
504       //
505       //
506       //
507       //
508       //
509       //
510       //
511       //
512       //
513       //
514       //
515       //
516       //
517       //
518       //
519       //
520       //
521       //
522       //
523       //
524       //
525       //
526       //
527       //
528       //
529       //
530       //
531       //
532       //
533       //
534       //
535       //
536       //
537       //
538       //
539       //
540       //
541       //
542       //
543       //
544       //
545       //
546       //
547       //
548       //
549       //
550       //
551       //
552       //
553       //
554       //
555       //
556       //
557       //
558       //
559       //
560       //
561       //
562       //
563       //
564       //
565       //
566       //
567       //
568       //
569       //
570       //
571       //
572       //
573       //
574       //
575       //
576       //
577       //
578       //
579       //
580       //
581       //
582       //
583       //
584       //
585       //
586       //
587       //
588       //
589       //
590       //
591       //
592       //
593       //
594       //
595       //
596       //
597       //
598       //
599       //
600       //
601       //
602       //
603       //
604       //
605       //
606       //
607       //
608       //
609       //
610       //
611       //
612       //
613       //
614       //
615       //
616       //
617       //
618       //
619       //
620       //
621       //
622       //
623       //
624       //
625       //
626       //
627       //
628       //
629       //
630       //
631       //
632       //
633       //
634       //
635       //
636       //
637       //
638       //
639       //
640       //
641       //
642       //
643       //
644       //
645       //
646       //
647       //
648       //
649       //
650       //
651       //
652       //
653       //
654       //
655       //
656       //
657       //
658       //
659       //
660       //
661       //
662       //
663       //
664       //
665       //
666       //
667       //
668       //
669       //
670       //
671       //
672       //
673       //
674       //
675       //
676       //
677       //
678       //
679       //
680       //
68
```

CODIGO:

```
import java.util.HashSet;

public class conjuntoADT<T>{

    private HashSet<T> set = new HashSet<>();

    public conjuntoADT(){

    }

    public conjuntoADT(HashSet<T> setPrev){
        this.set=setPrev;
    }
}
```

```

}

public int longitud(){
    return this.set.size();
}

public boolean contiene(T elemento){
    if(this.set.contains(elemento)){
        return true;
    }else{
        return false;
    }
}

public void agregar(T elemento){
    if(this.contiene(elemento)==true){
        System.out.println("El elemento ya esta dentro del set");
    }else{
        set.add(elemento);
    }
}

public void eliminar(T elemento){
    if(this.contiene(elemento)==true){
        this.set.remove(elemento);
    }else{
        System.out.println("El elemento no esta dentro del set");
    }
}

public boolean equals(HashSet<T> otroConjunto){
    if(this.set.equals(otroConjunto)){
        return true;
    }else{
        return false;
    }
}

public boolean esSubconjunto(HashSet<T> otroConjunto){
    if(otroConjunto.containsAll(this.set)){
        return true;
    }else{
        return false;
    }
}

```

```

    public HashSet<T> union(HashSet<T> otroConjunto){
        HashSet<T> setAux = new HashSet<>();
        setAux.addAll(this.set);
        setAux.addAll(otroConjunto);
        return setAux;
    }

    public HashSet<T> interseccion(HashSet<T> otroConjunto){
        HashSet<T> setAux= new HashSet<>(this.set);
        setAux.retainAll(otroConjunto);
        return setAux;
    }

    public HashSet<T> diferencia(HashSet<T> otroConjunto){
        HashSet<T> setAux= new HashSet<>(this.set);
        setAux.removeAll(otroConjunto);
        return setAux;
    }

    public HashSet<T> getSet(){
        return this.set;
    }

    public void imprimirSet(){

        if(this.longitud()==0){
            System.out.println("El set no tiene elementos");
        }

        for(T elemento : this.set) {
            System.out.print(elemento + ",");
        }
        System.out.println();
    }
}

```

```

import java.util.HashSet;

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

        conjuntoADT<Integer> conjuntoA= new conjuntoADT<>();
        conjuntoADT<Integer> conjuntoB= new conjuntoADT<>();
    }
}

```

```
//agregando elementos//
/*
conjuntoA.agregar(88);
conjuntoA.agregar(88);
conjuntoA.imprimirSet();
conjuntoA.eliminar(88);
conjuntoA.imprimirSet();
*/

conjuntoA.agregar(88);
conjuntoA.agregar(14);
conjuntoA.agregar(4);
conjuntoA.agregar(1);
conjuntoA.agregar(5);
conjuntoA.agregar(24);
conjuntoA.agregar(38);
conjuntoA.agregar(47);

conjuntoB.agregar(5);
conjuntoB.agregar(4);
conjuntoB.agregar(38);
conjuntoB.agregar(47);
conjuntoB.agregar(23);
conjuntoB.agregar(2);
conjuntoB.agregar(14);
conjuntoB.agregar(3);
conjuntoB.agregar(89);

/*
if(conjuntoA.contiene(1)){
    System.out.println("El conjuntoA si contiene ese numero");
}else{
    System.out.println("El conjuntoA no contiene ese numero");
}

if(conjuntoB.contiene(1)){
    System.out.println("El conjuntoB si contiene ese numero");
}else{
    System.out.println("El conjuntoB no contiene ese numero");
}
*/

HashSet<Integer> conjuntoUnion=conjuntoA.union(conjuntoB.getSet());
```

```

        conjuntoADT<Integer> conjuntoUnionADT= new
conjuntoADT<>(conjuntoUnion);
        System.out.println("Operación Unión:");
        conjuntoUnionADT.imprimirSet();

        HashSet<Integer>
conjuntoInterseccion=conjuntoA.interseccion(conjuntoB.getSet());
        conjuntoADT<Integer> conjuntoInterseccionADT= new
conjuntoADT<>(conjuntoInterseccion);
        System.out.println("Operación Intersección:");
        conjuntoInterseccionADT.imprimirSet();

        HashSet<Integer>
conjuntoDiferencia=conjuntoA.diferencia(conjuntoB.getSet());
        conjuntoADT<Integer> conjuntoDiferenciaADT= new
conjuntoADT<>(conjuntoDiferencia);
        System.out.println("Operación Diferencia:");
        conjuntoDiferenciaADT.imprimirSet();

        /*
        if(conjuntoA.equals(conjuntoB.getSet())){
            System.out.println("Los conjuntos son iguales");
        }else{
            System.out.println("Los conjuntos no son iguales");
        }

        if(conjuntoA.esSubconjunto(conjuntoB.getSet())){
            System.out.println("El conjunto A es subconjunto del conjunto
B");
        }else{
            System.out.println("El conjunto A no es subconjunto del conjunto
B");
        }
        */
    }
}

```