

1151044 - Programación Orientada a Objetos
Tarea 6 - 2026-I

Implementa un tipo de dato `Conjunto` que modele un conjunto que puede contener los enteros del 0 al 99.

```
class Conjunto {  
    //...  
};
```

Las operaciones públicas disponibles deben ser las siguientes.

- Inicialización por defecto.
El conjunto comienza vacío.

```
var c = new Conjunto( );
```

- Función miembro estática `static int minimo()`;
Devuelve el valor mínimo que el tipo `Conjunto` permite para inserción (en este caso, el valor 0).

```
int min = Conjunto.minimo( );
```

- Función miembro estática `static int maximo()`;
Devuelve el valor máximo que el tipo `Conjunto` permite para inserción (en este caso, el valor 99).

```
int max = Conjunto.maximo( );
```

- Función miembro `boolean inserta(int v)`;¹
Inserta el valor `v` al conjunto. Devuelve verdadero si el elemento se insertó exitosamente. Una inserción falla si el valor ya existía en el conjunto o si el valor que se solicitó insertar está fuera de rango.

```
boolean b = c.inserta(3);
```

- Función miembro `boolean elimina(int v)`;
Elimina el valor `v` del conjunto. Devuelve verdadero si el elemento se eliminó exitosamente. Una eliminación falla si el valor que se solicitó eliminar no existía en el conjunto.

```
boolean b = c.elimina(4);
```

- Función miembro `int size()`;
Devuelve la cantidad de elementos del conjunto.

```
int tam = c.size( );
```

- Función miembro `int get(int i)`;
Devuelve una copia del valor del i -ésimo elemento del conjunto. Los elementos del conjunto se consideran implícitamente ordenados de menor a mayor y reciben índices a partir de cero. Es decir, el elemento más pequeño del conjunto es `c.get(0)`, el siguiente elemento más pequeño es `c.get(1)`, etc. Puedes suponer que el índice es válido.

```
int v = c.get(0);
```

- Función miembro `boolean equals(Conjunto otro)`;
Determina si el conjunto actual tiene el mismo valor que el otro conjunto. Dos conjuntos tienen el mismo valor si y sólo si tienen el mismo tamaño y también contienen los mismos elementos.

```
var c = new Conjunto( );  
var d = new Conjunto( );  
//...  
boolean b = c.equals(d);
```

¹Java usa `%b` en `printf` para imprimir un `boolean`, además de que no admite conversiones (ni implícitas ni explícitas) de `boolean` a `int`.

El tipo de dato debe implementarse de modo de que sea imposible que un conjunto adquiriera un estado inválido. Un programa que manipule a lo mucho cien conjuntos y que haga a lo mucho mil llamadas a función debe terminar en menos de un segundo. Puedes consultar una página de prueba en <https://racc.mx/uam/trimestre-actual/2026-i/poo/tarea6.html>. El código que envíes no debe contener `main` y no debe leer ni imprimir nada directamente.

Envía tu código fuente desde tu cuenta institucional al formulario en <https://forms.gle/p1FLFEmrwfnRD9cQA>. Tu código será evaluado con varios casos de prueba y se espera que cumpla la semántica descrita.

Ejemplo de uso	Ejemplo de salida
<pre> void main() { var c = new Conjunto(); boolean b1 = c.inserta(3); boolean b2 = c.inserta(1); boolean b3 = c.inserta(3); System.out.printf("%b %b %b\n", b1, b2, b3); System.out.printf("%d\n", c.size()); for (int i = 0; i < c.size(); ++i) { System.out.printf("%d ", c.get(i)); } System.out.println(); var d = new Conjunto(); d.inserta(1); d.inserta(3); System.out.print(c.equals(d)); } </pre>	<pre> true true false 2 1 3 true </pre>