

A thick dark blue vertical bar is positioned on the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the date. Below the banner, several thin, curved lines in dark blue and light grey sweep upwards from the bottom left corner.

23-2-2019

PRUEBA DEL PROGRAMA PERSONA-TELEVISIÓN

PROGRAMACIÓN ORIENTADA A OBJETOS

INTEGRANTES:

JOSÉ ÁNGEL GARCÍA GARCÍA
MELVIN PAUL GONZALEZ PASCUAL
LAURA YESSENIA SANCHEZ LOPEZ
SERGIO ROMERO CORTEZ

UNA PERSONA PUEDE PRENDER Y APAGAR, BAJAR Y SUBIR VOLUMEN, BAJAR Y SUBIR CANAL, DE UN TELEVISOR

The screenshot shows the BlueJ IDE interface. On the left, the 'Proyecto' tab displays a class hierarchy with 'Televisión' and 'Persona' classes. Below this, the 'Teamwork' section shows two objects: 'persona1: Persona' and 'tv: Televisión'. On the right, the 'Inspeccionar' (Inspect) window is open for the 'tv: Televisión' object. It displays the following attributes and values:

Attribute	Value
private String nombre	"Manolo"
private boolean estado	false
private int vol	0
private int volmax	5
private int ncan	1
private int canalmax	7

Buttons for 'Inspeccionar', 'Obtener', 'Mostrar campos estáticos', and 'Cerrar' are visible for each object's inspection window.

Lo primero es crear los objetos de cada clase, empezando por la clase persona. La persona se llamará “Manolo”, la televisión tendrá atributos como: el numero de volumen actual en 0, el volumen máximo que puede tener será 5, el numero de canal actual es 1 y el canal máximo es 7, además la televisión se encuentra “false”; en este programa vamos a tomar a “true” como encendida y a “false” como apagada.

The screenshot shows the 'heredado de Object' (Inherited from Object) menu in the BlueJ IDE. The method list includes:

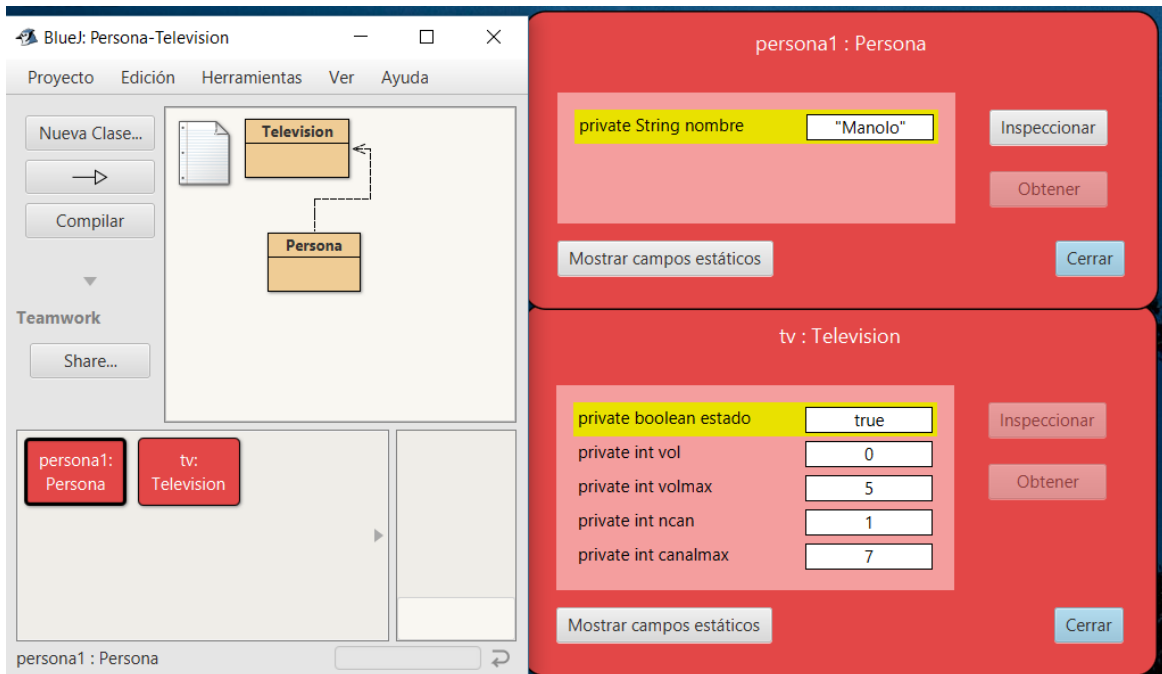
- void apagarencender(Televisión t)
- void bajacanal(Televisión t)
- void bajarvol(Televisión t)
- String getnombre()
- void setnombre(String nombre)
- void subircanal(Televisión t)
- void subirvol(Televisión t)

Buttons for 'Inspeccionar' and 'Remover' are also visible.

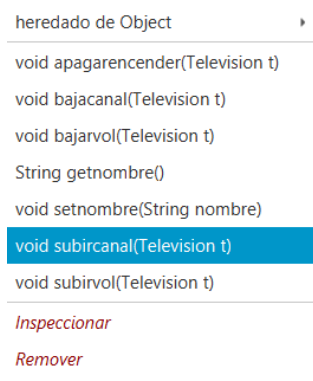
El primer método por ejecutar es el de encender, puesto que, si no está encendida la televisión, no podemos ejecutar los demás métodos.

Este método requiere como parámetros la televisión, ya que es a quien le va a modificar el estado. Le colocamos como argumento tv, ya que ese nombre recibe el objeto de nuestra clase televisión.

The screenshot shows the 'Llamado a Método' (Method Call) dialog in the BlueJ IDE. The method being called is 'void apagarencender(Televisión t)'. The argument 'tv' is entered in the input field. Buttons for 'Aceptar' (Accept) and 'Cancelar' (Cancel) are visible.

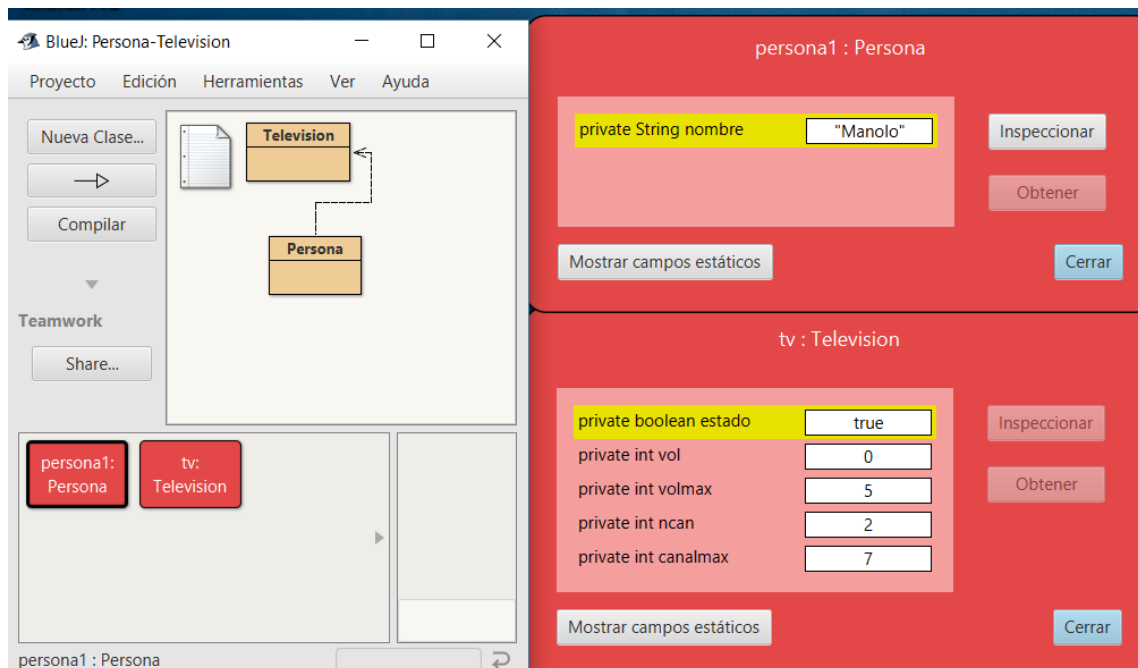
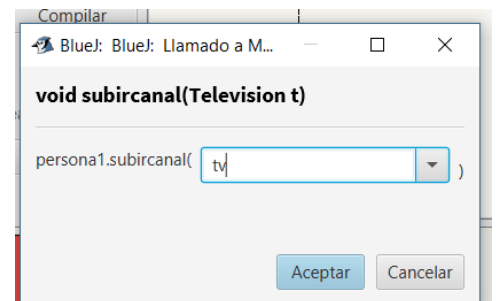


Como podemos observar, el método de encender se ejecuto de manera correcta, ahora tenemos la televisión encendida, el estado optimo para proceder con los demás métodos.



El método subir canal, requiere como único parámetro a la televisión, ya que en ella hará la modificación del canal actual en el que se encuentre, en este método aumentará el canal de uno en uno.

Observando la imagen podemos ver que el método se ejecutó de la forma correcta. Aumentando el canal de 1 a 2.



heredado de Object

void apagarencender(Television t)

void bajacanal(Television t)

void bajarvol(Television t)

String getnombre()

void setnombre(String nombre)

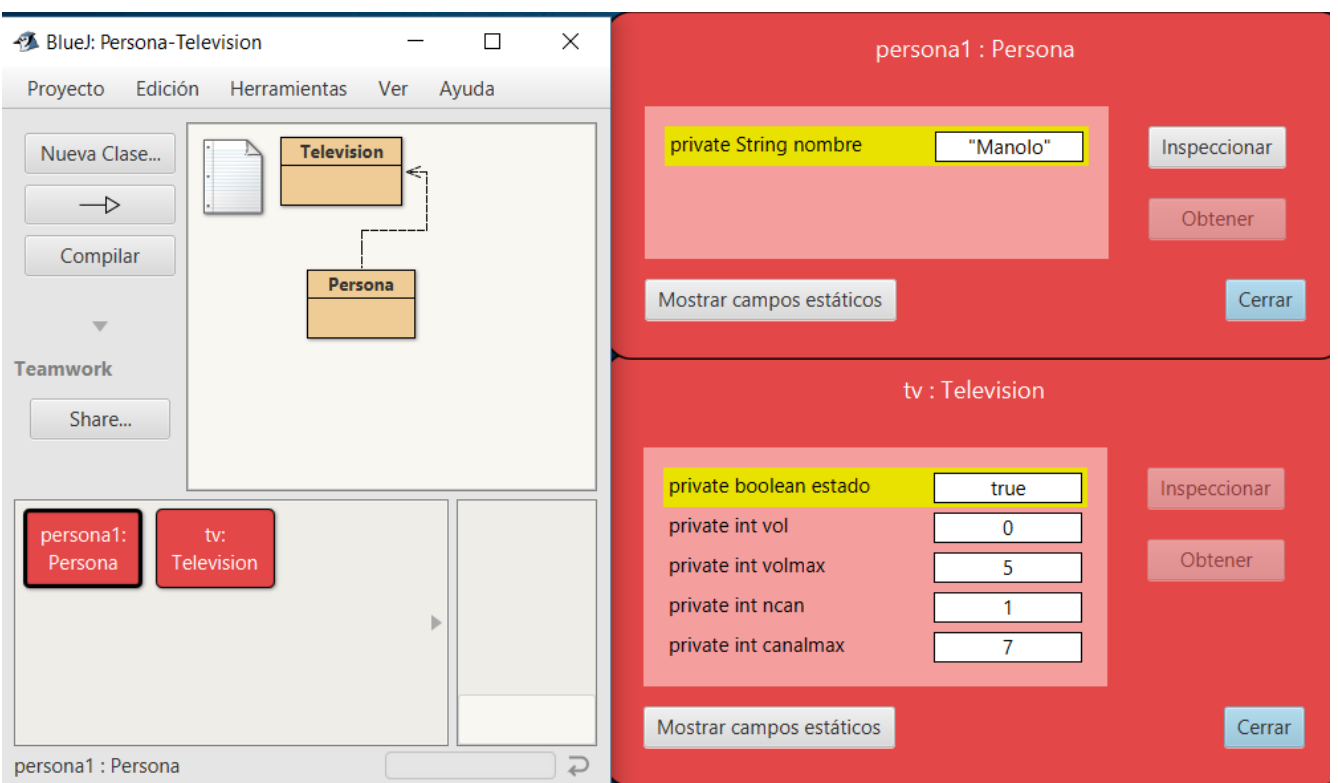
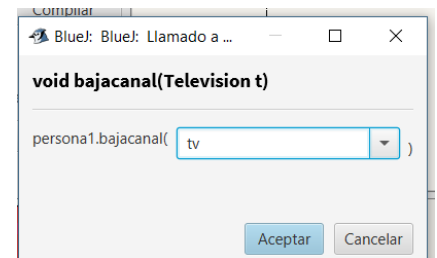
void subircanal(Television t)

void subirvol(Television t)

Inspeccionar

Remover

El próximo método a ejecutara es el de bajar canal, que de igual forma que el anterior, requiere solo de un parámetro, que es a la televisión. Por lo que le damos como argumento tv, que es el objeto al que le cambiara el atributo numero de canal, haciendo que disminuya en uno.



Como podemos ver en la imagen, el numero del canal cambio de 2 a 1, es decir, disminuyo en uno, por lo que podemos concluir que el método bajar canal se ejecutó de la manera correcta.

heredado de Object

void apagarencender(Television t)

void bajacanal(Television t)

void bajarvol(Television t)

String getnombre()

void setnombre(String nombre)

void subircanal(Television t)

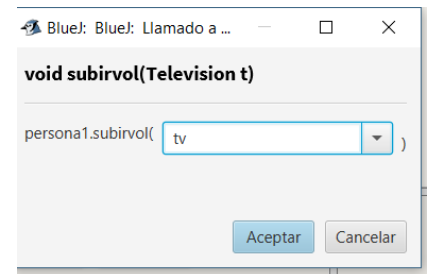
void subirvol(Television t)

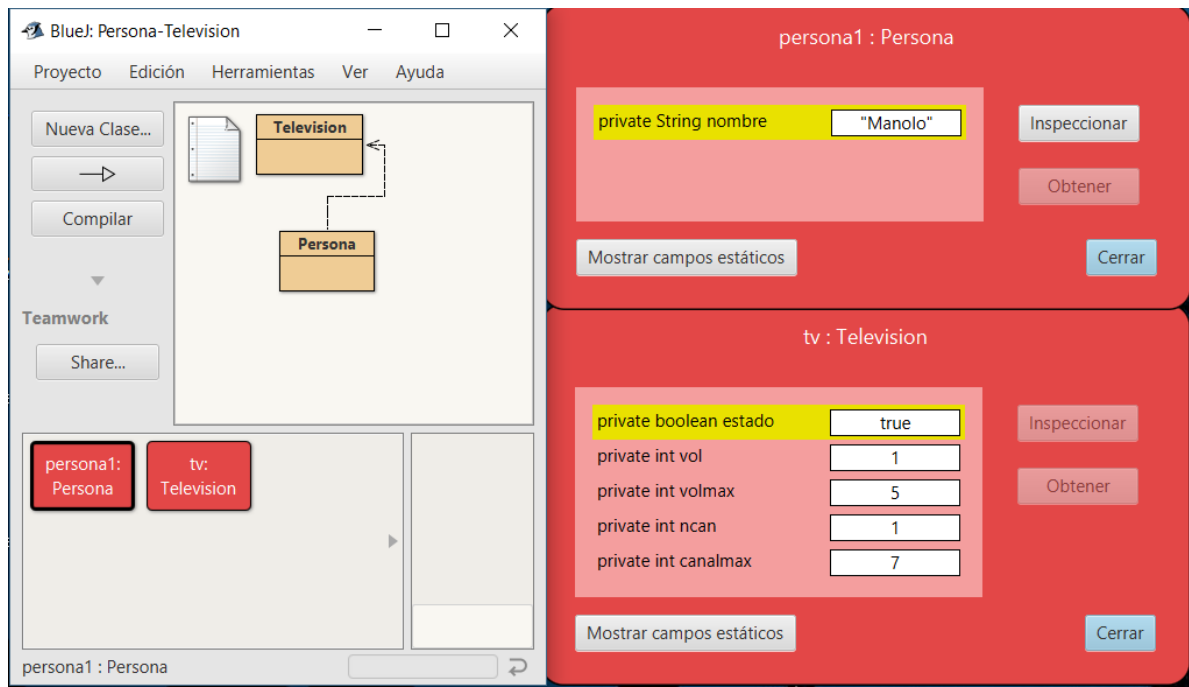
Inspeccionar

Remover

Procedemos a ejecutar el siguiente método que es el de subir volumen, que de igual forma que los métodos anteriores necesitan un parámetro, que es la televisión, pero en este caso va a cambiar el número de volumen haciendo que aumente en uno.

Le damos como argumento a la tv, para que se pueda ejecutar y cambiar en el objeto dicho atributo, haciendo que este aumente.





En la imagen se puede ver claramente que el método se ejecuto de la forma correcta puesto que el numero de volumen actual cambio de 0 a 1; es decir, se subió el volumen de la televisión.

heredado de Object

- void apagarencender(Televisión t)
- void bajacanal(Televisión t)
- void bajarvol(Televisión t)**
- String getnombre()
- void setnombre(String nombre)
- void subircanal(Televisión t)
- void subirvol(Televisión t)

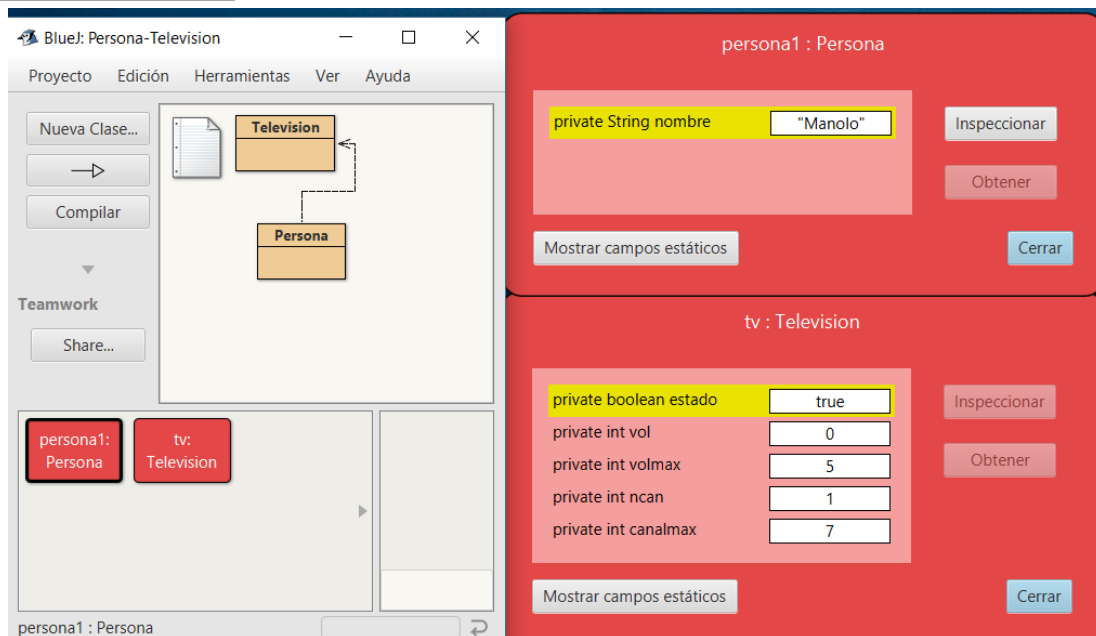
Inspeccionar

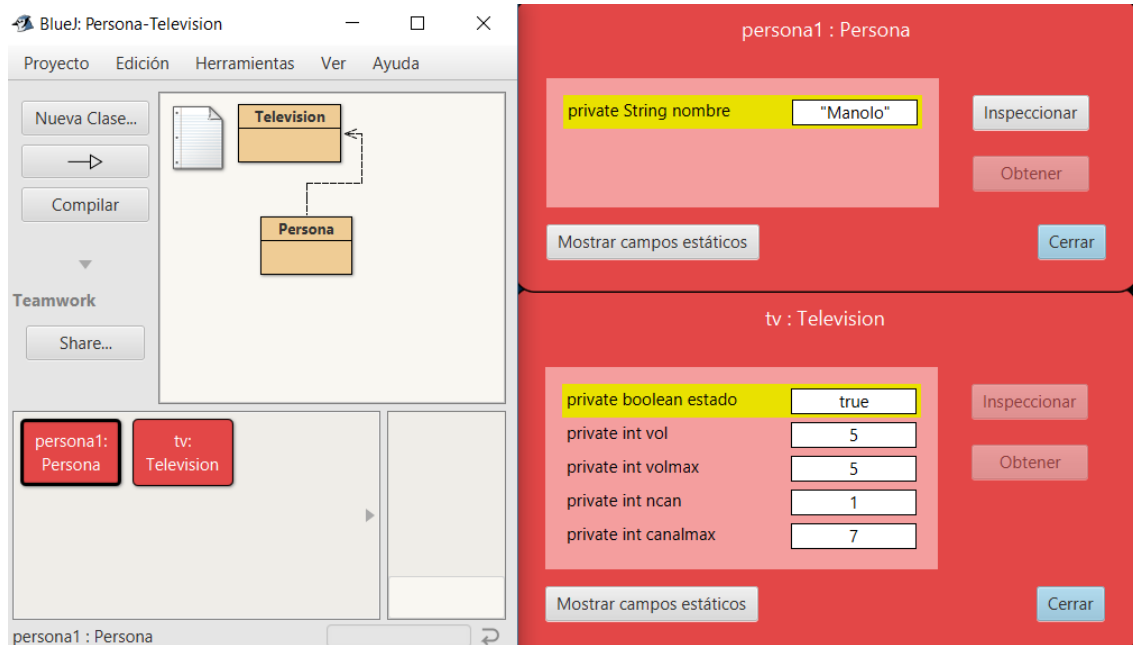
Remover

El otro método que se tiene es el de bajar volumen y funciona de forma inversa al de subir, es decir, lo único diferente es que disminuirá el volumen en uno.

Partiendo que el volumen lo tenemos en 1, procederemos a ejecutar el método. El volumen cambiara de 1 a 0, como se ve claramente en la imagen de abajo.

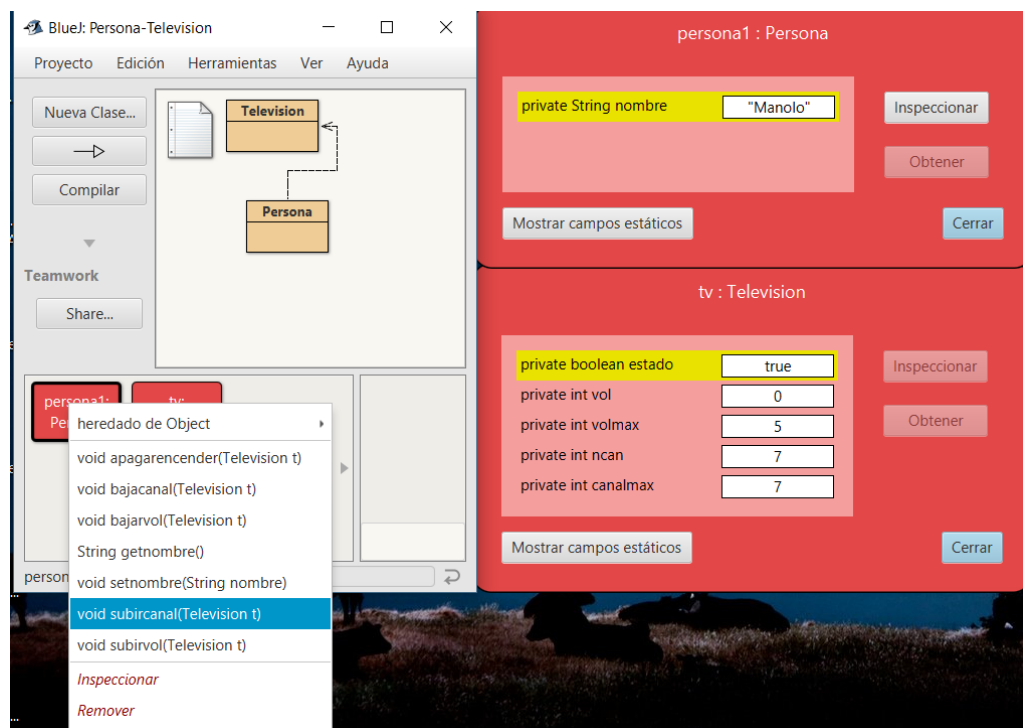
The screenshot shows a 'Llamado a ...' (Called ...) dialog box for the 'void bajarvol(Televisión t)' method. The 'persona1.bajarvol(' method call is shown with 'tv' selected in the dropdown menu. The 'Aceptar' (Accept) button is highlighted.

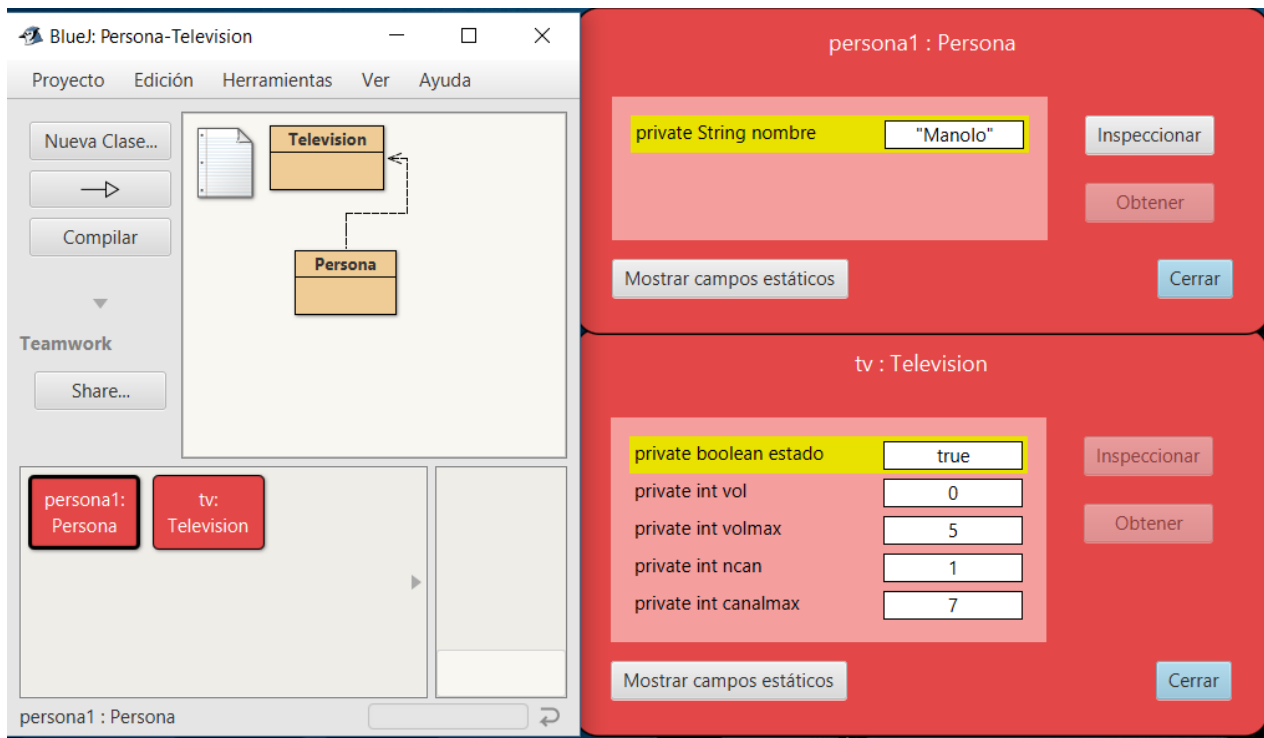




Al ejecutar el método de subir volumen varias veces y llegar hasta igual el número máximo, lo que el programa hace es dejar el volumen igual al máximo, es decir, ya no se puede tener un volumen mayor y no regresa a uno, ni disminuye, simplemente se queda como máximo. Esto se observa en la imagen de arriba.

Ahora al ejecutar de igual forma varias veces el método subir canal y llegar al máximo de número de canal, así como se muestra en la imagen, se tiene un caso especial, puesto que el número de canal actual no actúa de igual forma que el volumen, este a diferencia de el número de volumen, regresar a 1, para volver a empezar a subir canal por canal, es decir, hace un ciclo.





Ahora observamos que el número del canal volvió a uno, puesto que llegó a rebasar al número máximo de canal, así volviendo a empezar y permitiendo que dicho método se ejecute de n formas.

heredado de Object
void apagarencender(Television t)
 void bajacanal(Television t)
 void bajarvol(Television t)
 String getnombre()
 void setnombre(String nombre)
 void subircanal(Television t)
 void subirvol(Television t)
 Inspeccionar
 Remover

Por último, el método apagar, que se ejecuta solo cuando la televisión se encuentra encendida. Tiene un solo parámetro, que es la televisión. Para que se ejecute el método, le introducimos como argumento tv, que es el objeto al que modificara su estado.

En la imagen de abajo se puede observar como el método funciona correctamente, haciendo que el estado de la televisión cambie de "true" a "false"; es decir, la televisión se apague.

