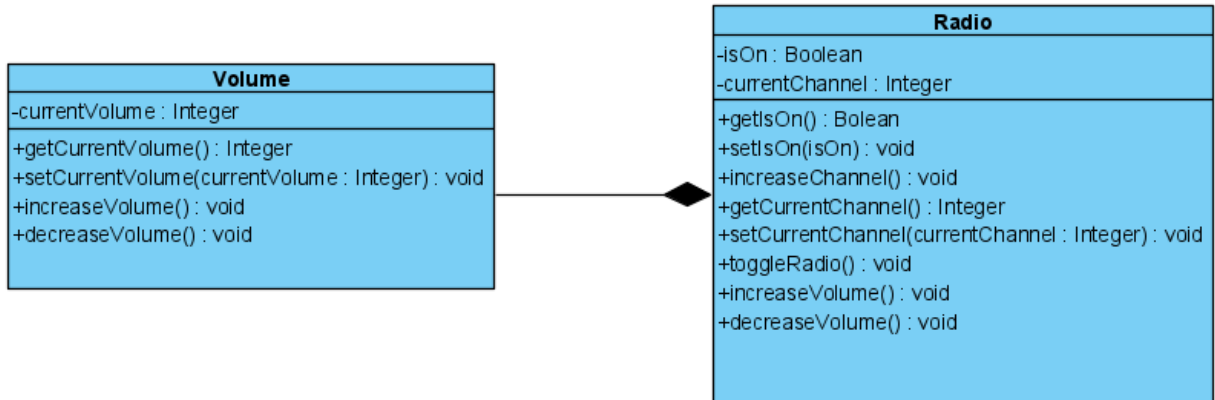


Task 1



OCL constraints:

context Radio **inv:** $1 \leq \text{self.currentChannel} \leq 5$

context Radio:currentChannel: Integer **init:** 1

context Volume **inv:** $0 \leq \text{self.currentVolume} \leq 100$

context Radio::increaseVolume() **pre:** on = True

context Radio::decreaseVolume() **pre:** on = True

context Radio::increaseChannel() **pre:** on = True

Source code:
JAVA

```
public class Radio {
    private boolean isOn;
    private int currentChannel;
    private Volume volume;

    public Radio(){
        isOn= false;
        currentChannel = 1;
        volume = new Volume();
    }

    public boolean getIsOn(){
        return isOn;
    }

    public void setIsOn(boolean v){
        isOn = v;
    }

    public void toggleRadio(){
        setIsOn(!isOn);
    }

    public int getCurrentChannel() {
        return currentChannel;
    }

    public void setCurrentChannel(int channel){
        if(isOn){

            if(channel>=1&&channel<=5){
                currentChannel = channel;
            }else{
                System.out.println("WRONG CHANNEL");
            }
        }
    }
}
```

```
} public void increaseChannel() {  
    if (isOn) {  
        if (currentChannel == 5) {  
            setCurrentChannel(1);  
        } else {  
            setCurrentChannel(currentChannel + 1);  
        }  
    }  
}  
  
public void increaseVolume(){  
    if(isOn) {  
        volume.increaseVolume();  
    }  
}  
  
public void decreaseVolume(){  
    if(isOn) {  
        volume.decreaseVolume();  
    }  
}  
}
```

```
public class Volume {  
    private int currentVolume;  
  
    public Volume(){  
        currentVolume = 0;  
    }  
    public int getCurrentVolume(){  
        return currentVolume;  
    }  
    public void setCurrentVolume(int vol){  
        if(vol>=0&&vol<=100){  
            currentVolume = vol;  
        }  
        else{  
            System.out.println("WRONG VOLUME");  
        }  
    }  
    public void increaseVolume(){  
        setCurrentVolume(currentVolume+1);  
    }  
    public void decreaseVolume(){  
        setCurrentVolume(currentVolume-1);  
    }  
}
```

Assertions:

```
import org.junit.Assert;
import org.junit.Test;
public class tests {
    @Test
    public void testVolume() {
        Radio radio = new Radio();
        for(int i = 0; i < 200; i++){
            radio.decreaseVolume();
        }
        Assert.assertEquals(radio.getVolume(), actual: 0);
        for(int i = 0; i < 200; i++){
            radio.increaseVolume();
        }
        Assert.assertEquals(radio.getVolume(), actual: 0);
        radio.toggleRadio();
        for(int i = 0; i < 200; i++){
            radio.decreaseVolume();
        }
        Assert.assertEquals(radio.getVolume(), actual: 0);
        for(int i = 0; i < 200; i++){
            radio.increaseVolume();
        }
        Assert.assertEquals(radio.getVolume(), actual: 100);
    }
    @Test
    public void testChannel() {
        Radio radio = new Radio();
        //test when off
        radio.decreaseVolume();
        Assert.assertEquals(radio.getCurrentChannel(), actual: 1);
        radio.increaseChannel();
        Assert.assertEquals(radio.getCurrentChannel(), actual: 1);

        radio.toggleRadio();
        radio.increaseChannel();
        radio.increaseChannel();
        radio.increaseChannel();
        radio.increaseChannel();
        Assert.assertEquals(radio.getCurrentChannel(), actual: 5);
        radio.increaseChannel();
        Assert.assertEquals(radio.getCurrentChannel(), actual: 1);
    }
}
```

Output:

| | | |
|---|-------------|-------|
| ✓ | tests | 90 ms |
| ✓ | testChannel | 4 ms |
| ✓ | testVolume | 86 ms |