

Bereich: Dokumentation**Radio kommentiert****Musterlösung****Package:** de.dhbwka.java.exercise.docu**Klasse:** Radio

```
package de.dhbwka.java.exercise.docu;
```

```
/**
 * @author DHBW lecturer
 * @version 1.0
 *
 * Part of lectures on 'Programming in Java'.
 * Baden-Wuerttemberg Cooperative State University.
 *
 * (C) 2015 by J. Sidler, T. Schlachter, C. Schmitt, W. Süß
 *
 * Die Klasse Radio implementiert das Verhalten eines Radios. Hierfür stellt sie
 * verschiedene Konstruktoren und Methoden inkl einer main()-Methode zum Testen
 * bereit.
 */
public class Radio {

    /**
     * Zustand des Radios. Eingeschaltet oder nicht eingeschaltet.
     */
    public boolean eingeschaltet = false;

    /**
     * Wie laut spielt das Radio Musik. Die Lautstärke soll nur im Bereich von 0
     * bis 10 liegen.
     */
    public int lautstaerke = 5;

    /**
     * Frequenz des gewählten Senders. Erlaubter Frequenzbereich ist zwischen
     * 85.0 und 110.0.
     */
    public double frequenz = 98.4;

    /**
     * Erzeugt ein Standard-Radio-Objekt. Standardwerte: eingeschaltet = false,
     * lautstaerke = 5, frequenz = 98.4
     */
    public Radio() {
    }

    /**
     * Erzeugt ein Radio-Objekt mit den angegebenen Attributwerten.
     *
     * @param einaus Zustand des Radios: eingeschaltet (true) oder ausgeschaltet
     * (false)
     * @param laut Lautstärke
     * @param freq Frequenz
     */
    public Radio(boolean einaus, int laut, double freq) {
```

```
        eingeschaltet = einaus;
        if (laut >= 0 && laut <= 10) {
            lautstaerke = laut;
        }
        waehleSender(freq);
    }

    /**
     * Reduziert die Lautstärke um eine Einheit. Ist nur möglich im Zustand an.
     */
    public void leiser() {
        if (eingeschaltet && lautstaerke > 0) {
            lautstaerke--;
        }
    }

    /**
     * Erhöht die Lautstärke um eine Einheit. Ist nur möglich im Zustand an.
     */
    public void lauter() {
        if (eingeschaltet && lautstaerke < 10) {
            lautstaerke++;
        }
    }

    /**
     * Setzt den Zustand des Radios auf eingeschaltet. Das Attribut
     * eingeschaltet wird auf true gesetzt.
     */
    public void an() {
        eingeschaltet = true;
    }

    /**
     * Setzt den Zustand des Radios auf ausgeschaltet. Das Attribut
     * eingeschaltet wird auf false gesetzt.
     */
    public void aus() {
        eingeschaltet = false;
    }

    /**
     * Gibt Informationen über den internen Zustand als String zurück. Es wird
     * eine Zeichenkette der Form "Radio an: Freq=98.4, Laut=2" zurückgeben.
     */
    @Override
    public String toString() {
        String s = "Radio ";
        if (eingeschaltet) {
            s += "an:";
            s += " Freq=" + frequenz + ", ";
            s += "Laut=" + lautstaerke + "";
        } else {
            s += "aus.";
        }
        return s;
    }
}
```

```
/**
 * Speichert eine Frequenz. Ist die gewählte Frequenz ausserhalb der
 * erlaubten Frequenzbereichs, so wird die Frequenz 99.9 gewählt.
 */
public void waehleSender(double frequenz) {
    if ((frequenz <= 110.0) && (frequenz >= 85.0)) {
        this.frequenz = frequenz;
    } else {
        this.frequenz = 99.9;
    }
}

/**
 * Dient zum Testen der Methoden der Klasse Radio.
 */
public static void main(String[] args) {
    Radio radio = new Radio(true, 8, 99.9);
    radio.leiser();
    System.out.println(radio.toString());
    radio.leiser();
    System.out.println(radio.toString());
    radio.lauter();
    System.out.println(radio.toString());
    radio.waehleSender(112.8);
    System.out.println(radio.toString());
    radio.waehleSender(93.5);
    System.out.println(radio.toString());
    radio.aus();
    System.out.println(radio.toString());
    radio.leiser();
    System.out.println(radio.toString());
    radio.an();
    System.out.println(radio.toString());
}
}
```