

```

import java.util.Vector;

/**
 * Classe di Test.
 *
 * @author Filippo Fontanelli , Francesca Brogi
 *
 */
public class testClassRegLezioni {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            System.out
                .println("\n=====Provo la classe
'RegLezioni'...\n=====");
            RegLezioni[] dip = new RegLezioni[5];
            System.out.println(".");
            // PROVO COSTRUTTORI
            try {
                // Caso Null
                dip[0] = new RegLezioni(null);
                System.out
                    .println("Attento: il costruttore deve lanciare una
'NullPointerException' \nse la lista di lezioni e' uguale a null: correggi e riprova
.");
            } catch (NullPointerException e) {
            }

            System.out.println("..");

            // PROVO SIZE
            dip[0] = new RegLezioni(new Vector<Lezione>());

            if (dip[0].size() != 0)
                System.out
                    .println("Attento: il valore restituito da size()()\n non
corrisponde al numero degli elementi");

            // PROVO GETTERS
            if (!dip[0].getElemReg().equals(new Vector<Lezione>()))
                System.out
                    .println("Attento: il valore restituito da getElemReg()()\n
non corrisponde al valore passato al costruttore");

            System.out.println("....");

            System.out.println(".....");

            // PROVO addLezione
            try {
                dip[0].addLezione(null);
                System.out

```

```

        .println("Attento: il metodo addLezione deve lanciare una
'IllegalArgumentException' \nse tento di inserire una lezione null: correggi e
riprova. ");

    } catch (NullPointerException e) {
    }
    Lezione app = new Lezione(new Data(12, 10, 2008), "Prima", "dopo",
        new RegStudenti(), 0);
    try {
        dip[0].addLezione(app);
        dip[0].addLezione(app);
        System.out
            .println("Attento: il metodo addLezione deve lanciare una
'DuplicatedLezioniException' \nse tento di inserire una lezione che e' gia presente :
correggi e riprova. ");
    } catch (DuplicatedLezioniException e) {
    }

    Lezione app2 = new Lezione(new Data(15, 10, 2008), "Adesso",
        "Subito", new RegStudenti(), 0);
    dip[0].addLezione(app2);
    if (dip[0].getElemReg().indexOf((Lezione) app2) < dip[0]
        .getElemReg().indexOf((Lezione) app))
        System.out
            .println("Attento: il metodo addLezione deve mantenere l
'elenco delle lezioni ordinato : correggi e riprova. ");

    System.out.println(".....");

    // PROVO removeLezioni
    try {
        dip[0].removeLezioni(null);
        System.out
            .println("Attento: il metodo removeLezioni deve lanciare una
'IllegalArgumentException' \nse tento di rimuovere una lezione null: correggi e
riprova. ");
    } catch (NullPointerException e) {
    }
    Lezione rem = new Lezione(new Data(15, 10, 2008), "Domani",
        "Subito", new RegStudenti(), 0);
    try {
        dip[0].removeLezioni(rem);
        System.out
            .println("Attento: il metodo removeLezioni deve lanciare una
'NullPointerException' \nse tento di rimuovere una lezione che non e' presente nell
'archivio : correggi e riprova. ");
    } catch (NotExistentLezioniException e) {
    }

    // PROVO Modify
    // *****/
    // PROVO CercaNome

```

```
        int x = dip[0].cercaNome("Adesso");
        if (x == -1)
            System.out
                .println("Attento: il metodo CercaNome non funziona come
dovrebbe : correggi e riprova. ");

        // PROVO CercaData
        x = dip[0].cercaData(new Data(12, 10, 2008));
        if (x == -1)
            System.out
                .println("Attento: il metodo CercaNome non funziona come
dovrebbe : correggi e riprova. ");

        // PROVO isIn
        if (!dip[0].isIn(app))
            System.out
                .println("Attento: il metodo isIn non funziona come dovrebbe
: correggi e riprova. ");
        try {
            // Caso Null
            dip[0].isIn(null);
            System.out
                .println("Attento: isIn deve lanciare una
'IllegalArgumentException' \nse la lezione e' uguale a null: correggi e riprova.");
        } catch (NullPointerException e) {
        }

        } catch (Exception ecc) {
            System.out
                .println("\n OOPS!!! Stavo provando la tua classe 'RegLezioni', \
nma si e' verificata un'eccezione:");
            ecc.printStackTrace();
            System.exit(-1);
        }

        System.out.println("OK: non sono riuscito a trovare alcun errore.");
        System.out.println("=====");
    }
}
```