

```

import java.util.Vector;

/**
 * Classe di Test.
 *
 * @author Filippo Fontanelli , Francesca Brogi
 *
 */

public class testClassRegStudenti {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            System.out
                .println("\n=====Provo la classe
'RegStudenti'...\n=====");
            RegStudenti[] dip = new RegStudenti[5];
            System.out.println(".");
            // PROVO COSTRUTTORI
            try {
                // Caso Null
                dip[0] = new RegStudenti(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 RegStudenti(new Vector<Studente>());
            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<Studente>()))
                System.out
                    .println("Attento: il valore restituito da getElemReg()\n
non corrisponde al valore passato al costruttore");

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

            // PROVO TOSTRING
            /**
             * String s = dip[2].toString(); if (s == null ||
             * s.indexOf("Lezione") == -1 || s.indexOf("Argomento") == -1 ||
             * s.indexOf("10") == -1) System.out.println("Attento: sembra che
             * toString() non restituisca\n tutte le informazioni richieste");

```

```

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

        // PROVO addStudenti
        try {
            dip[0].addStudenti(null);
            System.out
                .println("Attento: il metodo addStudenti deve lanciare una
'IllegalArgumentException' \nse tento di inserire uno studente null: correggi e
riprova. ");

            } catch (NullPointerException e) {
            }
            Studente app = new Studente("Mario", "Rossi", 2, new RegLezioni(),
                10);
            try {
                dip[0].addStudenti(app);
                dip[0].addStudenti(app);
                System.out
                    .println("Attento: il metodo addStudenti deve lanciare una
'DuplicatedLezioniException' \nse tento di inserire uno studentee che e' gia presente
: correggi e riprova. ");
            } catch (DuplicatedStudentiException e) {
            }

            Studente app2 = new Studente("Maria", "Rossi", 25,
                new RegLezioni(), 10);
            dip[0].addStudenti(app2);
            if (dip[0].getElemReg().indexOf((Studente) app2) > dip[0]
                .getElemReg().indexOf((Studente) app))
                System.out
                    .println("Attento: il metodo addStudenti deve mantenere l
'elenco degli studenti ordinato : correggi e riprova. ");

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

            // PROVO removeStudenti
            try {
                dip[0].removeStudenti(null);
                System.out
                    .println("Attento: il metodo removeStudenti deve lanciare una
'IllegalArgumentException' \nse tento di rimuovere uno studente null: correggi e
riprova. ");

            } catch (NullPointerException e) {
            }
            Studente rem = new Studente("Mario", "Bianchi", 25,
                new RegLezioni(), 10);
            try {
                dip[0].removeStudenti(rem);
                System.out
                    .println("Attento: il metodo removeStudenti deve lanciare una
'NullPointerException' \nse tento di rimuovere uno studente che non e' presente nell

```

```

'archivio : correggi e riprova. ");
    } catch (NotExistentStudentiException e) {
    }

    // PROVO Modify
    // *****/
    // PROVO CercaNome
    int x = dip[0].cercaMat(2);
    if (x == -1)
        System.out
            .println("Attento: il metodo cercaMat 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 (IllegalArgumentException e) {
    }

    } catch (Exception ecc) {
        System.out
            .println("\n OOPS!!! Stavo provando la tua classe 'RegStudenti',
\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("=====");

}
}

```