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

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        try {
            System.out
                    .println("\n================================\nProvo la classe
'RegistroFrequenzeInterfaccia'...\n================================");
            RegistroFrequenzeImp[] dip = new RegistroFrequenzeImp[5];
            System.out.println(".");
            // PROVO COSTRUTTORI
            try {
                // Caso Null
                dip[0] = new RegistroFrequenzeImp(null, "Test",
                        new RegLezioni(), new RegStudenti(), 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il fileNome e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", null,
                        new RegLezioni(), new RegStudenti(), 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il nome e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", "Test", null,
                        new RegStudenti(), 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il registro lezioni e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", "Test",
                        new RegLezioni(), null, 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
```

```
'NullPointerException' \nse il registro studenti e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp(null, "Test");
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il fileName e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", null);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il nome e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp(null);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il FileNome  e' null: correggi e riprova.");
            } catch (NullPointerException e) {
            }

            // Caso Stringa vuota ed interi < 0
            try {
                dip[0] = new RegistroFrequenzeImp("", "Test", new RegLezioni(),
                        new RegStudenti(), 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'IllegalArgumentException' \nse il fileName e' la stringa vuota: correggi e riprova."
);
            } catch (IllegalArgumentException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", "",
                        new RegLezioni(), new RegStudenti(), 5);
                System.out
                        .println("Attento: il costrutture deve lanciare una
'IllegalArgumentException' \nse il nome e' la stringa vuota: correggi e riprova.");
            } catch (IllegalArgumentException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", "Test",
                        new RegLezioni(), new RegStudenti(), -2);
                System.out
                        .println("Attento: il costrutture deve lanciare una
```

```
'IllegalArgumentException' \nse il numero obbligatorio di presenze e' minore di 0:
correggi e riprova.");
            } catch (IllegalArgumentException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("", "Test");
                System.out
                        .println("Attento: il costrutture deve lanciare una
'IllegalArgumentException' \nse il fileName e' la stringa vuota: correggi e riprova."
);
            } catch (IllegalArgumentException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("Test.txt", "");
                System.out
                        .println("Attento: il costrutture deve lanciare una
'IllegalArgumentException' \nse il nome e' la stringa vuota: correggi e riprova.");
            } catch (IllegalArgumentException e) {
            }

            try {
                dip[0] = new RegistroFrequenzeImp("");
                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse il FileNome  e' la stringa vuota: correggi e riprova.");
            } catch (IllegalArgumentException e) {
            }

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

            // PROVO getters
            dip[0] = new RegistroFrequenzeImp("Test.txt", "Test",
                    new RegLezioni(), new RegStudenti(), 5);

            if (dip[0].getNumObbligatorio() != 0)
                System.out
                        .println("Attento: il valore restituito da getNumObbligatorio
()()\n non corrisponde al valore passato al costruttore");
            if (!dip[0].getName().equals(("Test")))
                System.out
                        .println("Attento: il valore restituito da getName()()\n non
corrisponde al valore passato al costruttore");
            if (!dip[0].getFileName().equals(("Test.txt")))
                System.out
                        .println("Attento: il valore restituito da getFileName()()\n
non corrisponde al valore passato al costruttore");
            // PROVO Set
            dip[0].setNumObbligatorio(10);
            if (dip[0].getNumObbligatorio() != 10)
                System.out
                        .println("Attento: il metodo setNumObbligatorio non modifica
```

```
in modo esatto il valore del numero obbligatorio di presenze");
            System.out.println("....");

            // Provo Load

            dip[2] = new RegistroFrequenzeImp("Test.txt", "Test",
                    new RegLezioni(), new RegStudenti(), 5);

            dip[2].saveTesto();

            dip[2].saveBinario();

            try {
                dip[2].loadLezioni("Studenti.txt");

                System.out
                        .println("Attento: il costrutture deve lanciare una
'NumberFormatException' \nse i dati presenti sul File sono errati: correggi e riprova
.");
            } catch (NumberFormatException e) {
            }

            try {
                dip[2].loadLezioni("Studenti.txt");

                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse i dati presenti sul File sono errati: correggi e riprova
.");
            } catch (NullPointerException e) {
            }

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

            try {
                dip[2].loadLezioni("Lezioni.txt");

                System.out
                        .println("Attento: il costrutture deve lanciare una
'NumberFormatException' \nse i dati presenti sul File sono errati: correggi e riprova
.");
            } catch (NumberFormatException e) {
            }

            try {
                dip[2].loadLezioni("Studenti.txt");

                System.out
                        .println("Attento: il costrutture deve lanciare una
'NullPointerException' \nse i dati presenti sul File sono errati: correggi e riprova
.");
            } catch (NullPointerException e) {
```

```
            }

            // non occorre

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

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

    }
}
```