

```
import java.io.Serializable;

/**
 * Tipo record non modificabile relativo alla data di una lezione. L'oggetto e'
 * serializabile.
 *
 * @author Filippo Fontanelli , Francesca Brogi
 */
public class Data implements Comparable<Data>, Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 5L;

    /**
     * Anno della Data.
     */
    private int anno;

    /**
     * Mese della Data.
     */
    private int mese;

    /**
     * Giorno della Data.
     */
    private int giorno;

    // *****

    /**
     *
     * Crea una nuova Data. Se un parametro non rispetta le caratteristiche
     * richieste viene sollevata un'eccezione.
     *
     * @param giorno
     *         Stringa rappresentante il nome.
     * @param mese
     *         Stringa rappresentante il cognome.
     * @param anno
     *         int rappresentante la matricola.
     *
     * @exception <code>IllegalArgumentException</code> se : (giorno < 1) ||
     *         (giorno > 31) ; se ((mese < 1) || (mese > 12)) oppure se (anno
     *         < 1970)) giorno o il
     */
    public Data(int giorno, int mese, int anno) throws IllegalArgumentException {
        if (((giorno < 1) || (giorno > 31)) || ((mese < 1) || (mese > 12))
            || (anno < 1970))
    }
}
```

```
        throw new IllegalArgumentException("Data::Costruttore");
        this.giorno = giorno;
        this.mese = mese;
        this.anno = anno;
    }

    /**
     *
     * Crea una nuova Data con tutti i valori uguali a 0.
     */
    public Data() {
        this.giorno = 00;
        this.mese = 00;
        this.anno = 00;
    }

    /**
     * Compara oggetti di tipo Data.
     *
     * @param obj
     *         oggetto di tipo data.
     *
     * @return un intero <0 se la Data corrente precede la Data obj, 0 se sono
     *         uguali oppure un'intero >0 altrimenti cioe': >0 se anno >
     *         obj.anno <0 se anno < obj.anno SE anno = obj.anno: >0 se data >
     *         obj.data <0 se data < obj.data SE data = obj.data: > 0 se giorno
     *         > obj.giorno < 0 se giorno < obj.giorno 0 se giorno = obj.giorno
     */
    public int compareTo(Data obj) {

        if (anno != obj.anno)
            return anno - obj.anno;
        else {
            if (mese != obj.mese)
                return mese - obj.mese;
            else
                return giorno - obj.giorno;
        }
    }

    /**
     * Metodo costruito sulla base del compareTo.
     *
     * @param obj
     *         oggetto di tipo qualsiasi.
     *
     * @return <code>true</code> se i due oggetti sono uguali cioe' se hanno stessa
data.
     *         altrimenti ritorna <code>false</code>.
    */
```

```

    */
    public boolean equals(Object obj) {
        if (obj == null)
            return false;
        if (!(obj instanceof Data))
            return false;

        return (this.compareTo((Data) obj) == 0);
    }

    public static Data parseData(String x) {
        String n = "";
        int FSTindex = -1;
        int SNDindex = -1;
        boolean control = false;
        int anno = 0, mese = 0, giorno = 0;

        while (!control) {
            do {
                n = x;
                FSTindex = n.indexOf('/');
                SNDindex = n.indexOf('/', 3);
            } while (FSTindex != 2 || SNDindex != 5 || n.length() != 10); // cicla
            // finche'
            // '/'
            // non
            // ✦ in
            // 2 e
            // in 5
            // posizione

            String year = n.substring(6, 10);
            String month = n.substring(3, 5);
            String day = n.substring(0, 2);

            try {
                anno = Integer.parseInt(year);
                mese = Integer.parseInt(month);
                giorno = Integer.parseInt(day);
                // se non ci sono problemi con la conversione, sicuramente
                // sappiamo che i
                // numeri inseriti sono positivi. Infatti, un eventuale segno
                // '-' posto dinanzi
                // un numero, comporterebbe lo shifting verso destra di tutte le
                // posizioni
                // dei caratteri. Non riusciremmo quindi ad uscire dal
                // precedente ciclo.
                control = true;

                // controllo sulla congruenza dei valori
                String s = "Attenzione, la data inserita non e' corretta:";
                if (giorno < 1 || giorno > 31) {
                    s = s + " (GIORNI)";
                }
            }
        }
    }

```

```
        control = false;
    }
    if (mese < 1 || mese > 12) {
        s = s + " (MESI)";
        control = false;
    }
    if (anno < 1900) {
        s = s + " (ANNO)";
        control = false;
    }
    if (!control)
        System.out.println(s);

    } catch (NumberFormatException e) {
        System.out
            .println("Attenzione, il formato della data inserita non e'
corretto.");
    }

    }

    return new Data(giorno, mese, anno);

}

/**
 * Restituisce la rappresentazione testuale dell'oggetto.
 *
 *
 * Esempio: "10/02/1991".
 */
public String toString() {
    String mese = "";
    String giorno = "";

    if (this.mese < 10)
        mese = "/" + this.mese;
    else
        mese = "/" + this.mese;

    if (this.giorno < 10)
        giorno = "0" + this.giorno;
    else
        giorno = "" + this.giorno;
    return giorno + mese + "/" + this.anno;
}

}
```