

```

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;

/**
 * Una semplice classe per leggere stringhe e numeri dallo standard input. Con
 * l'aggiunta di qualche metodo specifico al progetto Registro Frequenze.
 *
 * @author Filippo Fontanelli , Francesca Brogi
 */

public class SafeInput extends Input {

    private static BufferedReader reader = new BufferedReader(
        new InputStreamReader(System.in));

    /**
     * Legge una linea di input. Nell'improbabile caso di una IOException, il
     * programma termina.
     *
     * @return restituisce la linea di input che l'utente ha battuto.
     */
    public static String readLineSafe() {
        boolean inizializzata = false;
        System.out.print("--->\t\t");
        String inputLine = "";
        do {
            try {
                inputLine = reader.readLine();
                for (int i = 0; i < inputLine.length(); i++) {
                    if (!((inputLine.charAt(i) >= 'A' && inputLine.charAt(i) <= 'Z')
                        || (inputLine.charAt(i) >= 'a' && inputLine
                            .charAt(i) <= 'z')
                        || (inputLine.charAt(i) == ' ') || (inputLine
                            .charAt(i) == '*'))))
                        throw new IllegalArgumentException();
                }
                inizializzata = true;
            } catch (IllegalArgumentException e) {
                // Se il valore inaserto nn è un numero entro nel catch e non
                // setto il boolean
                System.out
                    .print("Attenzione dammi una Stringa con caratteri congrui...
!\n");
                System.out.print("--->\t\t");
            } catch (IOException e) {
                System.out
                    .print("Attenzione dammi una Stringa con caratteri congrui...
!\n");
                System.out.print("--->\t\t");
            }
        } while (!inizializzata);
        // finche' nn è inizializzata correttamente...
    }
}

```

```
        return inputLine;
    }

    /**
     * Legge una linea di input e la converte in un intero. Eventuali spazi
     * bianchi prima e dopo l'intero vengono ignorati. Se il valore passato per
     * input non è un'intero, il metodo cicla fino a quando non avrà un input
     * accettabile.
     *
     * @return l'intero dato in input dall'utente
     */
    public static int readInt() {
        int n = 0;
        boolean inizializzata = false;

        do {
            try {
                String inputString = readLine();
                inputString = inputString.trim();
                n = Integer.parseInt(inputString);
                inizializzata = true;

            } catch (java.lang.NumberFormatException e) {
                // Se il valore inserito nn è un numero entro nel catch e non
                // setto il boolean
                System.out.print("Attenzione dammi un intero!\n");
            }
        } while (!inizializzata);
        // finche' nn è inizializzata correttamente...

        return n;
    }

    /**
     * Legge una linea di input e la converte in un numero in virgola mobile.
     * Eventuali spazi bianchi prima e dopo il numero vengono ignorati. Se il
     * valore passato per input non è un numero in virgola mobile, il metodo
     * cicla fino a quando non avrà un input accettabile.
     *
     * @return il numero dato in input dall'utente
     */
    public static double readDouble() {
        double x = 0;
        boolean inizializzata = false;

        do {
            try {
                String inputString = readLine();
                inputString = inputString.trim();
                x = Double.parseDouble(inputString);
                inizializzata = true;
            } catch (java.lang.NumberFormatException e) {
```

```
        System.out.print("Attenzione dammi un double!\n");
    }
} while (!inizializzata);

return x;
}

/**
 * Legge una linea di input e ne estrae il primo carattere.
 *
 * @return il primo carattere della riga data in input dall'utente
 */
public static char readChar() {
    System.out.print("--->\t\t");
    String inputString = readLine();
    char c = inputString.charAt(0);
    return c;
}

// *****
// Metodi di acquisizione dati specifici //
// *****

/**
 *
 *
 * Legge una linea di input usando il metodo SafeInput.readInt(). L'input è
 * accettato solo se è un'intero compreso tra 1 e 4
 *
 * @return l'intero dato in input dall'utente
 *
 */
public static int interoUnoQuattro() {
    int n = -1;

    do {
        System.out.print("(Intero compreso tra 1 e 4): ");
        n = SafeInput.readInt();
    } while (n < 1 || n > 4);

    return n;
}

/**
 *
 *
 * Legge una linea di input usando il metodo SafeInput.readInt(). L'input è
 * accettato solo se è un'intero compreso tra 1 e 3
 *
 * @return l'intero dato in input dall'utente
 *
 */
public static int interoUnoTre() {
    int n = -1;
```

```

    do {
        System.out.println("(Intero compreso tra 1 e 3): ");
        n = SafeInput.readInt();
    } while (n < 1 || n > 3);

    return n;
}

/**
 * Stringa non vuota
 */
public static String stringaNonVuota() {
    String n = "";

    do {
        System.out.println(" (Stringa nn vuota): ");
        n = SafeInput.readLine();
    } while ((n.equals("")) || (n.trim().equals("")));

    return n;
}

/**
 * Restituisce un oggetto di tipo Date a partire da una String.
 *
 * @return <code>Date</code>
 */
public static Data readDate() {
    String n = "";
    int FSTindex = -1;
    int SNDindex = -1;
    boolean control = false;
    int anno = 0, mese = 0, giorno = 0;

    while (!control) {
        do {
            System.out.print(" (gg/mm/aaaa): ");
            n = SafeInput.readLine();
            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 è corretta:";
    if (giorno < 1 || giorno > 31) {
        s = s + " (GIORNI)";
        control = false;
    }
    if (mese < 1 || mese > 12) {
        s = s + " (MESI)";
        control = false;
    }
    if (anno < 1970) {
        s = s + " (ANNO)";
        control = false;
    }
    if (!control)
        System.out.println(s);

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

}

return new Data(giorno, mese, anno);

}

}
```