

```

/**
 * TDA che rappresenta una scatola che può essere riempita, svuotata
 * e ruotata
 * @author Paolo
 * @version 1.0
 */
public class Scatola {
    private int h;
    private int l;
    private int c;

    /**
     * Crea una scatola vuota di dimensione 5x5
     */
    public Scatola() {
        this(5,5);
    }

    /**
     * Crea una scatola vuota.
     * @param h altezza della scatola
     * @param l larghezza della scatola
     */
    public Scatola(int h, int l) {
        this.h = h;
        this.l = l;
        this.c = 0;
    }

    /**
     * Aggiunge una certa quantità alla scatola.
     * @param x quantità con cui riempire la scatola. Se < 0,
     * negativo viene considerato uguale a 0. L'effetto di aggiungere una
     * quantità superiore alla capacità della scatola < 0,
     * lo stesso che si ottiene riempiendo la scatola.
     * @return un riferimento alla scatola dopo che < 0, stata riempita
     */
    public Scatola riempi(int x) {
        x = x < 0 ? 0 : x;
        c = (x >= h*l-c) ? h*l : c+x;
        return this;
    }

    /**
     * Toglie una certa quantità alla scatola.
     * @param x quantità da levare alla scatola. Se < 0,
     * negativo viene considerato uguale a 0. L'effetto di togliere una quantità
     * superiore al contenuto della scatola < 0, lo stesso che si
     * ottiene svuotando la scatola.
     * @return un riferimento alla scatola dopo che < 0, stata svuotata
     */
    public Scatola svuota(int x) {
        x = x < 0 ? 0 : x;
        c = (x >= c) ? 0 : c-x;
        return this;
    }

    /**
     * @return restituisce in forma di stringa una scatola di "*" riempita
     * con "o".
     * Se ad esempio la scatola < 0, alta 3, larga 2 e riempita di 3
     * elementi, il metodo restituirà:
     * <pre>
     * **
     * *o
     * oo
     */

```

```
* </pre>
*/
public String toString() {
    String temp = "";
    for (int i = 1; i<=h*l; i++) {
        temp += i <= h*l-c ? "*" : "0";
        if (i%l == 0)
            temp += "\n";
    }
    return temp;
}

/**
 * Ruota la scatola scambiando i valori di altezza e larghezza.
 * @return un riferimento alla scatola stessa dopo che &grave; stata
 * ruotata.
 */
public Scatola ruota() {
    int temp = h;
    h = l;
    l = temp;
    return this;
}

/**
 * @return l'altezza della scatola.
 */
public int getAltezza() {
    return h;
}

/**
 * @return la larghezza della scatola.
 */
public int getLarghezza() {
    return l;
}

/**
 * @return quanto &grave; piena la scatola.
 */
public int getContenuto() {
    return c;
}
}
```

```
/**
 * TDA che rappresenta una scatola che può essere riempita con un
 * determinato carattere
 * @author Paolo
 * @version 1.0
 */
public class ScatolaCaratteri extends Scatola {
    private char carattere;
    public ScatolaCaratteri(int h, int l, char c) {
        super(h,l);
        this.carattere = c;
    }

    /**
     * @return restituisce in forma di stringa una scatola di "*" riempita con
     * il carattere della scatola.
     * Se ad esempio la scatola &grave; alta 3, larga 2, riempita di 3
     * elementi e il suo carattere &grave; 'x', il metodo restituir&grave;;
     * <pre>
     * **
     * *x
     * xx
     * </pre>
     */
    public String toString() {
        String temp = "";
        int h = getAltezza();
        int l = getLarghezza();
        int c = getContenuto();
        for (int i = 1; i<=h*l; i++) {
            temp += i <= h*l-c ? '*' : carattere;
            if (i%l == 0)
                temp += "\n";
        }
        return temp;
    }

    /**
     * @return il carattere della scatola
     */
    public char getCarattere() {
        return this.carattere;
    }

    public ScatolaCaratteri setCarattere(char carattere) {
        this.carattere = carattere;
        return this;
    }
}
```

```
public class UsaScatola {
    public static void main(String[] args) {
        Scatola s1 = new Scatola(2,3);
        stampaScatola(s1);
        s1.riempi(3);
        stampaScatola(s1);
        s1.ruota();
        stampaScatola(s1);
        ScatolaCaratteri s2 = new ScatolaCaratteri(3,5,'a');
        stampaScatola(s2);
        Scatola s3 = new ScatolaCaratteri(4,6,'b');
        stampaScatola(s3);
    }
    static void stampaScatola(Scatola s) {
        System.out.println(s);
    }
}
```