

Java Eccezioni - Lettura scanner

```
import java.util.*;

class Solver{
    public static void main(String[] args){
        Macchinetta m = new Macchinetta();
        int soldi = 0;
        Scanner scanner = new Scanner(System.in);
        while(scanner.hasNext()){
            String choose = scanner.next();
            switch(choose){
                case "B":
                    m.aggiungiProdotto(new Bevanda(scanner.next(),
scanner.nextInt(), scanner.nextInt()));
                    break;
                case "M":
                    m.aggiungiProdotto(new Merendina(scanner.next(),
scanner.nextInt(), scanner.nextInt()));
                    break;
                case "S":
                    try{
                        int increment = scanner.nextInt();
                        if(increment < 0) throw new
Exception("negative value inserted");
                        soldi += increment;

                    }
                    catch(Exception e){
                        System.out.println(e.getMessage());
                        return;
                    }
                    break;
                case "CB":
                    Bevanda b = m.compraBevanda(soldi,
scanner.nextInt());
                    if(b != null){
                        System.out.println(b.toString());
                        soldi -= b.getCosto();
                    }
                    break;
                case "CM":
                    Merendina mer = m.compraMerendina(soldi,
scanner.nextInt());
                    if(mer != null){
                        System.out.println(mer.toString());
                        soldi -= mer.getCosto();
                    }
            }
        }
    }
}
```

```

        break;
    case "SORT":
        m.sort();
        break;
    case "F":
        return;
    }
}
}
}
}

```

```

abstract class Prodotto{
    private int costo;
    private String nome;

    public Prodotto(int c, String n){
        costo = c;
        nome = n;
    }

    public int getCosto(){
        return costo;
    }

    public String getNome(){
        return nome;
    }

    public abstract String toString();
}

```

```

class Bevanda extends Prodotto{
    private int volume;

    public Bevanda(String n, int c, int v){
        super(c, n);
        volume = v;
    }

    public int getVolume(){
        return volume;
    }

    public String toString(){
        return "Il prodotto selezionato è " + super.getNome() + " ha
un costo di " + super.getCosto() + " e un volume di " + volume;
    }
}

```

```
}  
}
```

```
class Merendina extends Prodotto{  
    private int calorie;  
  
    public Merendina(String n, int c, int cal){  
        super(c, n);  
        calorie = cal;  
    }  
  
    public int getCalorie(){  
        return calorie;  
    }  
  
    public String toString(){  
        return "Il prodotto selezionato è " + super.getNome() + " ha  
un costo di " + super.getCosto() + " e " + calorie + "calorie";  
    }  
}
```

```
import java.util.*;  
  
class Macchinetta{  
    private ArrayList<Prodotto> prodotti;  
  
    public Macchinetta(){  
        prodotti = new ArrayList<Prodotto>();  
    }  
  
    public void aggiungiProdotto(Prodotto element){  
        prodotti.add(element);  
        return;  
    }  
  
    public Merendina compraMerendina(int soldi, int calorie){  
        for(Prodotto element: prodotti){  
            if(element instanceof Merendina){  
                if(element.getCosto() <= soldi && ((Merendina)  
element).getCalorie() <= calorie){  
                    prodotti.remove(element);  
                    return ((Merendina) element);  
                }  
            }  
        }  
        return null;  
    }  
  
    public Bevanda compraBevanda(int soldi, int volume){
```

```

        for(Prodotto element: prodotti){
            if(element instanceof Bevanda){
                if(element.getCosto() <= soldi && ((Bevanda)
element).getVolume() >= volume){
                    prodotti.remove(element);
                    return ((Bevanda) element);
                }
            }
        }
        return null;
    }

    public void sort(){
        System.out.println("*****");
        for(Prodotto el: prodotti){
            System.out.println(el.toString());
        }
        System.out.println("*****");

        Collections.sort(prodotti, new ComparatorProdotti());

        System.out.println("*****");
        for(Prodotto el: prodotti){
            System.out.println(el.toString());
        }
        System.out.println("*****");
        return;
    }
}

```

```

B coca 10 10
B cocagrande 15 30
B cocamedia 10 15
M fiesta 10 20
M duplo 5 10
M crackers 5 6
CB 1
S 14
CB 1
CM 5
CM 10
S 20
CM 10
CB 1
S 10
CM

```