

```
1  /*
2      ES01
3      Sul prezzo di un prodotto
4      viene praticato uno sconto del 20
5      % se costa meno di 500€
6      e del 25% per prezzi
7      superiori a 500€. Calcolare il
8      prezzo da pagare.
9  */
10 import java.util.Scanner;
11
12 public class Main {
13     public static void main(
14         String[] args) {
15         final double sconto1 = 0.
16             2;
17         final double sconto2 = 0.
18             25;
19         Scanner input = new
20             Scanner(System.in);
21         double prc, prcFinale;
22         System.out.print("
23             Inserire prezzo: ");
24         prc = input.nextDouble();
25         if(prc <= 500)
```

```
18         prcFinale = prc * (1-
    sconto1);
19         else
20         prcFinale = prc * (1-
    sconto2);
21         System.out.println("
Prezzo finale: " + prcFinale);
22     }
23 }
24
25 //-----
    -----
26 //ES02
27 import java.util.Scanner;
28
29 public class Main {
30     public static void main(
String[] args) {
31         Scanner input = new
Scanner(System.in);
32         int temp;
33         System.out.print("
Inserire temperatura: ");
34         temp = input.nextInt();
35         if(temp < 0)
36             System.out.println("

```

```
36 Molto freddo");
37         else if(temp < 10)
38             System.out.println("
Freddo");
39         else if(temp < 16)
40             System.out.println("
Fresco");
41         else if(temp < 25)
42             System.out.println("
Clima mite");
43         else if(temp < 32)
44             System.out.println("
Caldo");
45         else
46             System.out.println("
Molto caldo");
47     }
48 }
49
50 //-----
-----
51 /*
52     ES03
53     L'indice di massa corporea si
        calcola dividendo il proprio
        peso espresso in kg per il
```

```
53 quadrato dell'altezza
54     espressa in metri: IMC =
    massa corporea / altezza2.
55     Secondo l'Organizzazione
    Mondiale della Sanità l'IMC, o
    indice di massa corporea,
56     raggruppabile in 4 categorie
    : sottopeso (IMC al di sotto di
    19), normale (IMC compreso tra 19
    e 24),
57     sovrappeso (IMC compreso tra
    25 e 30), obesità (IMC al di
    sopra di 30).
58 */
59
60 import java.util.Scanner;
61
62 public class Main {
63     public static void main(
        String[] args) {
64         Scanner input = new
            Scanner(System.in);
65         double wgt, hgt, imc;
66         System.out.print("
Inserire peso (in kg): ");
67         wgt = input.nextDouble();
```

```
68         System.out.print("
    Inserire altezza (in m): ");
69         hgt = input.nextDouble();
70         imc = wgt / Math.pow(hgt,
    2);
71         System.out.println("Imc
    : " + imc);
72         if(imc < 19)
73             System.out.println("
    Sottopeso");
74         else if(imc < 24)
75             System.out.println("
    Normale");
76         else if(imc < 30)
77             System.out.println("
    Sovrappeso");
78         else
79             System.out.println("
    Obeso");
80     }
81 }
82
83 //-----
    -----
84 /*
85     ES04
```

```
86      Scrivi un algoritmo che  
      legga una sequenza di voti (tra  
      1 e 10). Quando viene inserito 0  
      , viene stampata la  
87      media dei voti inseriti.  
88 */  
89  
90 import java.util.Scanner;  
91  
92 public class Main {  
93     public static void main(  
        String[] args) {  
94         Scanner input = new  
        Scanner(System.in);  
95         int voto, tot = 0, num  
        = 0;  
96         float media;  
97         System.out.print("  
        Inserire voto (0 per uscire): "  
        );  
98         voto = input.nextInt();  
99         while(voto != 0){  
100             tot += voto;  
101             num++;  
102             System.out.print("  
            Inserire voto (0 per uscire): "
```

```
102 );
103         voto = input.nextInt
104         ();
105     }
106     if(num > 0){
107         media = (float) tot/
108         num;
109         System.out.println("
110 Media: " + media);
111     }
112     else
113         System.out.println("
114 Nessun voto inserito");
115     }
116 }
117 //-----
118 -----
119 /*
120     ES05
121     Leggere una sequenza di
122     numeri interi. Terminare la
123     lettura quando si incontra un
124     valore pari a 9999.
125     Determinare quanti sono
126     stati i valori pari e i valori
```

```
118 dispari.
119 */
120
121 import java.util.Scanner;
122
123 public class Main {
124     public static void main(
125         String[] args) {
126         Scanner input = new
127             Scanner(System.in);
128         int n, cP = 0, cD = 0;
129         System.out.print("
130             Inserire numero (9999 per uscire
131             ): ");
132         n = input.nextInt();
133         while(n != 9999){
134             if(n % 2 == 0)
135                 cP++;
136             else
137                 cD++;
138             System.out.print("
139                 Inserire numero (9999 per uscire
140                 ): ");
141             n = input.nextInt();
142         }
143         if(cP != 0 || cD != 0){
```



```
138         System.out.println("
    Pari: " + cP);
139         System.out.println("
    Dispari: " + cD);
140     }
141     else
142         System.out.println("
    Nessun numero inserito");
143     }
144 }
145
146 //-----
    -----
147 /*
148     ES06
149     Crea un programma
    organizzato a menu che permette
    di calcolare le aree di figure
    geometriche:
150     triangolo, rettangolo, rombo
    , trapezio, cerchio, poligono
    regolare.
151     Vincolo: utilizzare una
    funzione per ciascuna
    funzionalità.
152 */
```

```
153
154 import java.util.Scanner;
155
156 public class Main {
157     public static void main(
158         String[] args) {
159         int sc;
160         do{
161             menu();
162             sc = lgInt(0,6,"
163             Inserire scelta: ");
164             switch(sc){
165                 case 1:{
166                     triangolo();
167                     break;
168                 }
169                 case 2:{
170                     rettangolo
171                     ();
172                     break;
173                 }
174                 case 3:{
175                     rombo();
176                     break;
177                 }
178                 case 4:{
```

```
176         trapezio();
177         break;
178     }
179     case 5:{
180         cerchio();
181         break;
182     }
183     case 6:{
184         poligonoReg
185     );
186         break;
187     }
188     }while(sc != 0);
189 }
190
191 public static void menu(){
192     System.out.println("
Figure di cui calcolare l'area:"
);
193     System.out.println("1)
Triangolo;");
194     System.out.println("2)
Rettangolo;");
195     System.out.println("3)
Rombo;");
```

```
196         System.out.println("4)
    Trapezio;");
197         System.out.println("5)
    Cerchio;");
198         System.out.println("6)
    Poligono regolare;");
199         System.out.println("0)
    Termina il programma.");
200     }
201
202     public static void triangolo
    () {
203         double base, hgt;
204         base = lgDbL(1, Double.
    MAX_VALUE, "Inserire base: ");
205         hgt = lgDbL(1, Double.
    MAX_VALUE, "Inserire altezza: "
    );
206         System.out.println("Area
    : " + (base*hgt)/2);
207     }
208
209     public static void
    rettangolo() {
210         double base, hgt;
211         base = lgDbL(1, Double.
```

```
211 MAX_VALUE, "Inserire base: ");
212         hgt = lgDbℓ(1, Double.
MAX_VALUE, "Inserire altezza: "
);
213         System.out.println("Area
: " + base*hgt);
214     }
215
216     public static void rombo(){
217         double dMag, dMin;
218         dMag = lgDbℓ(1, Double.
MAX_VALUE, "Inserire diagonale
maggiore: ");
219         dMin = lgDbℓ(1, Double.
MAX_VALUE, "Inserire diagonale
minore: ");
220         System.out.println("Area
: " + (dMag*dMin)/2);
221     }
222
223     public static void trapezio
(){
224         double bMag, bMin, hgt;
225         bMag = lgDbℓ(1, Double.
MAX_VALUE, "Inserire base
maggiore: ");
```

```
226         bMin = lgDb1(1, Double.  
MAX_VALUE, "Inserire base minore  
: ");  
227         hgt = lgDb1(1, Double.  
MAX_VALUE, "Inserire altezza: "  
);  
228         System.out.println("Area  
: " + (bMag+bMin)*hgt/2);  
229     }  
230  
231     public static void cerchio  
    () {  
232         double r;  
233         r = lgDb1(1, Double.  
MAX_VALUE, "Inserire raggio: ");  
234         System.out.println("Area  
: " + Math.pow(r,2)*Math.PI);  
235     }  
236  
237     public static void  
    poligonoReg() {  
238         double per, apo;  
239         per = lgDb1(1, Double.  
MAX_VALUE, "Inserire perimetro  
: ");  
240         apo = lgDb1(1, Double.
```

```
240 MAX_VALUE, "Inserire apotema: "
    );
241         System.out.println("Area
    : " + per*apo/2);
242     }
243
244     public static int lgInt(int
    vmin, int vmax, String mex){
245         int x;
246         System.out.print(mex);
247         Scanner sc = new Scanner
    (System.in);
248         x = sc.nextInt();
249         while(x<vmin || x>vmax){
250             System.out.println("
    Errore");
251             System.out.print(mex
    );
252             x = sc.nextInt();
253         }
254         return x;
255     }
256
257     public static double lgDb1(
    double vmin, double vmax, String
    mex){
```

```
258         double x;
259         System.out.print(mex);
260         Scanner sc = new Scanner
            (System.in);
261         x = sc.nextDouble();
262         while(x<vmin || x>vmax){
263             System.out.println("
Errore");
264             System.out.print(mex
                );
265             x = sc.nextDouble();
266         }
267         return x;
268     }
269 }
270
271 //-----
    -----
272 /*
273     ES07
274     Scrivi una funzione che dati
        due punti A e B, restituisce la
        misura della distanza tra A e B
        .
275 */
276
```



```
277 import java.util.Scanner;
278
279 public class Main {
280     public static void main(
        String[] args) {
281         double x1, x2, y1, y2;
282         x1 = lgDb1(Double.
            MIN_VALUE, Double.MAX_VALUE, "
            Inserire x del primo punto: ");
283         y1 = lgDb1(Double.
            MIN_VALUE, Double.MAX_VALUE, "
            Inserire y del primo punto: ");
284         x2 = lgDb1(Double.
            MIN_VALUE, Double.MAX_VALUE, "
            Inserire x del secondo punto: "
        );
285         y2 = lgDb1(Double.
            MIN_VALUE, Double.MAX_VALUE, "
            Inserire y del secondo punto: "
        );
286         System.out.println("
            Distanza: " + dist(x1, x2, y1,
            y2));
287     }
288
289     public static double dist(
```

```
289 double x1, double x2, double y1
    , double y2){
290     double dist;
291     dist = Math.sqrt(Math.
pow(x2-x1,2) + Math.pow(y2-y1,2
));
292     return dist;
293 }
294
295 public static double lgDbL(
double vmin, double vmax, String
mex){
296     double x;
297     System.out.print(mex);
298     Scanner sc = new Scanner
(System.in);
299     x = sc.nextDouble();
300     while(x<vmin || x>vmax){
301         System.out.println("
Errore");
302         System.out.print(mex
);
303         x = sc.nextDouble();
304     }
305     return x;
306 }
```

```
307 }
```