

```

1  /* VEICOLI IN GARAGE
2  Si realizzi una applicazione java per la gestione di un garage secondo le specifiche:
3      - Il garage ha al max 15 posti ognuno dei quali è identificato da un num a
        partire da 0 e può ospitare solo auto moto e furgoni. Partendo dalla classe base
        veicolo a motore V, la si estenda, realizzando anche le classi che modellano le
        entità furgone (F) auto (A) e moto (M).
4      - Ridefinire il metodo toString in modo che ogni entità possa esternalizzare in
        forma di stringa tutte le informazioni che la riguardano.
5      - Si implementi una classe che modelli il garage sopradescritto offrendo le
        seguenti operazioni di gestione:
6          1] immissione di un nuovo veicolo
7          2] estrazione dal garage del veicolo che occupa un determinato posto
            (ritornare l'istanza del veicolo stesso)
8          3] stampa della situazione corrente dei posti nel garage veicolo:
9              marca, anno, cilindrata;
10             auto:   porte, alimentazione (diesel/benzina)
11             moto:   tempi
12             furgone:capacità */
13
14  import java.util.Scanner;
15  class Main {
16      final static int max = 15;
17
18      public static void main (String args []) {
19          Scanner in = new Scanner(System.in);
20          String mar;           //marca
21          char db;              //diesel o benzina
22          int cil, year, par4;
23          char ch, scelta;
24          int i, j = 0;
25          V T[] = new V[max];   //istanziamento del vettore
26          do {
27              System.out.print("m,a,f?:"); ch = in.next().charAt(0);
28              System.out.print("marca:"); mar = in.next();
29              System.out.print("anno:"); year = in.nextInt();
30              System.out.print("cilindrata:"); cil = in.nextInt();
31              switch(ch) {
32                  case 'a': {
33                      System.out.print("porte:");
34                      par4 = in.nextInt();
35
36                      System.out.print("diesel o benzina?(d/b):");
37                      db = in.next().charAt(0);
38
39                      A a = new A(j, mar, year, cil, par4, db);
40                      T[j] = a;
41                  }break;
42
43                  case 'f': {
44                      System.out.print("carico:"); par4 = in.nextInt();
45
46                      F f = new F(j, mar, year, cil, par4);
47                      T[j] = f;
48                  }break;
49
50                  case 'm': {
51                      System.out.print("tempi:"); par4 = in.nextInt();
52
53                      F f = new F(j, mar, year, cil, par4);
54                      T[j] = f;
55                  }break;
56
57                  default: System.out.println("scelta non valida");
58              } //fine switch
59
60              j++;
61              for(i = 0; i < j; i++)
62                  System.out.println(T[i]);
63              System.out.print("continui?(s/n):");
64              scelta = in.next().charAt(0);
65          } while(scelta != 'n' && j < max); //fine do-while
66
67

```

```

68         System.out.print("indice mezzo da cercare:"); i = in.nextInt();
69         if(i < j && i >= 0)
70             System.out.println(T[i]);
71         else
72             System.out.println("elemento non presente");
73     } //fine main
74 } //fine classe Main
75
76
77 class V {
78     protected int id;
79     protected String marca;
80     protected int anno, cyl;
81
82     V(int j,String s, int a, int c){
83         id = j;
84         marca = s;
85         anno = a;
86         cyl = c;
87     }
88 } //fine classe V
89
90 class A extends V {
91     private int porte;
92     private char alimentazione;
93     A(int j, String s, int a, int c, int p, char d) {
94         super(j, s, a, c); porte = p; alimentazione = d;
95     }
96
97     public int getid() {return id;}
98     public String getm() {return marca;}
99     public int geta() {return anno;}
100    public int getc() {return cyl;}
101    public int getp() {return porte;}
102    public char getd() {return alimentazione;}
103
104    public String toString() {
105        String st = getid()+"] "+getm()+" "+geta()+" "+getc()+" "+getp()+" "+getd();
106        return st;
107    }
108 } //fine classe A
109
110 class F extends V {
111     private int carico;
112     F(int j, String s, int a, int c, int ca) {super(j, s, a, c); carico = ca;}
113
114     public int getid() {return id;}
115     public String getm() {return marca;}
116     public int geta() {return anno;}
117     public int getc() {return cyl;}
118     public int getca() {return carico;}
119
120     public String toString() {
121         String st = getid()+"] "+getm()+" "+geta()+" "+getc()+" "+getca();
122         return st;
123     }
124 } //fine classe F
125
126 class M extends V {
127     private int tempi;
128     M(int j, String s, int a, int c, int t) {super(j, s, a, c); tempi = t;}
129
130     public int getid() {return id;}
131     public String getm() {return marca;}
132     public int geta() {return anno;}
133     public int getc() {return cyl;}
134     public int gett() {return tempi;}
135
136     public String toString() {
137         String st = getid()+"] "+getm()+" "+geta()+" "+getc()+" "+gett();
138         return st;
139     }
140 } //fine classe M

```

```
141
142
143
144 OUTPUT VIDEO
145 m,a,f?:a
146 marca:Fiat
147 anno:1000
148 cilindrata:100
149 porte:1
150 diesel o benzina?(d/b):d
151 0] Fiat 1000 100 1 d
152 continui?(s/n):s
153 m,a,f?:m
154 marca:Honda
155 anno:2000
156 cilindrata:200
157 tempi:2
158 0] Fiat 1000 100 1 d
159 1] Honda 2000 200 2
160 continui?(s/n):s
161 m,a,f?:f
162 marca:Ford
163 anno:3000
164 cilindrata:300
165 carico:3
166 0] Fiat 1000 100 1 d
167 1] Honda 2000 200 2
168 2] Ford 3000 300 3
169 continui?(s/n):s
170 m,a,f?:a
171 marca:MG
172 anno:4000
173 cilindrata:400
174 porte:4
175 diesel o benzina?(d/b):b
176 0] Fiat 1000 100 1 d
177 1] Honda 2000 200 2
178 2] Ford 3000 300 3
179 3] MG 4000 400 4 b
180 continui?(s/n):n
181 indice mezzo da cercare:3
182 3] MG 4000 400 4 b
183
```