

```

1: /**
2:  * Name: Parallelepipedo.h      classe
3:  * Author: Gerboni
4:  * Date: 12/02/19 19:48
5:  * Description: Parallelepipedo rettangolo
6:  *
7:  *          Si deve poter:
8:  *          - Visualizzare le dimensioni;
9:  *          - Calcolare e comunicare la superficie totale;
10: *          - Calcolare e comunicare il volume;
11: *          - Comunicare se il parallelepipedo è un cubo.
12: */
13:
14:
15: class Parallelepipedo {
16:     private:
17:         float lunghezza;      //attributi (variabili membro)
18:         float larghezza;
19:         float altezza;
20:
21:     public:
22:         // --- metodi (funzioni membro)
23:
24:         // costruttori e distruttori
25:         Parallelepipedo(){          // costruttore di default
26:             lunghezza = 0;
27:             larghezza = 0;
28:             altezza = 0;
29:         }
30:         Parallelepipedo(float x, float y, float z){ // costruttore personalizzato
31:             lunghezza = x;
32:             larghezza = y;
33:             altezza = z;
34:         }
35:
36:         ~Parallelepipedo() {          // distruttore personalizzato
37:             cout<<"\nParallelepipedo distrutto"<<endl;
38:             system("pause");
39:         }
40:
41:         // metodi set          per assegnare un valore agli attributi
42:         void setLunghezza(float x)  { lunghezza = x; }
43:         void setLarghezza(float x)  { larghezza = x; }
44:         void setAltezza(float x)    { altezza = x; }
45:
46:         // metodi get          per conoscere il valore degli attributi
47:         float getLunghezza(float x) { return lunghezza; }
48:         float getLarghezza(float x) { return larghezza; }
49:         float getAltezza(float x)   { return altezza; }
50:
51:
52:         // altri metodi
53:         float supDiBase( ) {
54:             return lunghezza * larghezza;
55:         }
56:
57:         float supLaterale( ) {
58:             return ((lunghezza + larghezza) * 2 * altezza);
59:         }
60:

```

```

61:     float supTotale( ) {
62:         return supDiBase( ) * 2 + supLaterale( );
63:     }
64:
65:     float volume( ) {
66:         return supDiBase( ) * altezza;
67:     }
68:
69:     bool isCubo( ) {
70:         return ((lunghezza == larghezza) && (larghezza == altezza));
71:     }
72:
73:     void comunicaDati( ) {
74:         cout << "\n-----";
75:         cout << "\nParallelepipedo di dimensioni: "<<lunghezza
76:             <<" x "<< larghezza<<" x "<<altezza<<endl;
77:         cout << "\n La superficie totale e': "<< supTotale( ) << endl;
78:         cout << "\n Il volume e': "<< volume() << endl;
79:         if (isCubo( ))
80:             cout<<"\n E' anche un CUBO"<<endl;
81:         else
82:             cout<<"\n NON e' un CUBO\n"<<endl;
83:     }
84: };
85:

```