

```
1  /*-----
2  EinfacheKlasseQuader.cpp
3  -----
4  ITA-3. Semester PTP
5  -----*/
6 #include "stdafx.h"
7 #include <iostream>
8 #include <string>
9 #include <iomanip>
10 using namespace std;
11
12 class Quader {
13 private:
14     int Breite;
15     int Hoehe;
16     int Tiefe;
17 public:
18     Quader(int b, int h, int t);
19     ~Quader();
20
21     void setBreite(int b);
22     void setHoehe(int h);
23     void setTiefe(int h);
24
25     int getBreite();
26     int getHoehe();
27     int getTiefe();
28
29     void zeichne();
30     int getVolumen();
31     bool istWuerfel();
32 };
33
34 Quader::Quader(int b, int h, int t) {
35     Breite= b;
36     Hoehe = h;
37     Tiefe = t;
38 }
39 Quader::~Quader() {
40     cout << "Quader (" << Breite << ", " << Hoehe << ", " << Tiefe << ") wird      ↵
41     geloescht." << endl;
42 }
43 void Quader::setBreite(int b) {
44     Breite = b;
45 }
46
47 void Quader::setHoehe(int h) {
48     Hoehe = h;
49 }
50
51 void Quader::setTiefe(int t) {
52     Tiefe = t;
53 }
54
55 int Quader::getBreite() {
```

```
56     return Breite;
57 }
58
59 int Quader::getHoehe() {
60     return Hoehe;
61 }
62
63 int Quader::getTiefe() {
64     return Tiefe;
65 }
66
67 void Quader::zeichne() {
68     cout << "Front: " << setw(Breite*2 - 2) << " " << "Seite: "
69     << setw(Tiefe*2 - 2) << " " << "Oben:" << endl;
70     int h;
71     for (h=0; h<Hoehe; h++) {
72         // Frontansicht:
73         for (int b=0; b<Breite; b++) {
74             cout << "* ";
75         }
76         cout << "      ";
77         // Seitenansicht:
78         for (int t=0; t<Tiefe; t++) {
79             cout << "* ";
80         }
81         cout << "      ";
82         if (h<=Breite) {
83             // Obenansicht gleich miterledigen:
84             for (int t=0; t<Tiefe; t++) {
85                 cout << "* ";
86             }
87             cout << endl;
88         }
89     }
90     if (Breite>h) {
91         // restliche Obenansicht erledigen:
92         cout << setw((Breite+Tiefe+5)*2) << " ";
93         for (int t=0; t<Tiefe; t++) {
94             cout << "* ";
95         }
96         cout << endl;
97     }
98 }
99
100 int Quader::getVolumen() {
101     return Breite*Hoehe*Tiefe;
102 }
103
104 bool Quader::istWuerfel() {
105     return (Breite==Hoehe && Breite==Tiefe);
106 }
107
108 int main() {
109     // 1.: Quader erzeugen und mit Kantenlängen 4, 3 und 5 initialisieren (per Konstruktor)
110     // dazu Maße des Quaders anzeigen und Quader zeichnen,
```

```
111     // berechnetes Volumen anzeigen und auf Würfel prüfen
112     Quader Q1(4,3,5);
113     cout << "Abmessungen (b=" << Q1.getBreite();
114     cout << ", h=" << Q1.getHoehe() << ", t=" << Q1.getTiefe() << "): " << endl << ↵
115     endl;
116     Q1.zeichne();
117     cout << endl;
118     cout << "Volumen = " << Q1.getVolumen() << endl;
119     cout << "Ist Wuerfel: " << (Q1.istWuerfel() ? "ja" : "nein") << endl;
120     cout << endl;
121     Q1.setBreite(5);
122     Q1.setHoehe(5);
123     cout << "Abmessungen (b=" << Q1.getBreite();
124     cout << ", h=" << Q1.getHoehe() << ", t=" << Q1.getTiefe() << "): " << endl << ↵
125     endl;
126     Q1.zeichne();
127     cout << endl;
128     cout << "Volumen = " << Q1.getVolumen() << endl;
129     cout << "Ist Wuerfel: " << (Q1.istWuerfel() ? "ja" : "nein") << endl;
130     cout << endl;
131     system("pause");
132     return 0;
133 }
```