1. Code-Kommentare

Betrachten Sie den folgenden Code-Ausschnitt aus einem C-Programm.

```
// [1] Dieser Code summiert die ersten 100 Elemente des Array arr[] auf
// und schickt die einzelnen Elemente über ein Netzwerkverbindung an
// einen Kommunikationspartner. Bei Position 17 findet sich immer ein
// falscher Wert, der ignoriert (auch nicht versendet) wird.

double    float arr[500];          // [2] double floats für hohe Genauigkeit
int       i;                       // [3] i ist ein Integer
int       sum;
const int DEFAULT_VAL = 15;        // [4] In jahrelangen Messungen hat sich 15 als sinnvoll erwiesen

// [5] i durchläuft eine Schleife von 0 bis 99
sum = 0;
for (i=0; i<100; i++) {
  if (i!=17) {
    sum += arr[i];                 // [6] addiere arr[i] zu sum
    send_to_partner (i, arr[i]);   // [7] der Partner wurde bei der Initialisierung festgelegt
  }
  else
    sum += DEFAULT_VAL;            // [8] bei i=17 immer falscher Wert im Array
}

// [9] bei zu großer Summe droht Zerstörung der Datenbank!
if (sum > 599)
  fail_and_exit (ERR_TOOBIG);      // [10] Fehlercode ist in localerrs.h definiert
```

Es gibt hier diverse Kommentare, die mit [1] bis [10] durchnummeriert wurden. Ordnen Sie jedem Kommentar eine Kommentar-Kategorie (Wiederholung, Erklärung etc. nach McConnell) zu und geben Sie an, welche Kommentare entfernt oder verändert (wie?) werden sollten.

2. Selbst-dokumentierender Code

Betrachten Sie das folgende Programm `TachoTest.java` (das auch über die Webseite erreichbar ist). Finden Sie zunächst heraus, welchen Zweck die einzelnen Methoden haben und wie sie implementiert sind. Bewerten Sie dann, inwiefern es sich hier um selbst-dokumentierenden Code handelt.

Ändern Sie das Programm so ab, dass es dem Ideal des selbst-dokumentierenden Code möglichst nah kommt.

```
/**
 * Java-Programm, das einen Fahrrad-Computer (Tacho) implementiert
 **/

/*
 * Version 1.0, Hans-Georg Esser, fom.hgesser.de
 * Software Engineering, SS 2015
 */

class Tacho
{
    int[] v;    // Messwerte
    int   n;    // max. Anzahl der Messwerte
    int   mx;   // Maximum
    float av;   // Durchschnittswert
    int   cnt;  // Anzahl der bisher erfolgten Messungen

    /** Konstruktor **/
    public Tacho(int sz)
    {
        n = sz;
        v = new int[sz];
        mx = 0; av = 0; cnt = 0;
    }
```

```
    /** Neue Messung einfuegen **/
    public boolean add(int m)
    {
        if (cnt < n)
        {
            v[cnt] = m;
            if (m > mx)  mx = m;
            av = (cnt * av + m) / (cnt+1);
            cnt++;
            return true;
        }
        else
            return false;
    }

    /** Tacho zuruecksetzen **/
    public void setZero()
    {
        cnt = 0;
    }

    /** Maximum neu berechnen **/
    private void updateMx()
    {
        mx = 0;
        for (int i = 0; i < cnt; i++)
            if (v[i] > mx)  mx = v[i];
    }

    /** Letzten Eintrag loeschen **/
    public void undo()
    {
        if (cnt > 1)
        {
            av = (cnt * av - v[cnt-1]) / (cnt-1);
            cnt--;
            updateMx();
        }
        else if (cnt == 1)
        {
            av = 0; mx = 0;
            cnt--;
        }
    }

    /** Alle Messungen anzeigen **/
    public void show()
    {
        System.out.print("\n" + cnt + " Tacho-Messungen: ");
        for (int i = 0; i < cnt; i++)
            System.out.print(v[i] +" ");
        System.out.println("\nDurchschnitt: " + av);
        System.out.println("Maximum: " + mx);
    }

    /** Statistik der letzten N Messungen **/
    public void showN(int z)
    {
        if (z>cnt)
        {
            System.out.println ("Fehler: Es gibt nur " + z + " Messwerte.");
            return;
        }
        float v1 = 0;   // Summe, Durchschnitt
        int   v2 = 0;   // Maximum

        for (int i = cnt-z; i < cnt; i++) {
            v1 += v[i];
            if (v[i] > v2)  v2 = v[i];
        }
        v1 = v1 / z;   // Durchschnitt
        System.out.println("Für letzte " + z + " Messungen:");
        System.out.println("Durchschnitt: " + v1);
        System.out.println("Maximum: " + v2);
    }
}
```

```
/** Class TachoTest **/
public class TachoTest
{
    public static void main(String[] args)
    {
        Tacho t = new Tacho(100);
        // 1. Test
        t.add(19); t.add(13); t.add(100);
        t.undo();
        t.show ();
        // 2. Test
        t.setZero();
        t.add(20); t.add(21); t.add(22); t.add(23); t.add(134);
        t.undo(); t.undo();
        t.add(19);
        t.show(); t.showN(3); t.showN(1);
    }
}
```