Das Problem war einfach: Eine LED soll genau 5 mal blinken und dann aus bleiben.
Folgender Sketch bot sich an:

```
int n=0;      // Zaehlervariable n

void setup() {
 pinMode(13, OUTPUT);   //LED an Pin 13
}

void loop() {
  if (n<=5)   {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  }
n=n+1;          // Hochzaehlen
delay(1);       // Trick: Loop verlangsamen
}
```

n ist unsere Zählvariable die als Integer-Zahl beim Start auf 0 gesetzt wird. Durch die Zeile n=n+1; wird diese Variable bei jedem Loop hinter der if-Abfrage um eins erhöht. Zunächst wird der Trick mit dem „delay 1“ noch nicht eingefügt.
Ergebnis des Sketches: Die Diode blinkt fortlaufend. Das hätten wir nicht erwartet. Irgendwann wird n=6 und ab dann sollten die Anweisungen in der if-Bedingung nicht mehr ausgeführt werden und die LED aus bleiben. Was ist passiert? Klar, ab n=6 wird die Anweisungen der if-Abfrage nicht mehr ausgeführt. Damit rast der Loop aber gigantisch schnell (max. mit der Taktfrequenz des Arduino) und ruck-zuck wird die Bedingung in der if-Abfrage wieder erfüllt. Aber warum?

Dazu muss man wissen, dass für eine Integer-Zahl im Speicher des Arduino ein Platz von 2 Byte=16 Bit reserviert wird. Für eine 16-Bit-Zahl gibt es 2^16 = 65.536 verschiedene Möglichkeiten. Damit auch negative Zahlen verwendet werden können, teilt man diese von –32768 bis +32768 auf. Unsere Integer-Zahl darf also nur in diesem Bereich liegen.

Um genauer zu sehen, was wirklich passiert, machen wir einen Trick: Wir verlangsamen den Loop durch ein Delay von 1ms.

Dann lassen wir den veränderten Sketch wieder laufen. Jetzt wird es noch merkwürdiger. Nach 5 mal Blinken hört die LED wirklich auf, ist also alles o.k.? Aber nach ca. 32 Sekunden fängt sie wieder an zu blinken. Wieso denn das?
Nun, nach 5 Sekunden Blinkzeit läuft der Loop ja weiter. Er benötigt jetzt jedoch ca. 1ms für jeden Durchlauf (unser Delay….). Nach (32763)*1ms also etwa 32,7 Sekunden wird der Bereich der Integer-Zahl überschritten. Es erfolgt ein „Overflow“, d.h. n springt von +32768 auf –32768. Damit wird die if-Bedingung wieder erfüllt (da -32768<5 ist) und die LED blinkt von neuem.
Überlege selbst, wie lange diese jetzt blinken wird, bevor sie wieder für ganze 32 Sekunden aussetzt.
Ist doch verrückt? Oder?

Der richtige Sketch sieht so aus: (Warum?)
…..nur eine **kleine** Änderung und schon läuft´s wie geplant.

```
int n=0;

void setup() {
  pinMode(13, OUTPUT);
}
void loop() {
  if (n<=5) {
  digitalWrite(13, HIGH);
  delay(500);
  digitalWrite(13, LOW);
  delay(500);
  n=n+1;
  }
}
```

Blinkende LED mit Hilfe einer for-Schleife:

```
void setup(){
pinMode(13, OUTPUT);
}
void loop() {
  for(int n=1; n<=6; n++){
    digitalWrite(13, HIGH);
    delay(500);
    digitalWrite(13, LOW);
    delay(500);
    }
}
```

Begründe, warum auch dieser Sketch falsch läuft!