# SOFTWARE ENGINEERING 2

## BLACKBOX-TESTVERFAHREN

# Aufgabe 1

Für eine Restweitenanzeige bei einem Pedelec wird der Ladezustand (State of Charge – SOC) mit fünf Balken angezeigt. Der SOC liegt zwischen 0 und 100%. Liegt der SOC zwischen 80 und 100%, so werden alle fünf Balken angezeigt. Entsprechendes gilt in 20%-Schritte für den restlichen Bereich. Liegt der SOC unter 10%, so soll der letzte Balken blinken.

Des Weiteren ist die Anzeige temperaturabhängig. Sinkt die Temperatur unter 5°C, so sind vom SOC Wert 5% abzuziehen, bei Frost 10%.

Die Funktion bekommt SOC und Temperatur als `sint` übergeben.
Der Rückgabewert ist die Anzahl der anzuzeigenden Balken, wobei der Wert 10 für einen einzelnen blinkenden Balken steht.

Entwerfen Sie Testfälle.
An welchen Stellen muss die Spezifikation präzisiert werden?

# Äquivalenzklassentests

- Zerlegung der möglichen Wertebereiche der Eingangsgrößen in Äquivalenzklassen (gleiches/ähnliches Verhalten)
  Hier: ÄK3: SOC = [20%...40%[
- Jede Äquivalenzklasse wird durch eine Belegung repräsentiert (Repräsentant der Äquivalenzklasse)
  Hier: ÄK3 → 35%
- Zu den Äquivalenzklassen gehören auch ungültige/nicht zugelassene Wertebereiche
  Hier: ÄK_UNG1 = ]- ∞ …0%[
- Bei konkreten Testfällen muss der implementierte Wertebereich beachtet werden.
  Hier: ÄK_UNG1 = [INT_MIN…-1]
- Achtung: Enumerations werden in C/C++ auf int abgebildet.
  Daher können solche Variablen auch Werte außerhalb des Definitionsbereichs annehmen.

Regeln für das Erstellen der Tests

1. Die Repräsentanten aller gültigen Klassen sind zu kombinieren.
2. Ein Repräsentant einer ungültigen Klasse ist nur mit Repräsentanten gültiger Klassen zu kombinieren (warum?).
3. Reduktion durch:
   1. Häufigkeit des Auftretens/Risiko
   2. Bevorzugung von Grenzwerten
   3. Paarweise Kombination statt vollständiger Kombination
4. Minimalanforderung: Jeder Repräsentant muss in mindestens einem Testfall auftreten.

- Äquivalenzklassenüberdeckung (ÄK-Überdeckung)

$$\text{ÄKÜ} = \frac{\#\text{getestete ÄK}}{\text{ÄK}}$$

- Auswahl der ÄK ist entscheidend
- Qualität der Anforderungsdokumente (ggf. „nachschärfen“)

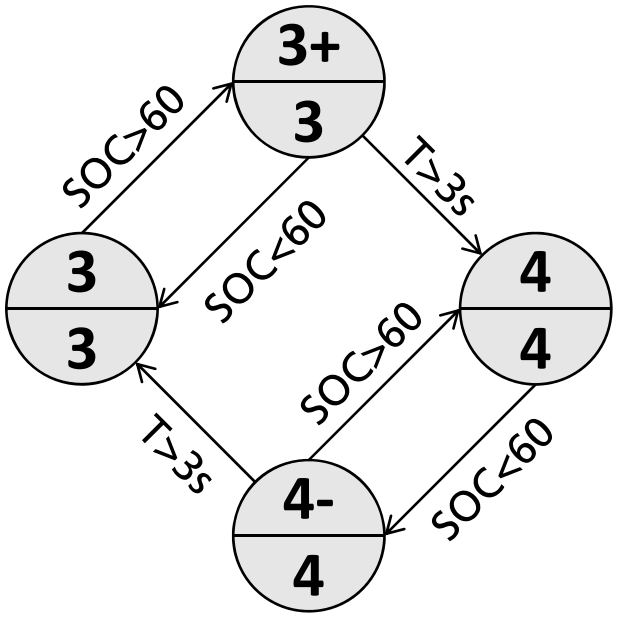
# Aufgabe 2

Damit die Anzeige nicht ständig springt, wird eine Entprellung eingeführt. Der nebenstehende Automat beschreibt die exemplarisch für den Übergang bei 60% (Moore-Automat; Ausgabe: Anzahl der angezeigten Balken).

Die Funktion wird einmal pro Sekunde aufgerufen.

Entwerfen Sie Testfälle, die das Verhalten „um 60%“ testen.

An welchen Stellen muss die Spezifikation präzisiert werden?



# Grenzwerttests

- Testverfahren, bei dem die Wertebelegung nahe an den Grenzen der Äquivalenzklassen liegen.
- Überprüfung, ob das Verhalten an den Grenzen richtig implementiert ist.
  Typische Fehlerzustände, die so ermittelt werden können:
  Abfrage auf  < anstelle von <= o.ä.
- Grenzwerttests bei numerischen Werten
  - Nahe an den Grenzen
  - Weit von den Grenzen entfernt
- Grenzwerttests bei Datenstrukturen
  - Leere Struktur
  - Voll belegte Struktur
- Grenzwerttests bei Feldern (Arrays)
  - Zugriff auf das erste und das letzte Element
  - Zugriff auf das zweite und das vorletzte Element
  - Zugriff auf das Element nach/vor dem letzten/ersten Element (off by one)