

# Aufgabenstellung

- Im Rahmen der Aufgabe ist eine digitale Stopp- und Eieruhr zu modellieren, deren Funktionalität auf den nachfolgenden Seiten näher beschrieben wird.
- Die Modellierung besteht aus 2 Teilen:
  - Erstellen eines aussagekräftigen UML-Zustandsdiagramms
  - Anschließende Implementierung des Zustandsautomaten in C++

# Beschreibung 1 / 2

- Die Anzeige ist vierstellig. Die ersten beiden Stellen zeigen die Minuten (00 bis 99) an, die hinteren beiden Stellen die Sekunden (00 bis 59).
- Bedient wird die Uhr über zwei Umschalttasten ...
  - Stopp/Weiter
  - Messen/Stellen
- ...sowie zwei weitere Tasten
  - Plus
  - Minus

## Beschreibung 2 / 3

- Zu Beginn zeigt die Uhr 0000 (also 0 Minuten und 0 Sekunden) an und befindet sich im Stellmodus.
- Im Stellmodus kann die gerade angezeigte Zeit mit den Tasten Plus und Minus minutenweise erhöht bzw. erniedrigt werden.
- Drückt man im Stellmodus die Taste Messen/Stellen, dann wechselt die Uhr entweder in den Stoppuhrmodus (falls die Stellzeit 0000 ist) oder in den Eieruhrmodus (ansonsten)
- Von beiden Modi führt erneutes Drücken der Taste Messen/Stellen zurück in den Stellmodus, wo die zuletzt eingestellte Stellzeit wieder angezeigt wird.

## Beschreibung 3 / 3

- Im Stoppuhrmodus wird die Zeitanzeige jede Sekunde um eine Sekunde erhöht. Die Stoppuhr läuft im Prinzip unbegrenzt lange, wobei auf 9959 die Zeitanzeige 0000 folgt. Die Zeitmessung kann mit der Taste StoppWeiter unterbrochen und durch erneutes Drücken dieser Taste wieder aufgenommen werden.
- Im Eieruhrmodus wird ausgehend von der eingestellten Zeit die Zeitanzeige jede Sekunde um eine Sekunde verringert. Beim Erreichen von 0000 endet die Zeitmessung: die Uhr läutet eine Minute lang (sofern nicht auf StellenMessen gedrückt wird) und wechselt dann automatisch in den Stellmodus, wo wieder die zuletzt eingestellte Stellzeit angezeigt wird. Wie beim Stoppuhrmodus kann die Zeitmessung mit der Taste StoppWeiter unterbrochen bzw. wieder aufgenommen werden.

# Individuelle Hilfestellung

- Die folgenden globalen Funktionen können bei Bedarf vom Fachlehrer bereitgestellt werden:
- `void zeige(unsigned short minuten, unsigned short sekunden)`
  - zeigt im Konsolenfenster ein vereinfachtes Uhrendisplay mit den übergebenen Werten für Minuten und Sekunden an
- `char holeEvent()`
  - liest den Tastaturpuffer und liefert das einer gedrückten Taste zugeordnete Zeichen zurück (bzw. 0, falls keine Taste gedrückt wurde)
- `void warte(unsigned int msekunden)`
  - realisiert eine Warteschleife, die erst nach *msekunden* Millisekunden wieder verlassen wird