# Fehlertoleranz in eingebetteten Systemen

Nebenläufige Programmierung, WS 18/19

Georg Ringwelski

# Agenda

- Begriffsklärung und Anforderungsspezifikation
- Methoden zur Fehlertoleranz
- Best Practices

# Zuverlässigkeit eingebetteter Systeme

Brainstorming: Was ist Softwarezuverlässigkeit?

# Definition Softwarezuverlässigkeit

- Lt. ISO 25010 ist **Softwarezuverlässigkeit** ein Qualitätsmerkmal eines Softwareprodukts
- SW-Zuverlässigkeit := die Fähigkeit von SW, ihre Funktion und Leistung über einen festgelegten Zeitraum unter definierten Bedingungen aufrecht zu erhalten
- Dazu gehört (Zitat):
  - **Fehlertoleranz**: Fähigkeit, ein spezifiziertes Leistungsniveau bei Software-Fehlern oder Nicht-Einhaltung ihrer spezifizierten Schnittstelle zu bewahren.
  - **Konformität**: Grad, in dem die Software Normen oder Vereinbarungen zur Zuverlässigkeit erfüllt.
  - **Reife**: Geringe Versagenshäufigkeit durch Fehlerzustände.
  - **Wiederherstellbarkeit**: Fähigkeit, bei einem Versagen das Leistungsniveau wiederherzustellen und die direkt betroffenen Daten wiederzugewinnen. Zu berücksichtigen sind die dafür benötigte Zeit und der benötigte Aufwand.

# Verallgemeinerung: Zuverlässigkeit eingebetteter Systeme

- Def: Die **Zuverlässigkeit** eines technischen Systems ist eine Eigenschaft, die angibt, wie verlässlich eine dem System zugewiesene Funktion in einem Zeitintervall erfüllt wird.
- Messung i.d.R empirisch
- In Eingebetteten Systemen ist die Zuverlässigkeit eine Produkt aus der Zuverlässigkeit der Software und der Zuverlässigkeit der Hardware

# Und was bedeutet das für uns?

- Die Zuverlässigkeit der Hardware können wir mit Lego nicht beeinflussen.
  - Messfehler der Sensoren
  - Ungenaue Mechanik bei Lenkung
  - Ungenaue Realisierung von Steuersignalen der Aktoren (zB Motoren)
  - ..
- Wir erhöhen deswegen die Zuverlässigkeit des Systems durch **Fehlertoleranz** der Software gegen Fehler/Ungenauigkeiten der Hardware
  1. Fehlerquellen identifizieren
  2. Fehlerwahrscheinlichkeit ermitteln
  3. Auswirkungen von Fehlern ermitteln
  4. Geeignete Fehlertoleranz implementieren

# Agenda

- Begriffsklärung und Anforderungsspezifikation
- Methoden zur Fehlertoleranz
- Best Practices

# Wiederholung

- Wie ist die Zuverlässigkeit technischer Systeme definiert?
- Wie wird sie gemessen?
- Welche Faktoren können die Zuverlässigkeit eingebetteter Systeme negativ beeinflussen?
- Was versteht man unter Fehlertoleranz?

# Methoden zur Fehlertoleranz

1. Fehlerquellen identifizieren
2. Fehlerwahrscheinlichkeit und -größe ermitteln
3. Auswirkungen von Fehlern ermitteln
4. Geeignete Fehlertoleranz implementieren

# Anforderungen an Fehlertoleranz

Fallstudie in Kleingruppen (15 min)

Wir bauen ein Fahrzeug für Robosoccer mit u.g. Sensoren bzw. Aktoren des EV3

Betrachten Sie eines der drei Bauteile und beschreiben Sie:

1. Wie sie das Bauteil einsetzen würden
2. Welche Fehler das Element dabei machen könnte und welche Auswirkungen das auf das Verhalten des Systems haben könnte
3. Wie Sie die Fehlerwahrscheinlichkeit und –größe ermitteln würden

- Gruppe1: EV3LargeRegulatedMotor
- Gruppe2: EV3UltrasonicSensor
- Gruppe3: EV3ColorSensor

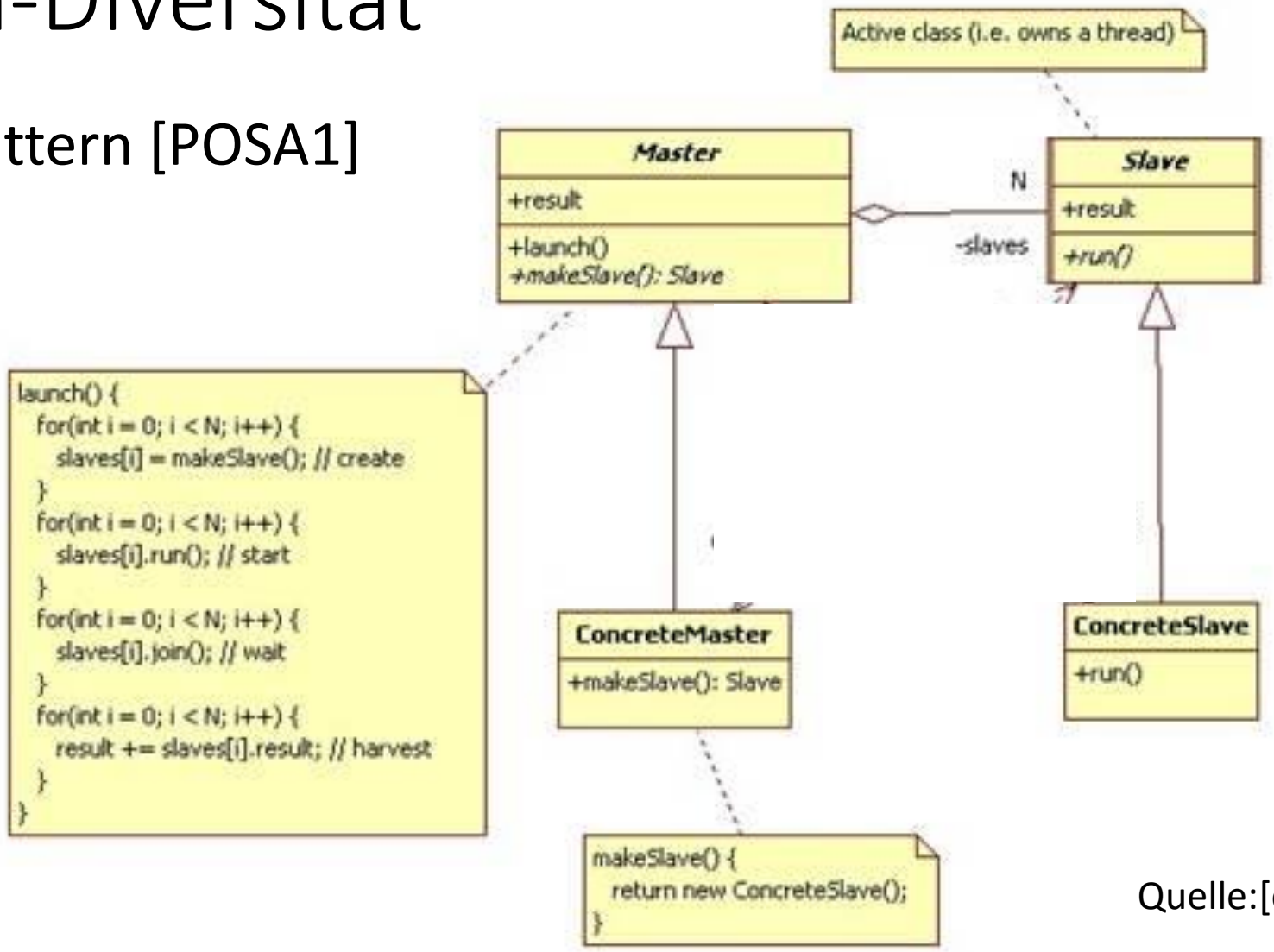
Präsentieren Sie Ihre Ergebnisse im Plenum mit drei Folien (zu jeder Frage eine)

# Implementierung von Fehlertoleranz

- Implementierung von Fehlertoleranz ist notwendig,
  - wenn Fehler negative Auswirkungen auf die Zuverlässigkeit haben
  - wenn das Fehlerrisiko (Wahrscheinlichkeit * Kosten) den Aufwand der Implementierung rechtfertigt
- Methoden der Implementierung
  1. Design-Diversität: verschiedene Implementierungen eines Algorithmus laufen parallel
  2. Daten-Diversität: die Eingabedaten werden leicht modifiziert mehrfach bearbeitet
  3. Temporale Diversität: ein Algorithmus wird mehrfach aufgerufen

# Implementierung von Fehlertoleranz: 1. Design-Diversität

Master-Slave Pattern [POSA1]




Quelle:[cs.sjsu.edu]

# Implementierung von Fehlertoleranz: 1. Design-Diversität

Diskussion im Plenum

Wie kann das Master-Slave-Pattern eingesetzt werden, um die Fehlertoleranz ggü Messfehlern von Sensoren zu verbessern?

- Woher bekommt man unterschiedliche Slave-Implementierungen?
- Das Ergebnis welches/r Slaves wird am Ende umgesetzt?

# Implementierung von Fehlertoleranz: 2. Daten-Diversität

Eingabedaten werden leicht modifiziert mehrfach bearbeitet

- Mögliche Modifizierungen
  - Rundung von Messwerten
  - Anpassung von Messwerten nach Plausibilität
  - Ignorieren von unplausiblen Messwerten
- Dabei gibt es unterschiedliche Abstufungen
  - Rundung auf wie viele Stellen?
  - Was ist plausibel?
  - Wie werden unplausible Werte verändert?

# Implementierung von Fehlertoleranz: 2. Daten-Diversität

Diskussion im Plenum

Wie können wir Daten-Diversität einsetzen, um Messfehler von Sensoren in mobilen Robotern auszugleichen?

# Implementierung von Fehlertoleranz: 3. Temporale Diversität

- Bei regelmäßigen Messungen entstehen zeitliche Sequenzen von Messwerten
- Messwerte können aus mehreren solchen Messungen berechnet werden (zB Durchschnitt, oder besser Median)
- Die Plausibilitätsentscheidung in der Daten-Diversität kann aus diesen Messreihen abgeleitet werden.

# Implementierung von Fehlertoleranz: 3. Temporale Diversität

Diskussion im Plenum

Wie können wir temporale Diversität einsetzen, um Messfehler von Sensoren in mobilen Robotern auszugleichen?

# Agenda

- Begriffsklärung und Anforderungsspezifikation
- Methoden zur Fehlertoleranz
- Best Practices

# Best Practices

Hardware und Software funktionieren nicht immer so wie sie spezifiziert ist!

- Untersuchen Sie die Zuverlässigkeit der Komponenten.
- Identifizieren Sie Fehlerquellen und beseitigen sie, wenn möglich.
- Bei bleibenden Fehlerquellen beurteilen Sie Fehlerwahrscheinlichkeit, Risiko und Aufwand zur Implementierung von Fehlertoleranz
- Implementieren Sie ggf. Fehlertoleranz durch Design-Diversität, Daten-Diversität oder temporale Diversität