

Übung zur Vorlesung Modellierung Nebenläufiger Prozesse, SoSe 25

Übungsblatt Nr. Projekt - 02, **Abgabe am: 06.07.2025 — 23:59**

Die Ausarbeitung der Übung bitte bis zum angegebenen Termin per Moodle abgeben.

23.06.2025

Die folgenden Modalitäten gelten für das Projekt:

- Um das folgende Projekt zu bestehen, muss Ihr Code genau das in der Aufgabe beschriebene Verhalten zeigen.
- Code der nicht kompiliert führt zum automatischen Nichtbestehen.
- Die Aufgaben sollen in Java mittels der Akka Actor Typed-Bibliothek in Version 2.9 oder höher bearbeitet werden.
- Nutzen Sie keine andere Bibliotheken als die Akka Actor Typed-Bibliothek und die Standardbibliothek.
- Sollte ihr Code auf Methoden des Classic Akka (nicht getypte Aktoren) aufbauen, führt dies zum Nichtbestehen. Ein Beispiel, Daten werden ausschließlich über Nachrichten ausgetauscht.
- Versehen Sie ihre Aufgabe mit sinnvollen Kommentaren.
- Sie dürfen in Gruppen bis zu 4 Studierenden abgeben. Notieren Sie Name und Matrikelnummer von allen Studierenden Ihrer Gruppe auf allen Javodateien.
- Ihr Code wird auf Plagiate getestet, fällt ein solches Plagiat auf, führt dies ebenfalls zum automatischen Nichtbestehen.
- Sollten Sie in einer größeren Gruppe als zu viert arbeiten, machen sie dies bitte im Code deutlich. Sonst wird dies als ein Plagiat gewertet und führt zum Nichtbestehen.
- Die Abgabe erfolgt als zip-Datei des kompilierbaren Projekts (Gradle!) über das Moodle. Es muss nur einer aus Ihrer Gruppe den Quellcode abgeben.
- Sollten Sie in der Bearbeitungszeit durch eine Krankheit am Arbeiten behindert werden, besorgen Sie sich bitte ein Attest für diesen Zeitraum. Dann wird die Bearbeitungszeit entsprechend verlängert.

Aufgabe 1 (Der Zulieferer)

(0 Punkte)

Der Autobauer MeinAutobauer hat einen Zulieferer für elektronische Bauteile EB gekauft um die eigene Produktivität zu steigern. Ein elektronisches Bauteil wird durch die Kombination von zwei Bausteinen gebaut. Ein Baustein kann dabei einer von vier Basisbausteinen BB sein, oder ein anderes elektronisches Bauteil.

Die elektronischen Bauteile haben folgende Baupläne:

- $EB_1 = (BB_1, BB_2)$
- $EB_2 = (EB_1, BB_2)$
- $EB_3 = (BB_3, EB_2)$
- $EB_4 = (EB_1, EB_3)$

Wenn ein elektronisches Bauteil produziert wird, wird für die Produktion jeder benötigten Komponente ein eigener Prozess gestartet (Aktor erstellt), so dass am Ende alles parallel produziert wird. Als Beispiel, wenn EB_4 produziert wird, starten Prozesse für EB_1 und EB_3 . Bei EB_3 startet einer für EB_2 und dort beginnt einer für EB_1 .

Aufgaben

- Implementieren Sie den Ansatz zum Bauen der elektronischen Bauteile, wobei für jeden verbauten Baustein und für jedes fixes Bauteil ein eigener Prozess gestartet wird. Beachten Sie, dass die Reihenfolge der Baupläne eingehalten werden muss.
- Geben Sie aus, wenn ein neues Bauteil gebaut wird, so dass in jedem Schritt ersichtlich ist was geschieht.
- Testen Sie die Ausgabe der Produktion, indem sie EB_4 produzieren.
- Erstellen Sie ein neues elektronisches Bauteil, aus den gegebenen Komponenten, dass wenigstens eine Tiefe von drei hat.

Einige Hinweise:

- Es ist integraler Bestandteil der Aufgabe, dass die Produktion der Komponenten parallel abläuft. Für die Produktion einer Komponente wird immer ein eigener Aktor gestartet.
- Die Baupläne bestehen immer aus zwei Komponenten und für den Bau jeder Komponente wird ein eigener parallel laufender Prozess gestartet. Das hinterlegen eines kompletten Aufrufs ist nicht korrekt und führt zum Nichtbestehen, als Beispiel $EB_2 = ((BB_1, BB_2), BT2)$.
- Die Baupläne für die einzelnen elektronischen Bauteile können Sie natürlich im Code hinterlegen.
- Es ist ausreichend, die Fertigstellung einer Komponente durch eine Lognachricht zu zeigen, nachdem diese produziert wurde. Beachten Sie die Reihenfolge, wenn Sie z.B. Baustein 2 bauen muss zuerst Baustein 1 gebaut werden. Ist die Reihenfolge nicht korrekt, führt dies zum Nichtbestehen des Projekts. '
- Sie werden für die Aufgabe dynamisch Aktoren instanziiieren müssen. Da alle Aktoren im System einen eindeutigen und unterschiedlichen Namen haben müssen, können Sie nicht mit `AkkaContext.spawn` erstellt werden. Sie können aber `AkkaContext.spawnAnonymous` verwenden.