

## Reussner Softwaretechnik 2

Dauer: 60 min  
Bemerkungen: -

Lösung: keine

Bestanden mit: ?/60 P

### Aufgabe 1: Wissensfragen (7 Punkte)

- a) Sollten Sie die Kopplung (*coupling*) zwischen Klassen in unterschiedlichen Paketen erhöhen oder reduzieren? Begründen Sie Ihre Antwort kurz. (1 Punkt)
- b) Nennen Sie die vier Einflussfaktoren, die die Leistung (*performance*) einer Softwarekomponente beeinflussen? (2 Punkte)
- c) Nennen und erläutern Sie zwei der drei Merkmale (*features*) eines Modells nach Stachowiak's Allgemeiner Modelltheorie. (2 Punkte)
- d) Nennen und erläutern Sie kurz zwei der fünf SOLID-Prinzipien. (2 Punkte)

### Aufgabe 2: Anwendungsfallbeschreibung (16 Punkte)

Folgende Beschreibung charakterisiert die zugrundeliegende Funktionalität eines Einreichungssystems für wissenschaftliche Beiträge:

*Das System erlaubt es einem Autor (author), eine Liste von Konferenzen anzuzeigen sowie einen Artikel (paper) auf einer Konferenz einzureichen. Zum Einreichen muss der Autor eine passende Konferenz auswählen, auf der ein Artikel eingereicht werden soll. Neben Autoren unterstützt das System die Organisatoren (organizer). Diese können Gutachter (reviewer) einladen und diesen Artikel zuweisen. Für die Zuweisung muss ein Organisator zuerst den zu begutachtenden Artikel auswählen und danach einen oder mehrere Gutachter auswählen. Gutachtern (reviewer) hingegen ermöglicht das System, die ihnen zugewiesenen Artikel aufzulisten und für diese ein Gutachten abzugeben. Die Abgabe des Gutachtens erfordert zuerst die Auswahl eines zugewiesenen Artikels und anschließend das Lesen des Artikels durch den Gutachter. Nachdem alle Gutachten eingetroffen sind, ermöglicht das System einem Organisator die Entscheidung zu treffen, ob ein eingereichter Artikel angenommen oder abgelehnt werden soll. Dafür wählt er zuerst Artikel aus, sichtet die abgegebenen Gutachten, legt den Status (angenommen oder abgelehnt) des Artikels fest und benachrichtigt abschließend die Autoren. Im Falle von Uneinigkeit unter den Gutachten kann der Organisator den Artikel einem weiteren Gutachter zuweisen oder selbst eine Entscheidung treffen.*

- a) Erstellen Sie das Anwendungsfalldiagramm (*use case diagram*) genau so, dass es die Anforderung im Eingangstext erfüllt. Zeichnen Sie nur Anwendungsfälle ein, die Nutzerziele (*user goals*) darstellen. Achten Sie darauf, an den geeigneten Stellen Beziehungen zwischen den Anwendungsfällen (*use cases*) zu modellieren. Beachten Sie, dass Anwendungsfälle, die Unterfunktionen (*subfunctions*) darstellen, zu Punktabzug führen. (8 Punkte)
- b) Verwenden Sie das textuelle Anwendungsfallbeschreibungsschema *Fully-Dressed Use Case* zur vollständigen Spezifikation des Anwendungsfalls *Entscheidung treffen* (make decision), in dem ein Organisator den eingereichten Artikel entweder annimmt oder ablehnt. Nutzen Sie hierfür die unten gegebene Schablone. Geben Sie auch mindestens eine Erweiterung an. (8 Punkte)

**Name d. Anwendungsfalls**

(Use Case Name)

**Zielebene**

(Goal Level)

**Primäraktor(en)**

(Primary Actor(s))

**Stakeholder(s)**

## Vorbedingungen

(Preconditions)

## Nachbedingungen

(Postconditions)

## Primäres Erfolgsszenario


(Main Success Scenario)

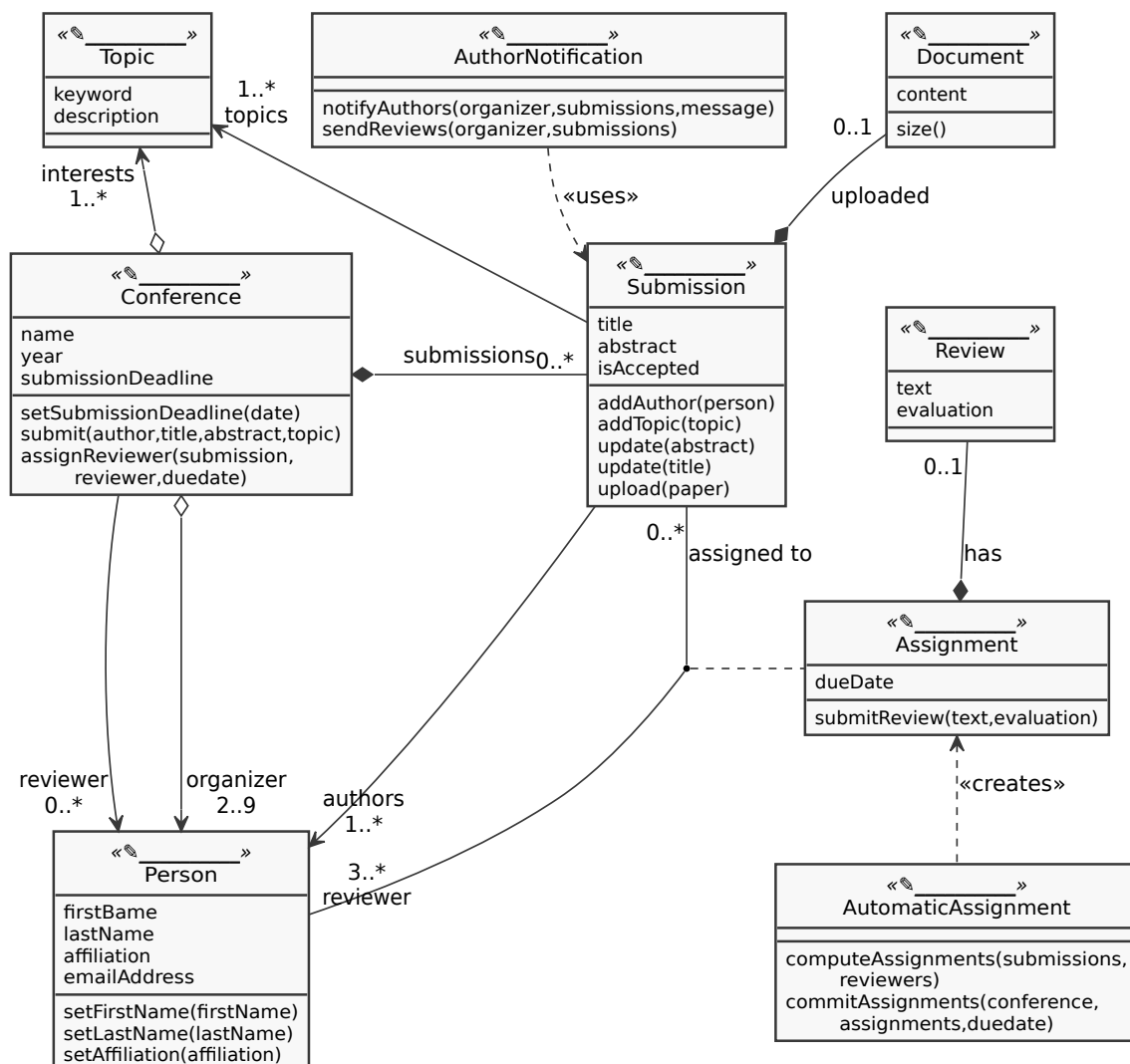
## Erweiterung (Weiterer Ablauf)

(Extension; Additional Flow)

# Aufgabe 3: Domänen-getriebener Entwurf (10 Punkte)

Der folgende Aufgabenkomplex widmet sich der Anwendung des Domänen-getriebenen Entwurfs des Einreichungssystems für wissenschaftliche Beiträge von Konferenzen. Dieses erlaubt es Personen, Beiträge (*submission*) zu einer Konferenz (*conference*) einzureichen. Zu einem Beitrag kann ein Dokument (*document*) hochgeladen werden, wobei das neue Dokument das vorhergehende Dokument ersetzt. Diese Beiträge werden dann Personen zur Begutachtung zugewiesen, welche ein Gutachten (*review*) zu diesen Beiträgen erstellen können. Die Organisatoren haben zusätzlich die Möglichkeit, diese Zuordnung automatisch ermitteln zu lassen sowie, nach erfolgter Begutachtung, die Autoren mehrerer Einreichung zu informieren oder Ihnen die entsprechenden Gutachten zuzusenden.

- a) Durch die Domänenanalyse des Einreichungssystems ist das folgende Klassendiagramm entstanden. Klassifizieren Sie jedes Domänenkonzept entsprechend der Bausteine (*building blocks*) des Domänen-getriebenen Entwurfs entweder als *ValueObject*, *Entity* oder *Service*. Tragen Sie hierzu die Klassifikation in die mit „“ markierten Bereiche der Domänenklassen ein. (4,5 Punkte)

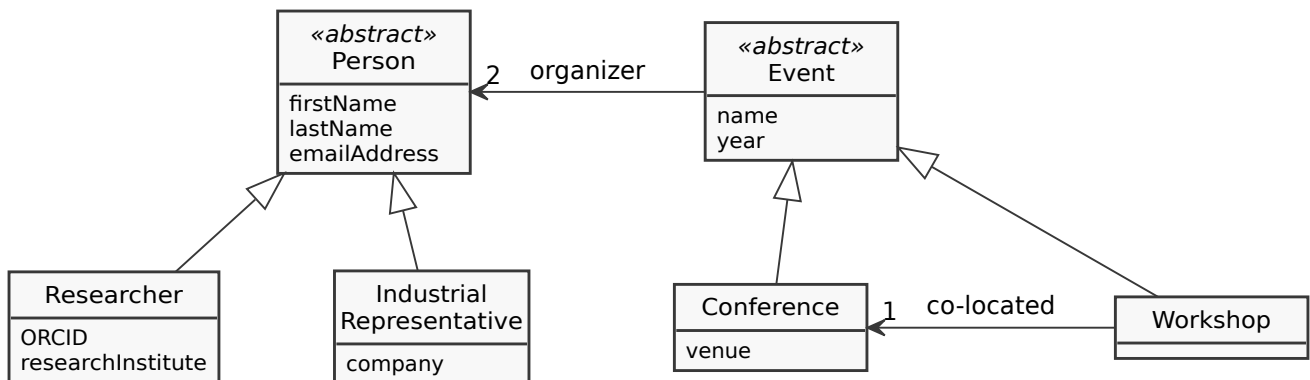


- b) Begründen Sie Ihre Entscheidung für die Klassifikation der Domänenklasse Review in Stichpunkten. (1 Punkt)
- c) Entwerfen Sie nun die Unterstützungsmuster *Repository* und *Factory* für Konferenzen (*conference*). Zeichnen Sie hierfür ein Klassendiagramm, welches ein *Repository*, eine *Factory* sowie ihre Methoden mit Parametern beinhaltet. Spezifizieren Sie für die *Factory* mindestens eine Methode und für das *Repository* mindestens drei Methoden. Zeichnen Sie zusätzlich die Domänenklasse Conference sowie ihre Beziehungen zum *Repository* beziehungsweise zur *Factory* ein. (4,5 Punkte)

**Hinweis:** Datentypen für Parameter (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden. Geben Sie dennoch den Rückgabotyp jeder Methode an. Beschriften Sie jede Assoziation mit mindestens einem Rollennamen oder Assoziationsnamen sowie mindestens einer Kardinalität.

## Aufgabe 4: Entwurfsmuster für Unternehmenssoftware (12 Punkte)

Das Domänenmodell eines Einreichungssystems enthält unter anderem die folgende Klassenhierarchie für Personen und Events. Um diese Klassenhierarchie in einer Datenbank abzubilden, sollen Sie in dieser Aufgabe die unterschiedlichen Objekt-Relationalen Strukturmuster (*object-relational structural patterns*), welche in der Vorlesung behandelt wurden, anwenden und bewerten.



- a) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Single Table Inheritance* Strukturmuster. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem <<table>> Stereotype) und ihrer Attribute. Führen Sie falls nötig neue Attribute ein, um die Vererbung darzustellen. (5 Punkte)

**Hinweis:** Datentypen (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden.

- b) Erstellen Sie ein Datenbankschema nach dem Strukturmuster *Concrete Table Inheritance*. Erstellen Sie hierzu ein UML-Klassendiagramm der nötigen Tabellen (dargestellt als UML-Klasse mit dem <<table>> Stereotype) und ihrer Attribute. Führen Sie falls nötig neue Attribute ein, um Primärschlüssel und Vererbung darzustellen. (6 Punkte)

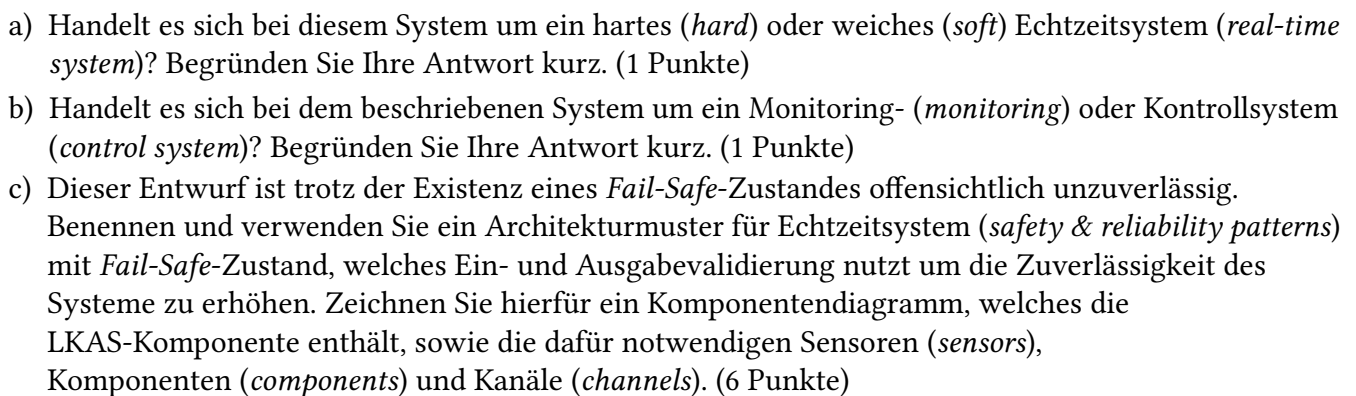
**Hinweis:** Datentypen (wie zum Beispiel int, long, Date, String, ...) müssen nicht angegeben werden.

- c) Geben Sie stichpunktartig einen Vorteil und einen Nachteil der *Concrete Table Inheritance* gegenüber der *Single Table Inheritance* an. (1 Punkte)

## Aufgabe 5: Echtzeitsysteme (15 Punkte)

Ihre Aufgabe ist es, die Zuverlässigkeit des aktiven Spurhalteassistent mit Lenkunterstützung (LKAS) für das Cabriolet *Regalia* der Firma *Lucis* zu bewerten und zu verbessern.

Die Architektur des Systems und Ihre Verteilung im Fahrzeug ist in dem Folgendem Komponentendiagramm dargestellt.



Komponentendiagramm des verbesserten Spurhalteassistent mit Lenkunterstützung:

Gegeben sind die folgenden vier Prozesse mit Angabe ihrer jeweiligen Startzeit (*arrival time*), Dauer (*duration*), und Frist (*deadline*):

**Hinweis:** Die Prozesse kommen zu Beginn des in *Arrival Time* angegebenen Zeitslots an und sollen bis zum Ende des in *Deadline* angegebenen Zeitslots beendet sein. Führen Sie das Scheduling auch dann fort, wenn die *Deadline* nicht eingehalten werden kann.

- E-Mail: [info-klausuren@fsmi.uni-karlsruhe.de](mailto:info-klausuren@fsmi.uni-karlsruhe.de)

$P_n$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$	$t_6$	$t_7$	$t_8$	$t_9$	$t_{10}$	$t_{11}$	$t_{12}$	$t_{13}$	$t_{14}$	$t_{15}$
$P_0$ $lax_0$															
$P_1$ $lax_1$															
$P_2$ $lax_2$															
$P_3$ $lax_3$															