

Name:

Matrikel-Nr.:

Aufgabe 1: Vermischtes (2+3+5=10 Punkte)

Welche Bedeutung hat das Kürzel UML?

- Unified Modeling Language
- Universal Modeling Language
- Unified Meta Language
- Universal Meta Language

Beim Testen kann man z.B. nach dem V-Modell oder dem W-Modell vorgehen. Erläutern Sie die Nachteile des V-Modells und wie diese im W-Modell adressiert werden.

Wann kann die Ausführung einer Aktion eines Aktivitätendiagramms beginnen?
Es sind mehrere Antworten möglich.

- Aktionen können immer ausgeführt werden
- Wenn mindestens eine eingehende Kontrollkante ein Kontrolltoken hat
- Wenn alle eingehenden Kontrollkanten ein Kontrolltoken haben
- Wenn mindestens ein InputPin ein ObjektToken hat
- Wenn alle InputPins ein ObjektToken haben

Name:	Matrikel-Nr.:
-------	---------------

Sie entwickeln ein Softwaresystem, das auf verschiedenen Plattformen laufen muss. Nun hören Sie, wie ein leitender Angestellter der Firma Pea eine Software seines Hauses präsentiert, die die syntaktische und semantische Korrektheit von Anforderungen, Modell des Problembereichs und Use Cases prüft. Wie reagieren Sie?

- Ich kaufe die Software nicht, weil man die semantische Korrektheit von Modellen nicht automatisch prüfen kann.
- Ich kaufe die Software nicht, weil die Verwendung des Produktes die Plattformunabhängigkeit meines Softwareprojektes gefährden könnte.
- Ich kaufe die Software, weil semantische Fehler in frühen Projektphasen besonders große Auswirkungen haben, die durch die Software vermieden werden können.

Kreuzen Sie alle wahren Aussagen an. Es sind mehrere Antworten möglich.

- OCL steht für Object Constraint Language
- Mit OCL lassen sich Vorbedingungen von Operationen spezifizieren
- Mit OCL lassen sich Nachbedingungen von Operationen spezifizieren
- OCL dient zur Manipulation von Modellen
- Mit OCL lassen sich Invarianten über die Struktur von Modellen spezifizieren

Erläutern Sie im Kontext der Modularisierung von Software die Begriffe High Cohesion und Low Coupling.

Nennen Sie drei Sprachelemente, die in einem Statechart Verwendung finden können, aber nicht in einem Zustandsautomaten.

Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 2: Modell des Problembereichs (24 Punkte)

Geben Sie ein Modell des Problembereichs an, das den im Anhang enthaltenen Text so präzise und vollständig wie möglich modelliert.

Name:	Matrikel-Nr.:
-------	---------------

Achtung: Platz für einen zweiten Versuch, nicht doppelt bearbeiten! Lösung eindeutig markieren!

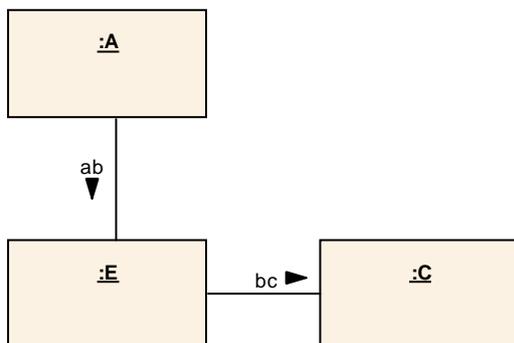
Name:

Matrikel-Nr.:

Aufgabe 3: Objektdiagramme (10+3+2=15 Punkte)

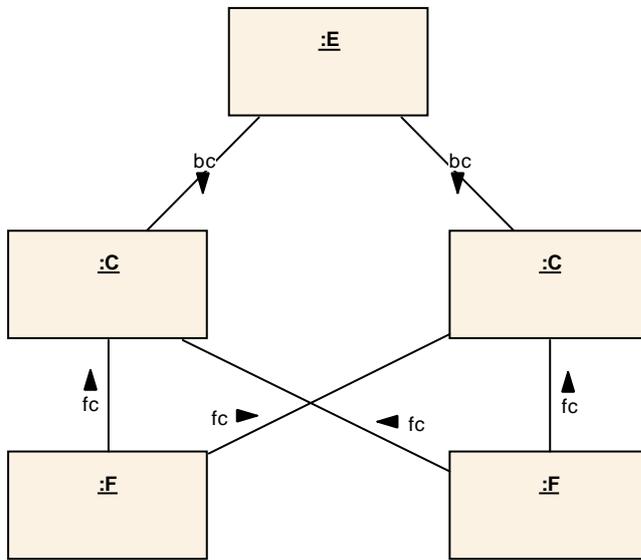
Überprüfen Sie für jedes der auf den folgenden Seiten angegebenen Objektdiagramme, ob es konsistent zum in der Referenz gegebenen Klassendiagramm ist. Ist ein Objektdiagramm konsistent, markieren Sie es mit „Ja“. Ist es nicht konsistent, markieren Sie es mit „Nein“ und geben Sie alle Inkonsistenzen an.

A)



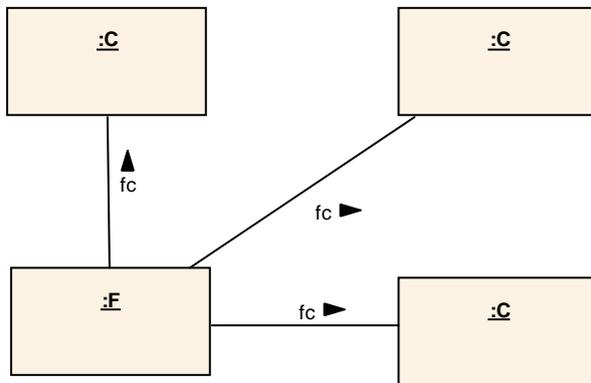
Name:	Matrikel-Nr.:
-------	---------------

B)



Name:	Matrikel-Nr.:
-------	---------------

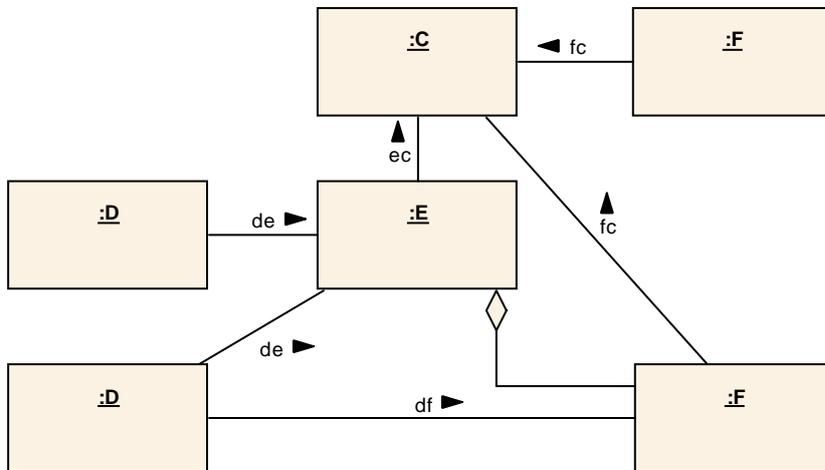
c)



Name:

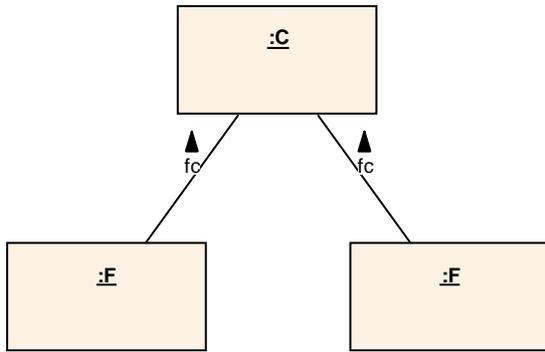
Matrikel-Nr.:

D)



Name:	Matrikel-Nr.:
-------	---------------

E)



Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 4: Aktivitätendiagramme (10 Punkte)

Stellen Sie den in der Referenz beschriebenen Ablauf der Eröffnung eines Kontos mit einem Aktivitätendiagramm dar. Nutzen Sie Nebenläufigkeit, wenn dies möglich ist.

Name:	Matrikel-Nr.:
-------	---------------

Achtung: Platz für einen zweiten Versuch, nicht doppelt bearbeiten! Lösung eindeutig markieren!

Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 5: Analysetabelle (19 Punkte)

- a) Erstellen Sie zu den Analysesequenzdiagrammen aus der Referenz eine Analysetabelle, die die Eigenschaften der genutzten Klassen in einer Tabelle zusammenführt. Geben Sie zusätzlich zum Klassennamen immer auch das Stereotyp-Symbol an.

Klassen	Aufgaben	Attribute	kennt (dauerhaft)

Name:	Matrikel-Nr.:
-------	---------------

Achtung: Platz für einen zweiten Versuch, nicht doppelt bearbeiten! Lösung eindeutig markieren!

Klassen	Aufgaben	Attribute	kennt (dauerhaft)

Name:	Matrikel-Nr.:
-------	---------------

- b) Geben Sie zur Analysetabelle im Anhang das zugehörige Analyse-Klassendiagramm an.

Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 6: Statecharts (12 Punkte)

Gegeben ist das in der Referenz abgebildete Statechart. Geben Sie das zu diesem Statechart äquivalente Zustandsdiagramm (auch Zustandsautomat genannt) an.

Name:	Matrikel-Nr.:
-------	---------------

Achtung: Platz für einen zweiten Versuch, nicht doppelt bearbeiten! Lösung eindeutig markieren!

Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 7: Implementierung (4+3+5=12 Punkte)

Wenn eine Sprache keine Mehrfachvererbung unterstützt, kann man diese trotzdem mit gegebenen Sprachmitteln implementieren. In der Vorlesung haben Sie die Implementierung von Mehrfachvererbung durch *Aufwärtsdelegation* und *Aufwärtsdelegation mit Interfaces* kennengelernt. Erläutern und vergleichen Sie die beiden Verfahren.

Name:

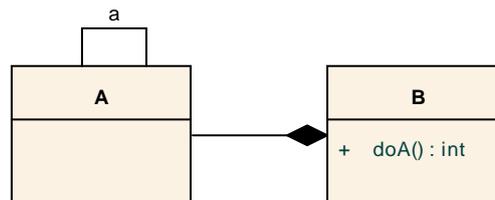
Matrikel-Nr.:

Aufgabe 8: Metamodellierung (1+2+2=5 Punkte)

Gegeben ist das aus der Vorlesung bekannte Metamodell für Klassendiagramme (siehe Referenz).

Geben Sie für das ebenfalls in konkreter Syntax gegebene Klassendiagramm (siehe unten) die entsprechende Repräsentation in abstrakter Syntax, also als Instanz des Metamodells an.

Hinweis: Geben Sie nur Attribute an, die sie neu belegen. TODO



Name:	Matrikel-Nr.:
-------	---------------

Achtung: Platz für einen zweiten Versuch, nicht doppelt bearbeiten! Lösung eindeutig markieren!

Name:	Matrikel-Nr.:
-------	---------------

Aufgabe 9: Protokolle

Es soll eine Klasse „Machine“ beschrieben werden. Diese soll sich wie folgt verhalten:

- Vor Nutzung von Instanzen der Klasse muss auf diesen zunächst die Methode `init()` aufgerufen werden
- Nun sind zwei Verwendungen möglich:
 - Durch Aufruf der Methode `load()` wird ein Objekt in den Speicher geladen. Dieses muss vor der Weiterverarbeitung der Aufruf der Methode `verify()` validiert werden
 - Durch Aufruf der Methode `create()` wird ein neues Objekt im Speicher erzeugt.
- Das nun im Speicher befindliche Objekt kann nun durch Aufruf der Methode `process()` verarbeitet werden. Wenn dies geschehen ist, geht die Machine-Instanz wieder in den Zustand vor Initialisierung.
- Sobald die Methode `init()` ausgeführt wurde, ist es jederzeit möglich, durch Aufruf der Methode `reset()` in den Zustand vor Initialisierung zu gelangen.

Geben Sie ein Klassendiagramm sowie ein dazu konsistentes Statechart an, die obigen Sachverhalt modellieren. Sie dürfen im Statechart keine Transitionsbeschriftung doppelt verwenden.