

Matr.-Nr.: \_\_\_\_\_

Nachname: \_\_\_\_\_

Studienkennzahl: \_\_\_\_\_

Vorname: \_\_\_\_\_

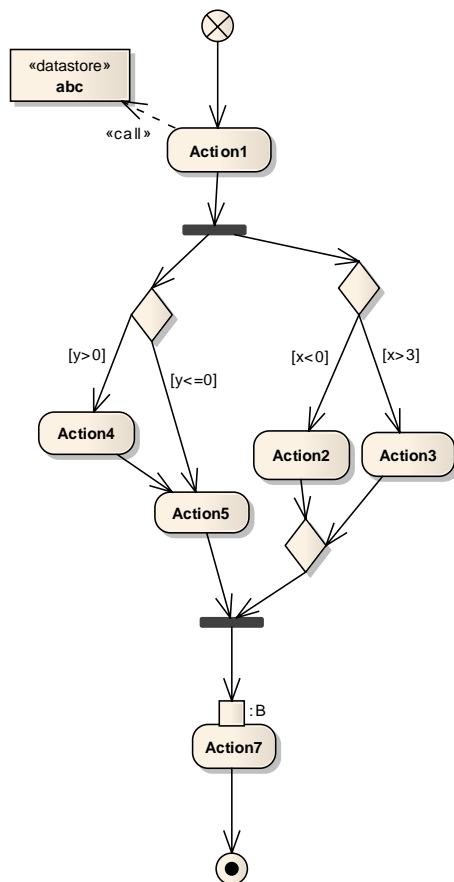
Erreichte Punkte: \_\_\_\_\_

*Hinweise zur Beantwortung:*

- Bewertung der Ja/Nein-Fragen:
  - korrekt beantwortet  $\Rightarrow$  3 Punkte
  - falsch beantwortet  $\Rightarrow$  1 Punkt Abzug
  - nicht beantwortet  $\Rightarrow$  0 Punkte
- Fragen zu UML beziehen sich ausschließlich auf den UML2-Standard bzw. auf den verwendeten Foliensatz, Spezifika bestimmter UML-Werkzeuge sind für die Beantwortung nicht relevant.
- Sie haben 30 Minuten Zeit.

*Frage 1. Fehlersuchbild Aktivitätsdiagramm (25 Punkte)*

Das folgende Aktivitätsdiagramm enthält 5 Fehler. Nennen Sie diese Fehler und führen Sie jeweils eine Begründung an.



Frage 2. Welche allgemeinen Aussagen über Aktivitätsdiagramme sind korrekt? (15 Punkte)

Der Fokus des Aktivitätsdiagramms sind prozedurale Verarbeitungsaspekte	<input type="checkbox"/> ja	<input type="checkbox"/> nein
Ein Vereinigungsknoten führt nebenläufige Abläufe wieder zusammen.	<input type="checkbox"/> ja	<input type="checkbox"/> nein
Token fließen entlang der Kanten vom Nachfolgerknoten zum Vorgängerknoten.	<input type="checkbox"/> ja	<input type="checkbox"/> nein
Aktionen sind nicht atomar und können abgebrochen werden.	<input type="checkbox"/> ja	<input type="checkbox"/> nein
Partitionen beeinflussen den Ablauf einer Aktivität.	<input type="checkbox"/> ja	<input type="checkbox"/> nein

Frage 3. Sequenzdiagramm unter Verwendung von vorgegebenen Operationen (35 Punkte)

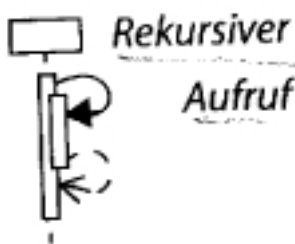
Gegeben sei das auf der folgenden Seite dargestellte Klassendiagramm, das einen Ausschnitt eines Informationssystems eines Consultingunternehmens zeigt. Folgende Annahmen führten zu diesem Klassendiagramm:

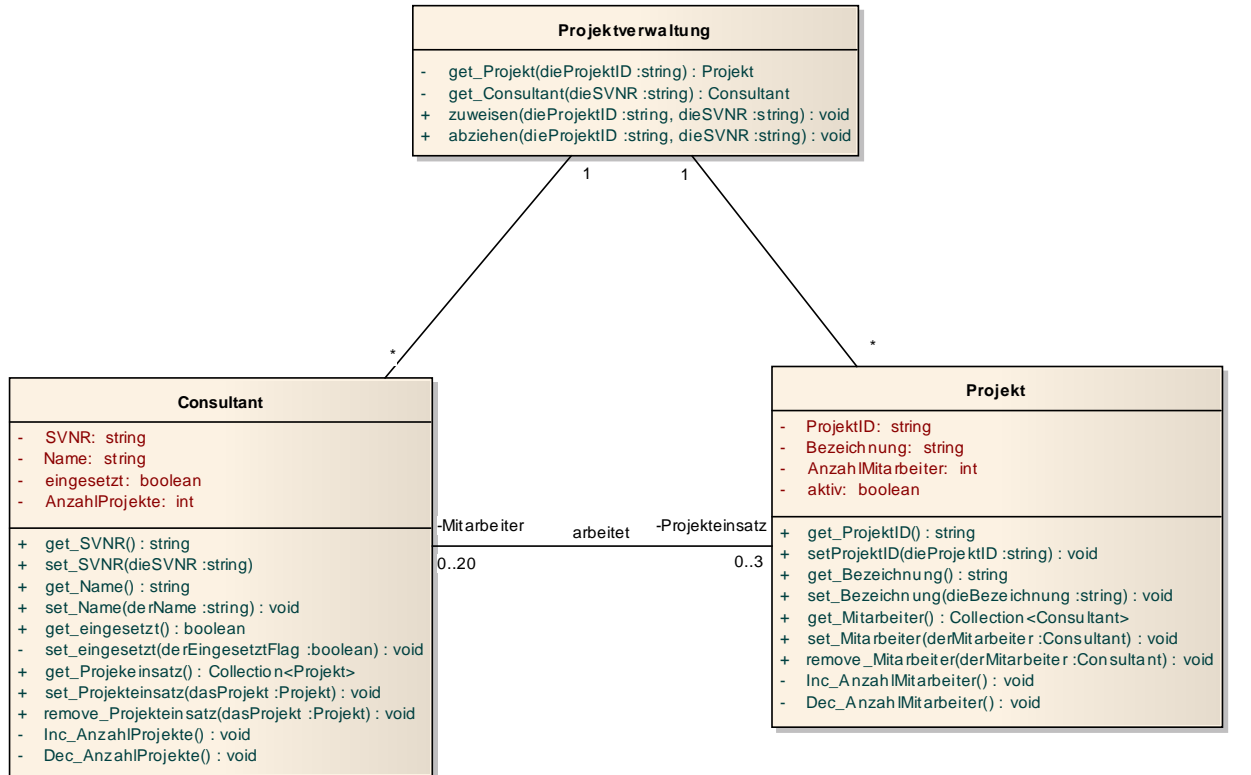
Die Klasse **Consultant** umfasst folgende Attribute: die eindeutig identifizierende Sozialversicherungsnummer (**SVNR**), den **Namen** und den Flag **eingesetzt**. Letzter ist auf **true** falls der Consultant im Moment einem Projekt zugewiesen ist, und sonst **false**. Für jedes **Projekt** werden die eindeutig identifizierende **ProjektID**, die **Bezeichnung**, **AnzahlMitarbeiter** und der Flag **aktiv** gespeichert. Letzterer ist **true** falls diesem Projekt mindestens ein Mitarbeiter zugewiesen ist, und sonst **false**. Für die beiden Klassen stehen get- und set-Operationen für die Attribute und die Beziehung zueinander zur Verfügung. Nur im Falle der Attribute **AnzahlMitarbeiter** und **AnzahlProjekte** gibt es stattdessen die Operationen zum Inkrementieren (**Inc...**) und Dekrementieren (**Dec...**). Weiters gibt es die Operation **removeProjekteinsatz** in der Klasse **Consultant**, um die Referenz auf ein **Projekt** zu löschen, und die Operation **removeMitarbeiter** in der Klasse **Projekt**, um einen **Consultant** aus der Collection der **Mitarbeiter** zu entfernen.

Die Klasse **Projektverwaltung** bildet die Geschäftslogik ab. Sie besitzt Beziehungen zu **Consultant** und **Projekt**, wodurch es ihr intern möglich ist aufgrund einer **ProjektID** ein **Projekt** (**getProjekt**), bzw. auf Grund einer **SVNR** einen **Consultant** (**getConsultant**) zu ermitteln. Die Projektverwaltung bietet Operationen zum Zuweisen eines Consultants zu einem Projekt, bzw. zum Abziehen eines Consultants von einem Projekt.

Vervollständigen Sie das Sequenzdiagramm unterhalb des Klassendiagramms, um die Sequenz der Aufrufe von Operationen für das erfolgreiche Abziehen eines Consultants (**SVNR='123'**) aus einem Projekt (**ProjektID='ABC'**) zu zeigen. (Sie können davon ausgehen, dass er danach immer noch zumindest einem Projekt zugeteilt ist.) Zu Beginn ruft ein Objekt der im Klassendiagramm nicht gezeigten Klasse **Anwendung** die geeignete Operation der geeigneten Klasse zum Abziehen eines Consultants auf. Die Geschäftslogik wird ausschliesslich über die Projektverwaltung abgebildet, d.h. es erfolgen keine Aufrufe zwischen **Consultant** und **Projekt**. Zeigen Sie auch die Aufrufe von nötigen *private* Operationen.

Hinweis:





:Anwendung

:Projektverwaltung

123 :Consultant

ABC :Projekt

.....

.....

.....

.....

*Frage 4. Zustandsdiagramm mit Call-Events (25 Punkte)*

Diese Frage basiert auf der Angabe und dem Klassendiagramm der vorhergehenden Frage. Modellieren Sie die Zustände von **eingesetzt** der Klasse **Consultant** mittels Zustandsdiagramm. Die Zustandsübergänge werden durch Call-Events der entsprechenden public Operationen ausgelöst. Im Rahmen der Events ausgelöste Aktivitäten sollen nicht direkt auf Attribute zugreifen, sondern geeignete Operationen aufrufen. Berücksichtigen Sie im Zustandsdiagramm auch die gegebenen Multiplizitäten zwischen Consultant und Projekt.