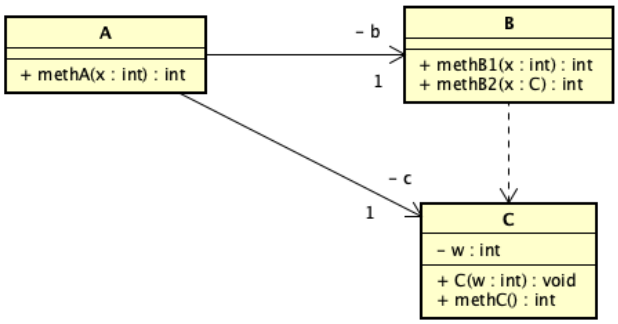
Aufgabe 6.1: Java-Klassen, UML-Klassen, UML-Sequenzdiagramm

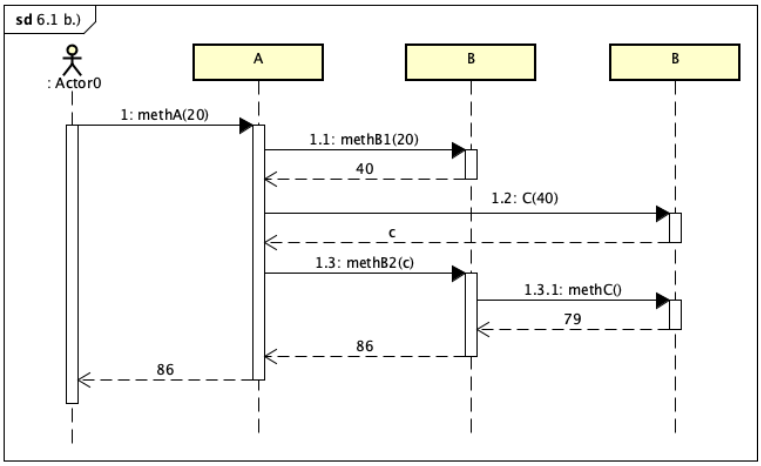
Gegeben seien nebenstehende Java-Klassen.

a) Modellieren Sie das passende UML-Klassenmodell inkl. der Beziehungen.**

b) Modellieren Sie in einem UML-Sequenzdiagramm den Ablauf beim Aufruf der Methode `methA(20)` mit den jeweiligen Rückgabewerten.

```
public class A {
    private B b= new B();
    private C c;

    public int methA(int x){
        int y= b.methB1(x);
        c = new C(y);
        return b.methB2(c);
    }
}
public class B {
    public int methB1(int x) {
        return x*2;
    }

    public int methB2(C c) {
        int z=c.methC();
        return z+7;
    }
}

public class C {
    private int w=0;

    public C(int i){
        this.w=i;
    }
    public int methC() {
        return w+39;
    }
}
```



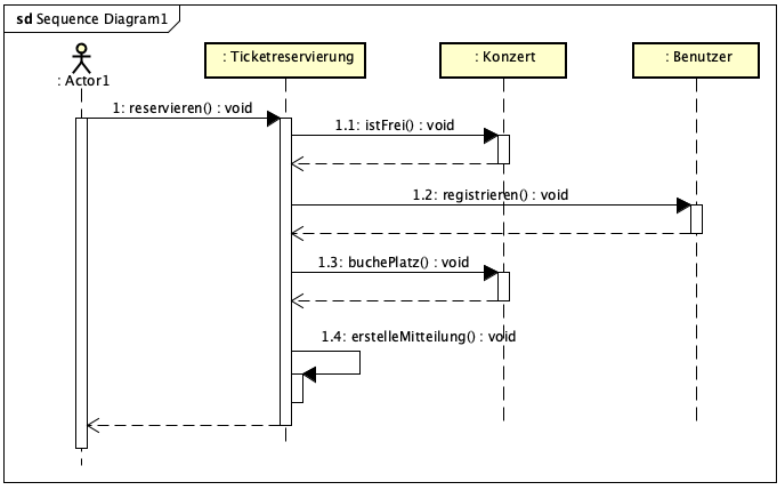


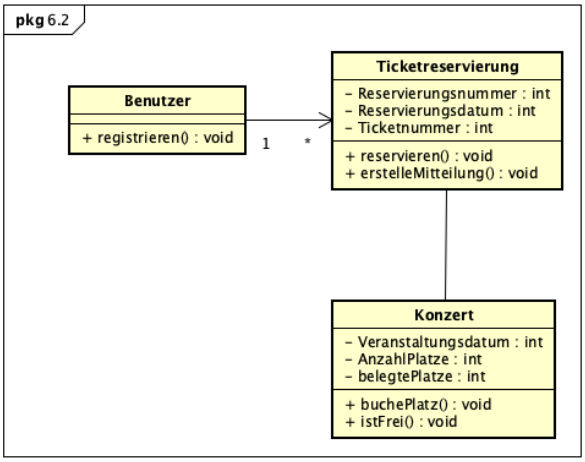
Aufgabe 6.2: UML-Klassendiagramm und UML-Sequenzdiagramm

Modellieren Sie die erfolgreiche Ticketreservierung (mit Reservierungsnummer, Reservierungsdatum, Ticketnummer, Veranstaltungsdatum etc.) ein Konzert mit ausschließlich Stehplätzen. Und zwar soll sich ein Benutzer für ein Konzert an einem bestimmten Tag eine Eintrittskarte reservieren können (mit der Methode `reserviere()`). Hierzu wird zunächst geprüft, dass es für das Konzert an diesem Tag noch freie Plätze gibt (Methode `istFrei`()). Da es sich um einen neuen Kunden handelt, werden dessen Daten erfasst (Methode `registrieren`()). Dann wird die Anzahl noch verfügbarer Tickets um eins verringert (Methode `buchePlatz()`). Und zum Schluss wird eine Mittelung erstellt (Methode `erstelleMitteilung`()), mit der dem Kunden per E-Mail die Bestätigung der erfolgreichen Reservierung gesendet wird.

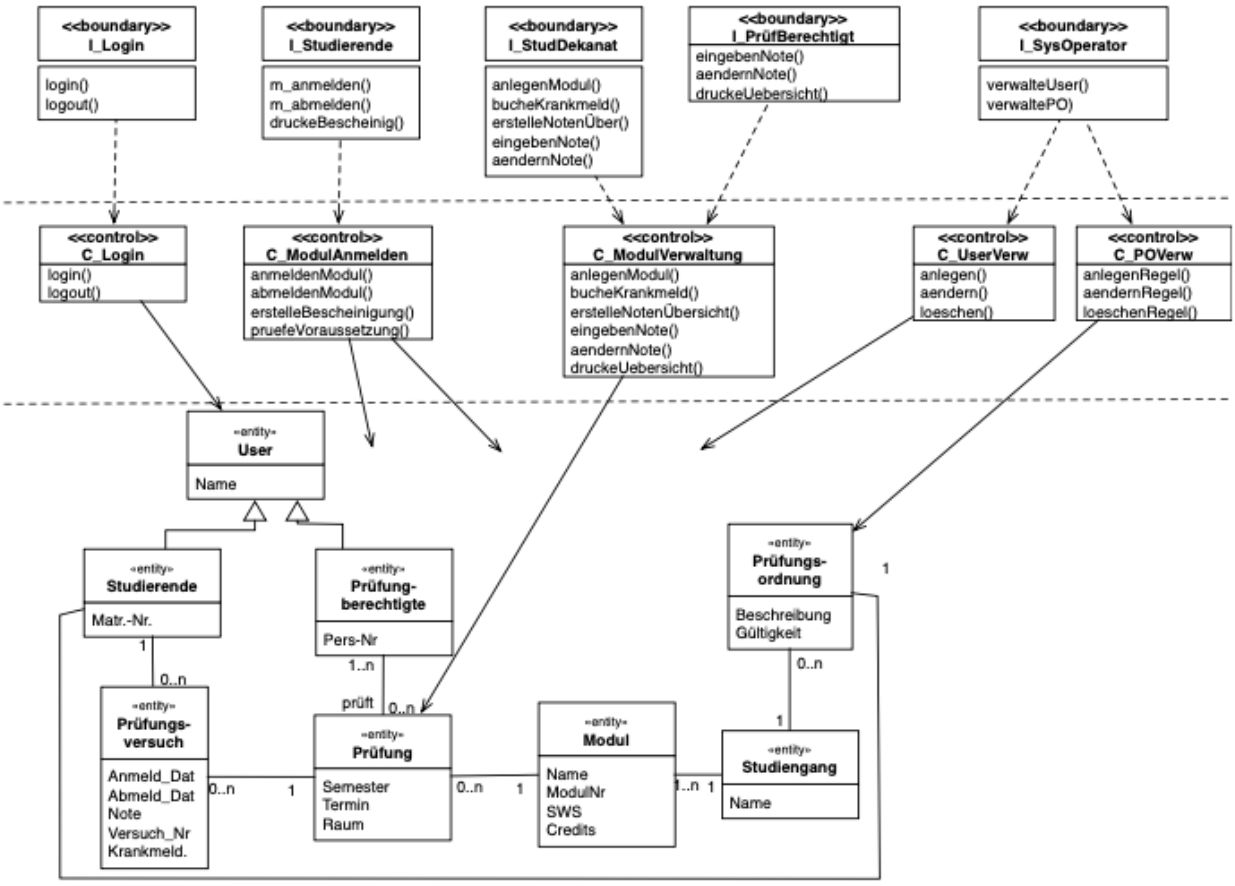
a) Bestimmen Sie die notwendigen Klassen* mit deren Beziehungen und ordnen Sie die Methoden jeweils der richtigen Klasse zu. Parameter, Attribute oder weitere Methoden sind nicht notwendig.

b) Modellieren Sie den Ablauf der erfolgreichen Reservierung in einem Sequenzdiagramm.





Aufgabe 6.3: Analyse des Modulbelegungsmanagementsystems ModMS – Pflichtaufgabe
Für das Modulbelegungsmanagementsystem soll die fachliche Analyse aus Übungsblatt 5 vervollständigt werden. Dabei sei in der Analyse das folgende fachliche Klassenmodell entwickelt worden. Alternativ können Sie natürlich auch das von Ihnen entworfene Klassenmodell verwenden.



Erstellen Sie für die beiden folgenden Use Cases ein **Sequenzdiagramm**. Erweitern Sie das Klassendiagramm bei Bedarf um zusätzliche Methoden und Beziehungen.
Bemerkungen: Achten Sie auf einen korrekten Informationsfluss, d.h. es muss sichergestellt sein, dass sich die interagierenden Objekte kennen und den aufgerufenen Methoden alle benötigten Informationen zur Verfügung stehen.

Use Case 1: „Modul anmelden" Actor: Studierende

Vorbedingung: Die/der Studierende ist im System als Nutzer registriert.

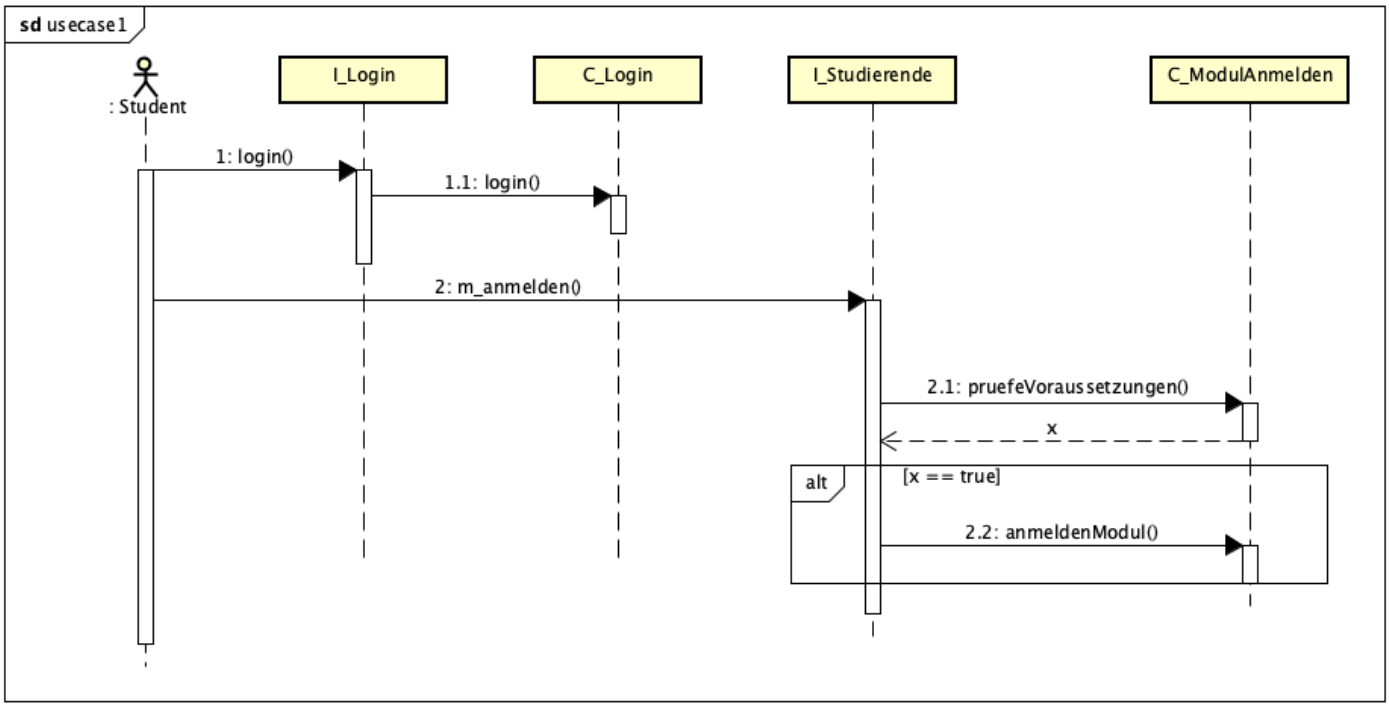
1. Die Studierende meldet sich mit Login-Namen und Kennwort an.
2. Das System zeigt die möglichen Funktionen an, z.B. Modulanmeldung/-abmeldung, Studienbescheinigung etc.
3. Die Studierende wählt die Funktion Modulanmeldung und das System bestimmt die im aktuellen Semester möglichen Module.
4. Die Studierende wählt eine oder mehrere Module aus und bestätigt die Anmeldung.
5. Das System überprüft, ob die Studentin berechtigt ist, an den ausgewählten Modulen (und den dazugehörigen Prüfungen) teilzunehmen.
6. Die Anmeldung zu den Modulen wird bestätigt und eine Modulliste generiert.

Nachbedingung: Die Anmeldung zu den Modulen ist im System als verbindlich verbucht.

Ausnahmen:

Zu 3: noch keine Module für das aktuelle Semester angelegt ⇒ eine entsprechende Meldung erscheint

Zu 5. eine oder mehrere Module dürfen nicht absolviert werden ⇒ die nicht erlaubten Module werden angezeigt, nur die erlaubten Module bleiben ausgewählt.



Use Case 2: „Noten verwalten - eingeben“ Actor: Studiendekanat, Prüfungsberechtigte

Vorbedingung: Die/der Prüfungsberechtigte ist im System als Nutzer registriert.

1. Die Prüfungsberechtigte meldet sich mit Login-Namen und Kennwort an.
2. Das System zeigt die möglichen Funktionen an.
3. Die Prüfungsberechtigte wählt die Notenverbuchung.
4. Das System bestimmt alle Modulprüfungen, die diese Prüfungsberechtigte anbietet.
5. Die Prüfungsberechtigte wählt ein Modulprüfung aus und das System bestimmt alle angemeldeten Studierenden.
6. Die Prüfungsberechtigte gibt die Noten für die Studierenden ein und bestätigt die Eingabe.
7. Das System überprüft die Eingabe auf Plausibilität.
8. Die Notenübersicht wird generiert und als PDF-Export zur Verfügung gestellt.

Nachbedingung: Die Noten sind im System verbucht.

Ausnahmen:

Zu 7: eine oder mehrere Noten sind nicht plausibel ⇒ die entsprechenden Noten werden hervorgehoben und können korrigiert werden; weiter mit Schritt 6.