Fachhochschule Deggendorf	***Prüfungsteilnehmer*** *(bitte in Druckschrift)*
Prüfungsfach: **SW Eng.**	***Name:***
Aufgabensteller: **Prof. Dr. P. Jüttner**	***Vorname:***
Probeklausur WS 2012	

Vorbemerkungen: Diese Probeklausur gibt den ungefähren Inhalt einer Klausur wieder. Die zu vergebenden Punkte sind als „ca.-Werte“ zu verstehen. Die Klausur ist bestanden bei ca. 45% der möglichen Punkte.

Allgemein

1. Warum können manchmal nicht alle Anforderungen an eine Software erfüllt werden? Erläutern Sie Ihre Antwort an einem Beispiel (2 Punkte)

   Anforderungen an SW können widersprüchlich sein, d.h. sie lassen sich nicht gemeinsam erfüllen- Beispiel: Hohe Qualität gegen niedrige Kosten

2. Nennen Sie 2 Eigenschaften von Software, die nie erfüllbar sind. Begründen Sie Ihre Aussage. (2 Punkte, ohne Begründung kein Punkt)

   Fehlerfreiheit (es gibt keine allgemeine Methode, Fehlerfreiheit von SW zu beweisen oder zu konstruieren)

SW Qualitätssicherung

3. Warum ist es wichtig das richtige Produkt zu entwickeln? (3 Punkte)

   Projekt-/Firmen-Erfolg hängt vom erfolgreichen Produkt ab

4. Warum ist eine Qualitätsmessung ohne weitere Massnahmen nicht sinnvoll?(3 Punkte)

   Messung an sich macht keinen Sinn, die gemessenen Ergebnisse müssen bewertet werden und Zielgrößen definiert werden.

5. Warum ist eine Vorbereitung der Teilnehmer vor dem Review-Meeting für ein erfolgreiches Objekt-Review wichtig?
   (3 Punkte)
   Während des Reviewmeetings besteht keine Möglichkeit den Review-Gegenstand zu prüfen.

6. Welche Aktion wird vom Management nach einem roten Projektreview erwartet? (3 Punkte)

   Unterstützung des Projekts

7. Der folgende Code (ohne #includes) soll ein Programm zur rekursiven Berechnung der Fakultät einer einzulesenden natürlichen Zahl (Typ unsigned long) realisieren. Führen Sie ein Codereview durch und notieren Sie Ihre Anmerkungen (maximal 4) in der unten angegebenen Tabelle. (10 Punkte)

```
1   void fakultaet(char z)
2   {    /* rekursive Berechnung der Fakultät */
3        long erg = 1;
4        long zaehler = z;
5        for (;zaehler>0; zaehler--)
6        {
7          erg = erg - zaehler;
8   }
9         return;
10 };
11
12 int main(void)
13 {
14     long s; /* Zahl deren Fakultät zu berechnen ist */
15     scanf("Bitte geben Sie eine Zahl ein:\n%f",s);
16     printf("\nDie Fakultaet von %ld ist %ld\n",fakultaet(s));
17     system("PAUSE");
18 }
```

Codezeile	Anmerkung
2 ff	keine rekursive Berechnung
1	Parameter vom Typ char falsch
5-7	Algorithmus falsch, es muss * statt - heissen
14	falscher Typ long, es muss unsigned long heissen

Requirements Engineering

8. Wie können Requirements ermittelt werden? Geben Sie 3 Beispiele an (3 Punkte)

   Interview mit Stakeholdern

   andere Projekte

   Übernahme aus Lastenheft

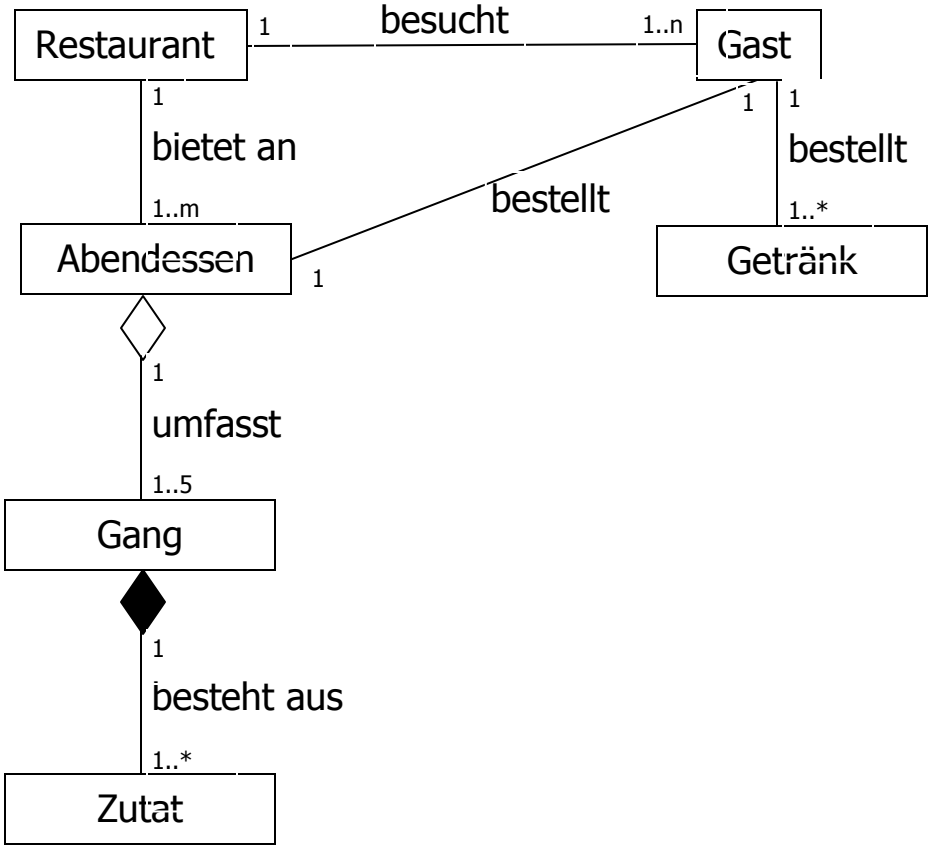
9. Worauf hin müssen Requirements überprüft werden? Geben Sie 5 Gesichtspunkte an (5 Punkte)

   Sind Requirements
   - adäquat (der Aufgabestellung angemessen)
   - tesbar
   - konsisten (widerspruchsfrei)
   - eindeutig
   - überdefiniert

OOA und OOD

10. Ein Abendessen in einem Restaurant besteht aus bis zu 5 Gängen. Jeder Gang besteht aus mehreren, unbegrenzt vielen Zutaten. Ein Gast im Restaurant isst ein Abendessen und trinkt dazu mehrere, beliebig viele Getränke. Modellieren Sie detailliert in einem UML Diagramm die Beziehungen zwischen den Klassen Gast, Restaurant, Abendessen, Gang, Zutat und Getränk. (10 Punkte)

    Hinweis: Das Diagramm unten stellt eine mögliche Lösung dar, andere Lösungen sind denkbar. Annahme ist, dass ein Gast nur ein Abendessen bestellt und das Lokal nur eine bestimmte Anzahl von Abendessen anbietet.


Restaurant
1
besucht
1..n
Gast
1
bietet an
1..m
bestellt
1
1
1
bestellt
1..*
Abendessen
Getränk
1
umfasst
1..5
Gang
1
besteht aus
1..*
Zutat

11. Im Folgenden wird im Rahmen einer Produktbeschreibung (Auszug) ein Geldautomat definiert:
    - Ein Geldautomat ermöglicht das Abheben von Geldbeträgen von einem Girokonto
    - Es kann ein Geldbetrag bis maximal 1000 Euro abgehoben werden
    - Zum Abheben ist eine gültige EC-Karte notwendig
    - Zusätzlich zur EC-Karte muss die PIN Nummer über Tastatur eingegeben werden.
    - Der abzuhebende Geldbetrag kann über Display oder über Tastatur gewählt werden
    - Ein Abhebevorgang kann jederzeit über Tastatur abgebrochen werden.

    - Identifizieren Sie Kandidaten für Klassen um die Software für den Geldautomaten zu modellieren. Begründen Sie Ihre Auswahl. Welche Begriffe sind keine Kandidaten für Klassen und warum? (10 Punkte)

      Klassenkandidaten:
      - Girokonto: spielt aktive Rolle
      - Display/Tastatur (Ein-/Ausgabeeinheit): spielt aktive Rolle im Ablauf
      - EC-Karte: kann aktive Rolle spielen
      - Abhebevorgang: beinhaltet Gesamtablauf

      Keine Klassenkandidaten
      - Geldbetrag: passive Rolle, nur Parameter mit Zahlenwert
      - PIN-Nummer: passive Rolle, nur Parameter mit Zahlenwert
      - Geldautomat: Gesamtsystem

SW Test

12. Warum kann durch Testen nicht die Korrektheit von SW bewiesen werden? (2 Punkte)

    Test hat immer Stichprobencharakter

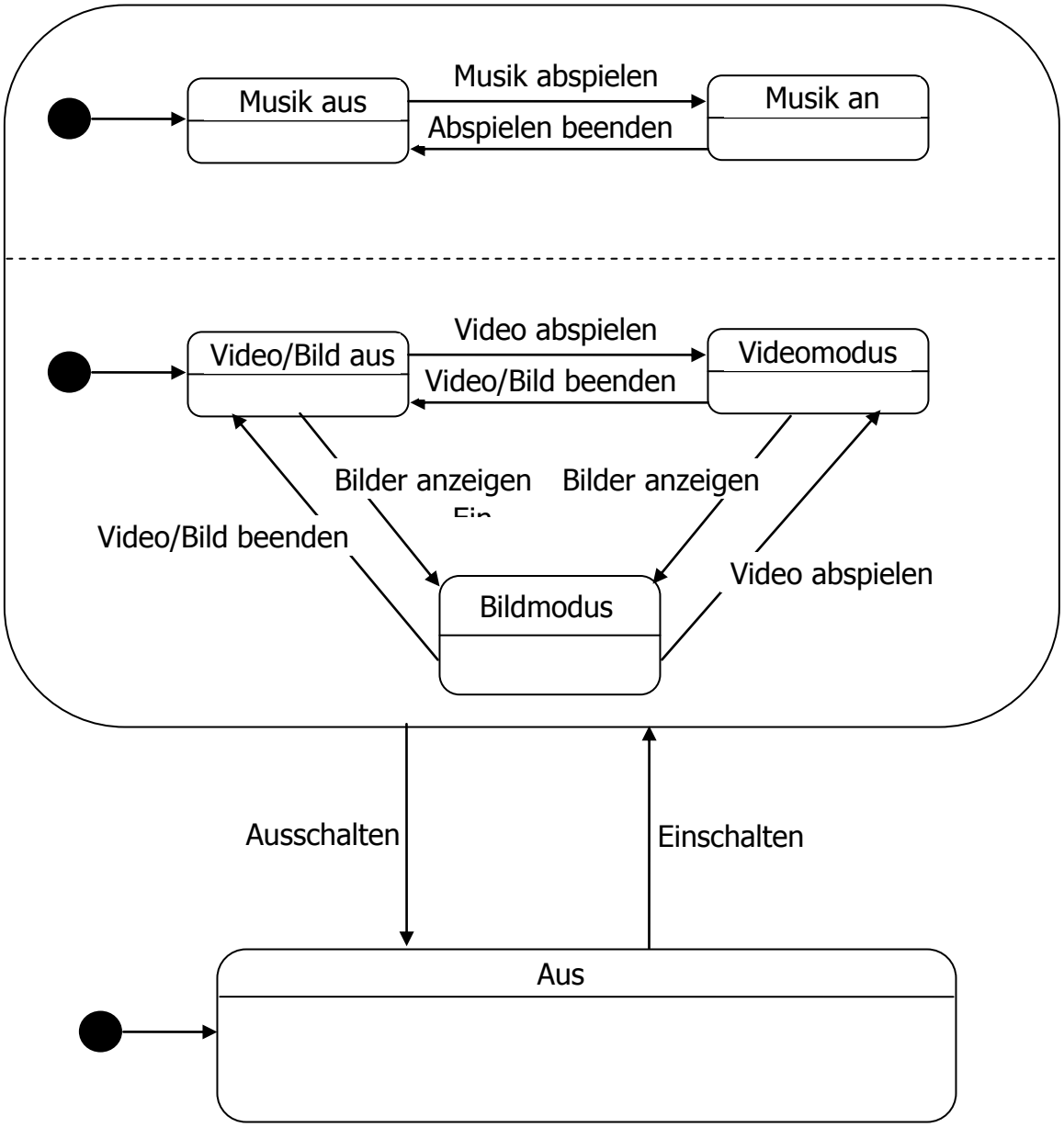
13. Für was wird ein Testvollständigkeitskriterium verwendet? (2 Punkte)

    Messen Testfortschritt, Qualitätsmaß des Tests

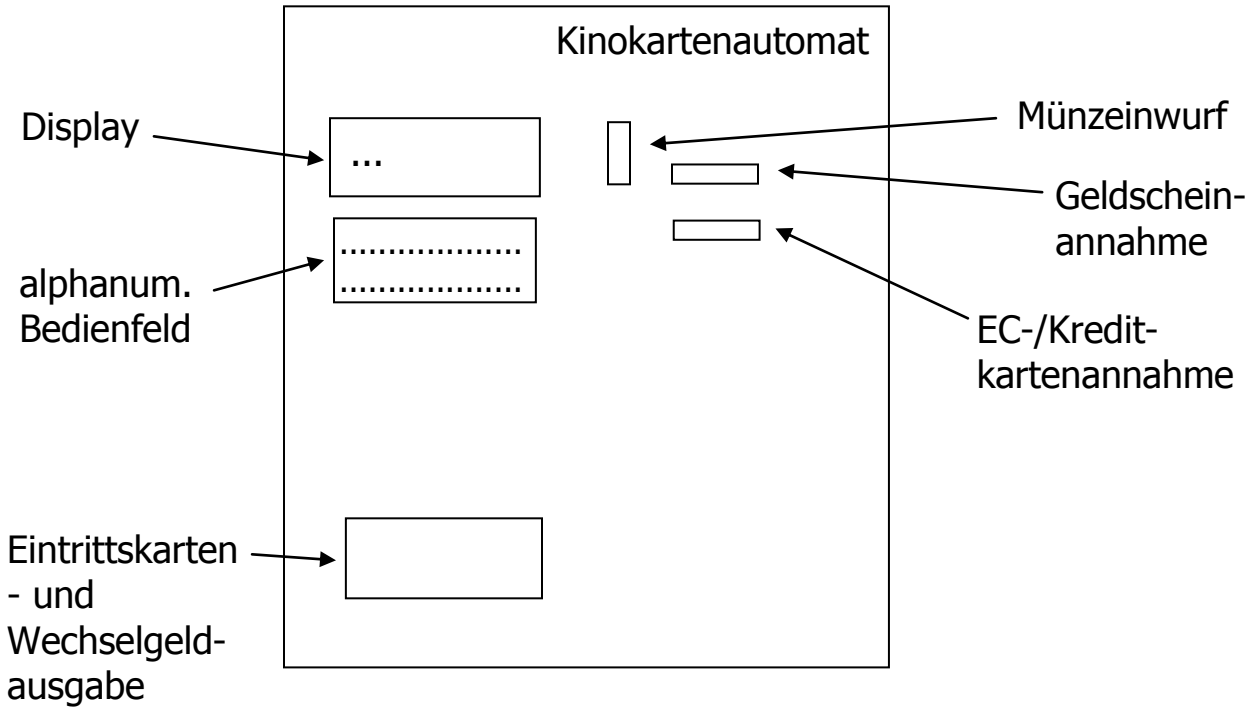
14. **Zustandsautomat MP3-Spieler (10 Punkte)**
    Ein MP3-Spieler lässt sich generell ein- und ausschalten. Er besitzt folgende Funktionen:
    - F1: Abspielen von Musik
    - F2: Abspielen von Videos
    - F3:Anzeigen von Bildern

F1 kann allein oder gleichzeitig mit F2 oder F3 ausgeführt werden, F2 und F3 können nur alleine oder zusammen mit F1 ausgeführt werden. Alle Funktionen werden über Bedienelemente gesteuert. Standardmäßig sind alle Funktionen ausgeschaltet und müssen erst aktiviert werden.

Zeichnen Sie einen entsprechenden Zustandsautomaten in UML 2.0 mit seinen Zuständen sowie den Zustandsübergängen mit den zugehörigen Auslösern (Trigger).



15. **Kinokartenautomat (25 Punkte)**
Im einem Regensburger Kino soll ein Automat zum Verkauf der Eintrittskarten installiert werden, der folgendermaßen aussieht:



Ihre Firma, die Globosoft AG, hat den Auftrag des Kinos gewonnen, die Software der neuen Automaten zu entwickeln. Ihr Auftraggeber hat Ihnen die Anforderungen spezifiziert, von denen Sie unten einen Ausschnitt sehen:

R-1.) Die Software soll auf Mikrocontrollern ablaufen, die den Kinokartenautomaten steuern.
R-2.) Der Automat soll Kinokarten für einen einmaligen Kinobesuch einer oder mehrerer Personen und für mehrfache Kinobesuche einer Person ausgeben.
R-3.) Die Eintrittskarten werden von einem Drucker gedruckt, der in den Automaten integriert ist.
R-4.) Im Fall, dass eine Karte für den einmaligen Besuch gekauft werden soll, wird über ein Alphanumerisches Bedienfeld die Anzahl der Personen ausgewählt.
R-5.) Die Berechnung des Preises einer Karte wird im Automat durchgeführt.
Eine Karte für einen einmaligen Besuch einer Person kostet 8€, jede weitere Person kostet 7€.
R-6.) Für den mehrfachen Kinobesuch einer Person soll der Kauf von bis zu 10 Karten auf einmal möglich sein. Die Anzahl der Karten für den mehrfachen Besuch wird über das Bedienfeld eingegeben.
R-7.) Die Bezahlung am Automat kann per Geldschein, Münzen, EC- oder Kreditkarten erfolgen. Bei Kartenzahlung muss der volle Betrag per Karte bezahlt werden.

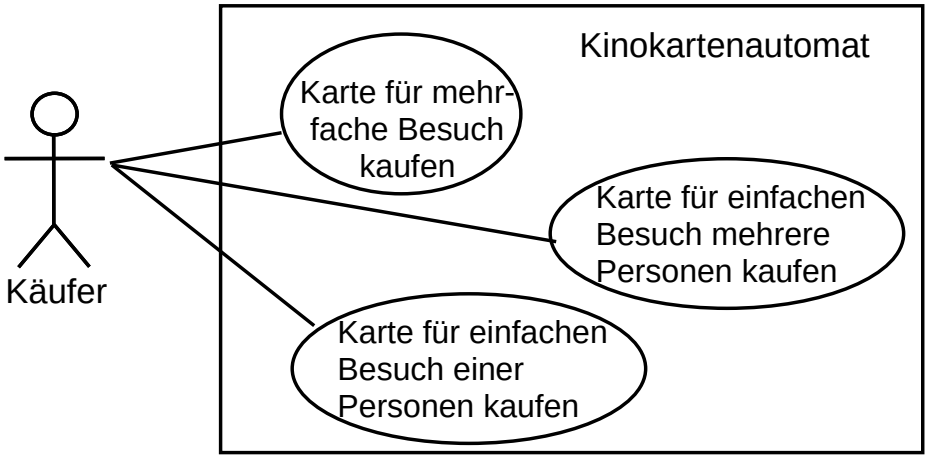
R-8.) Bestimmte große Geldscheine werden vom Automat nicht angenommen.

Führen Sie folgende Aufgabenschritte durch

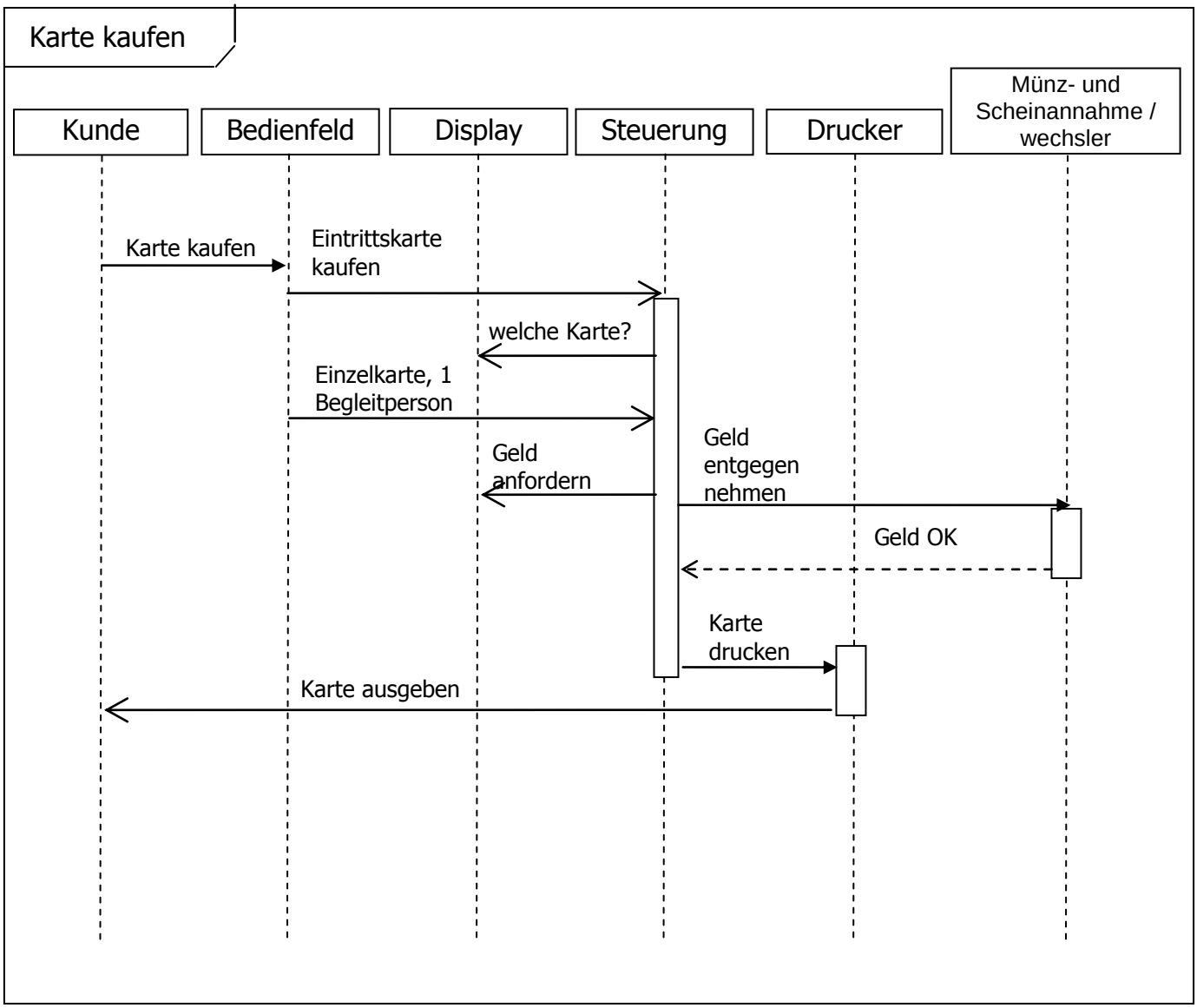
1.) Analysieren Sie die Requirements und identifizieren Sie 3 Requirements, die noch einer weiteren Klärung bedürfen. Geben Sie an, was zu klären ist.

R1: Typ Mikrocontroller nicht bekannt
R2: genaue Anzahl der Personen ist nicht definiert.
R8: welche Geldscheine werden nicht akzeptiert?

2.) Erstellen Sie ein UML 2.0 konformes Use Case Diagramm mit mindestens drei Use Cases.



3.) Erstellen Sie ein UML 2.0 konformes Sequenzdiagramm, das den Kauf genau einer Eintrittskarte für den einmaligen Besuch des Kinos mit einer Begleitperson beschreibt.

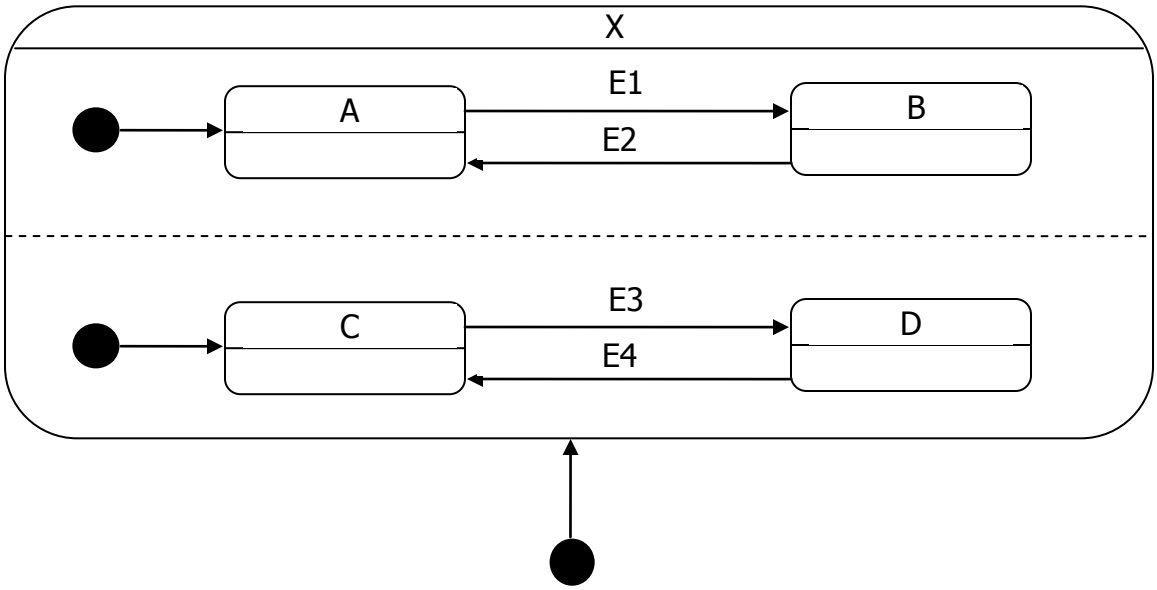


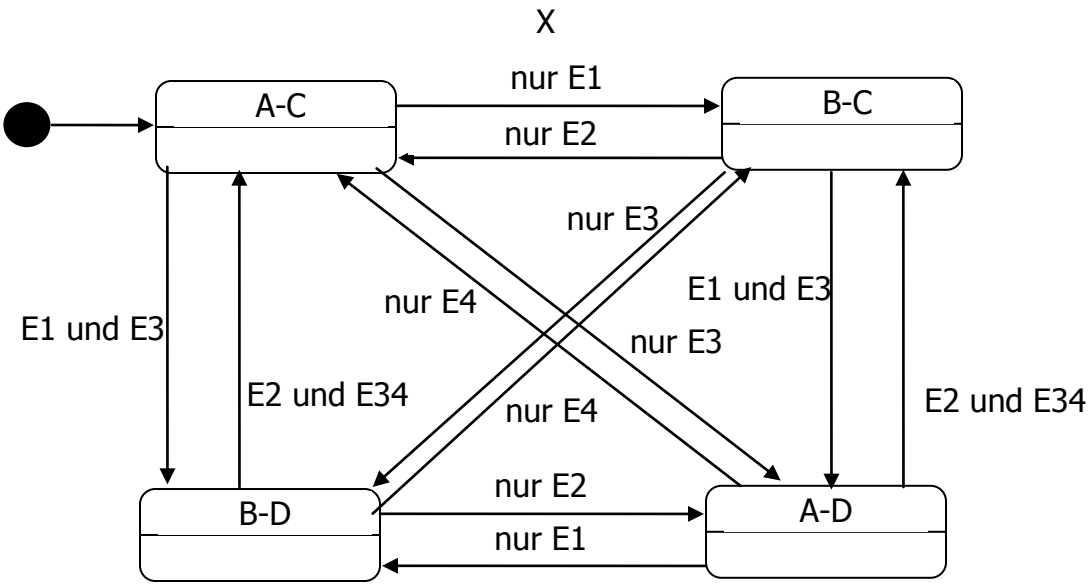
4.) Geben Sie zu Requirement R-6 einen negativen Testfall an

Vorbedingung: keine
Testdurchführung: 11 Karten für eine Person kaufen
Erwartete Reaktion: Kauf wird vom Automat abgelehnt.
Nachbedingung: Automat funktioniert nach wie vor.

14. **Sequentialisierung eines parallelen Automaten (10 Punkte)**
Gegeben ist folgender paralleler Automat:
Wandeln Sie den Automaten in einen äquivalenten, sequentiellen Automaten um.

Hinweis: Kombinieren Sie Zustände aus den parallelen Teilen in geeignete neue Zustände des sequentiellen Automaten. Verfahren Sie ebenso mit den Ereignissen




X
A-C
B-C
B-D
A-D
nur E1
nur E2
nur E3
nur E4
E1 und E3
E2 und E34
E1 und E3
nur E3
nur E4
E2 und E34
nur E2
nur E1