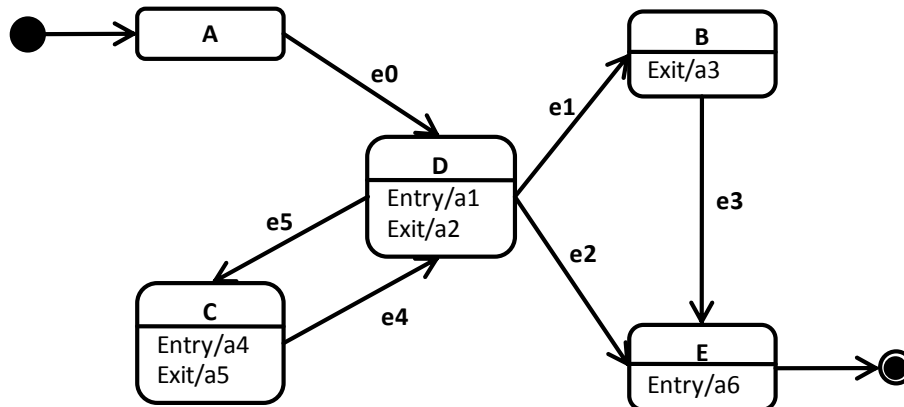




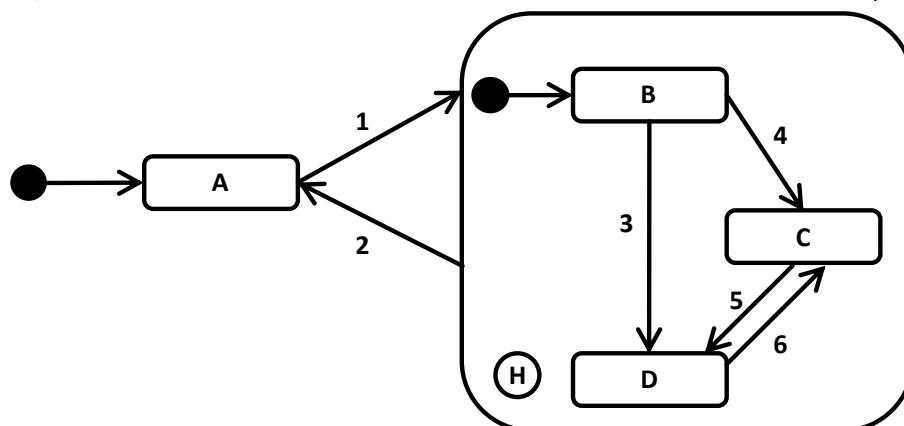
UML, Swing

Aufgabe 1: UML-Zustandsautomaten (4 Punkte)

- a) Wandeln Sie folgenden UML-Zustandsautomaten in einen äquivalenten Mealey-Automaten um. Die im UML-Zustandsautomaten angegebenen Entry-/Exit-Aktionen können Sie als Ausgabewerte auffassen. (2 Punkte)



- b) Wandeln Sie den folgenden, hierarchischen UML-Zustandsautomaten mit Gedächtnis in einen äquivalenten, nicht hierarchischen UML-Zustandsautomaten ohne Gedächtnis um. (2 Punkte)



Aufgabe 2: Modellierung mit UML (4 Punkte)

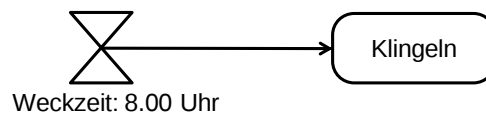
Gegeben Sei folgendes Szenario:

In einer Bibliothek können Bücher ausgeliehen werden. Von jedem Buch kann es ein oder mehr Exemplare geben. Bücher können von jedem Bibliotheksmitglied für maximal vier Wochen ausgeliehen werden. Mitglieder der Bibliothek dürfen maximal acht Exemplare entleihen. Ist das Mitglied jedoch ein Mitarbeiter der Bibliothek, so darf es maximal 14 Exemplare ausleihen.

Finden Sie in obigem Szenario alle beteiligten Klassen sowie Assoziationen zwischen den Klassen und zeichnen Sie ein Klassendiagramm. Berücksichtigen Sie dabei Multiplizitäten, Rollen und Leserichtungen.

Aufgabe 3: UML-Aktivitätsdiagramm (8 Punkte)

Modellieren Sie die nachfolgend beschriebene Seehundfütterung in einem einzigen Aktivitätsdiagramm. Achten Sie darauf, auch eventuelle Ein- und Ausgabeobjekte im Diagramm darzustellen. Zeitsignale werden im Diagramm durch eine Sanduhr dargestellt, wie im folgenden Beispiel zu sehen:



Der Tierpfleger der Karlsruher Zoos sitzt in seinem Pausenraum und wartet auf den Beginn der Seehundfütterung. Um 17 Uhr betritt er das Seehundgehege. Danach sucht er aus den Zuschauern einen aus, der ihm bei der Fütterung helfen darf. Hat er einen Zuschauer gefunden, betritt dieser zunächst auch das Gehege. Dann darf der Zuschauer die Seehunde Peter und Paula mit einem Pfiff herbei rufen. Peter und Paula watscheln daraufhin an den Beckenrand. Sobald beide Seehunde am Beckenrand sind, gibt der Tierpfleger dem Zuschauer einen Fisch aus dem Eimer. Der Zuschauer nimmt den Fisch und wirft anschließend gleichzeitig mit dem Tierpfleger den Seehunden den Fisch zu. Der Tierpfleger wirft seinen Fisch zur trächtigen Paula, der Zuschauer seinen zu Peter. Der Tierpfleger und der Zuschauer warten bis die Seehunde den Fisch gefressen haben. Wenn noch Fische im Eimer sind, gibt der Tierpfleger dem Zuschauer den nächsten Fisch und das Fütterungsritual beginnt von vorne. Ist der Eimer leer, muss zuerst der Zuschauer das Gehege verlassen. Ist der Zuschauer draußen, verlässt der Tierpfleger ebenfalls das Gehege und die Fütterung ist beendet.

An manchen Tagen findet der Tierpfleger keinen Zuschauer, der bei der Fütterung helfen will. In diesem Fall pfeift er die Seehunde selbst herbei. Diese watscheln auf den Pfiff an den Beckenrand. Sind beide Seehunde am Beckenrand, wirft der Tierpfleger zuerst der trächtigen Paula einen Fisch zu. Hat Paula den Fisch aufgefressen, bekommt Peter einen Fisch zugeworfen. Hat Peter den Fisch aufgefressen und es befinden sich noch Fische im Eimer beginnt das Fütterungsritual von vorne. Ist der Eimer leer, verlässt der Tierpfleger das Gehege und die Fütterung ist beendet.

Hinweis: Gehen Sie davon aus, dass immer eine gerade Anzahl von Fischen im Eimer ist. Das Diagramm wird dadurch einfacher. Verwenden Sie Partitionen (Schwimmbahnen) um zu kennzeichnen, welche Aktionen von dem Tierpfleger, dem Zuschauer, Peter und Paula durchgeführt werden.

Aufgabe 4: Swing - Adressbuch (10 Punkte)

Für diese Aufgabe steht Ihnen ein Grundgerüst in Form eines Eclipse-Projektes zur Verfügung. Dieses finden Sie als ZIP-Datei im SVN-Verzeichnis der Vorlesung:

<https://svn.ipd.uni-karlsruhe.de/lehre/vorlesung/SWT1/SS09/studies>

Das Grundgerüst bietet Funktionen zum Verwalten von Adressen an. Dazu zählen das Erstellen, Bearbeiten und Löschen von Adressen (**ab.Address**). Eine Adresse besteht aus mehreren Feldern (Vorname, Nachname, E-Mail, ...) und ist einer Kategorie (**ab.Category**) (Arbeit, Familie, Freunde, Sonstige) zugewiesen werden. Jede Kategorie besitzt einen Namen, eine Farbe sowie ein Icon. Alle Adressen sollen in die Tabelle vom Typ **ab.table.AddressTable** eingetragen werden. Diese Funktionalität ist jedoch noch nicht vollständig implementiert. Vervollständigen Sie das Programmgerüst so, dass alle Daten der eingetragenen Adressen in der Tabelle angezeigt werden. Die angezeigten Adressen sollen entsprechend der ihnen zugewiesenen Kategorie farblich, mit einem Icon sowie dem Namen der Kategorie hervorgehoben werden. Hierzu sind mindestens die im Programmgerüst durch die Kommentare „TODO“ markierten Stellen anzupassen. Zusätzlich kann es jedoch notwendig sein, neue Klassen zu erstellen.

Eine dauerhafte Speicherung der Adressen ist nicht notwendig. Die eingegebenen Adressen dürfen nach dem Beenden der Anwendung verloren gehen.

Hinweis: Bitte lesen Sie sich den folgenden Text aufmerksam durch. Missachtung der Anweisungen wird Sie Punkte kosten!

Den Quelltext Ihres Programmes drucken Sie bitte aus und geben ihn zusammen mit dem Übungsblatt ab. Zusätzlich müssen Sie sowohl den Quelltext als auch die Java-Anwendung an folgende E-Post-Adresse senden:

swt1@ipd.uka.de

Als Betreff geben Sie bitte **[Übungsblatt 3] <TUTNR> <MATRNR> <VORNAME> <NACHNAME>** an.

Hängen Sie den Java-Quelltext als ZIP-Archiv an die E-Mail an. Der Name des ZIP-Archivs muss folgendes Format haben: **<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.zip**. Das ZIP-Archiv darf nur den Java-Quelltext (d.h. Dateien mit der Endung .java) und evtl. benötigte Ressourcen (wie bspw. Icons) enthalten. Sofern Sie mit Paketen gearbeitet haben, muss diese Ordnerstruktur ebenfalls in Ihrem ZIP-Archiv vorhanden sein.

Verwenden Sie in Ihrem Java-Quelltext nur ASCII-Zeichen und vermeiden Sie IDE-spezifische Java-Annotationen. Die lauffähige Java-Anwendung hängen Sie als **ausführbares JAR-Archiv** zusätzlich zum ZIP-Archiv an die E-Mail an. Es werden nur Java-Anwendungen akzeptiert, welche als ausführbares JAR-Archiv an die E-Mail angehängt sind! Das JAR-Archiv muss als separater Anhang gesendet werden. Der Name des JAR-Archivs muss wie folgt lauten: **<TUTNR>_<MATRNR>_<VORNAME>_<NACHNAME>.jar**

Beispiel: Sie heißen Hans Müller, gehen in das Tutorium 7 und Ihre Matrikelnummer ist 123456. Ihr Betreff sieht dann wie folgt aus:

[Uebungsblatt 3] 07 123456 Hans Mueller

Die Anhänge heißen dann:

07_123456_Hans_Mueller.zip / 07_123456_Hans_Mueller.jar

Die E-Mail mit beiden Anhängen muss bis zum offiziellen Abgabetermin bei uns eingegangen sein. Vergewissern Sie sich, dass Sie alle Code-Dateien, die zur Übersetzung der Java-Anwendung notwendig sind, mitschicken und der Quelltext übersetzt werden kann. Prüfen Sie, dass die Java-Anwendung aus dem JAR-Archiv lauffähig ist. Andere Dateiformate werden nicht akzeptiert.

Bei Missachtung der obigen Anforderungen oder Teilen davon wird die gesamte Aufgabe mit 0 Punkte bewertet.