

Modellierung und Programmierung 1 Übungsserie 5

Abgabetermin: 12.01.2014, 23:55 Uhr

Grundsätzlich sind Nebenrechnungen anzugeben und Antworten zu begründen.
Einzureichen sind, bei mehreren Dateien als .zip-Archiv:
Lösungen und UML-Klassendiagramme als .pdf-Datei, Programme als Quellcode,
Ergebnisdateien, Online-Dokumentationen.

1. Klassen

Der einarmige Bandit ist ein Spielautomat. Der Spieler wirft zu Beginn seinen Einsatz ein. Anschließend werden drei Zahlen zwischen 0 und 9 gewürfelt. Sind alle drei Zahlen gleich, wird das Vierfache, sind zwei Zahlen gleich, das Doppelte des Einsatzes zurückgezahlt, sonst wird der Einsatz einbehalten.

- Schreiben Sie mittels der Klasse **Wuerfel** aus der Serie 4 ein Programm **BanditTest.java** zum Berechnen der Chancen im Banditspiel: Vereinbaren Sie drei Würfel mit den Werten 0 .. 9. Ermitteln Sie experimentell, durch eine genügend große Anzahl von Versuchen (10, 1 000, 100 000, 10 000 000, 1 000 000 000), die relative Häufigkeit, dass alle drei Würfel, zwei Würfel bzw. keiner der Würfel die gleiche Augenzahl gewürfelt hat.
Geben Sie am Ende Ihres Programms die ermittelten relativen Häufigkeiten aus. Leiten Sie die Ergebnisse aller Versuche in eine Datei **BanditTest.out** um.
- Berechnen Sie die zu erwartenden Wahrscheinlichkeiten für die genannten Ereignisse. Erweitern Sie Ihr Programm: Geben Sie jeweils die relativen Fehler der Wahrscheinlichkeiten zu den experimentell ermittelten Werten an.

Hinweis:

n	Anzahl der Versuche
E	zufälliges Ereignis
$H(E)$	absolute Häufigkeit, Anzahl der eingetretenen Ereignisse E
$h(E) = \frac{H(E)}{n}$	relative Häufigkeit
$P(E)$	Wahrscheinlichkeit, dass ein Ereignis E eintritt
$\Delta(E) = h(E) - P(E)$	absoluter Fehler
$\delta(E) = \left \frac{\Delta(E)}{P(E)} \right $	relativer Fehler

Die Berechnung dauert für große n einige Minuten!

2. Modellierung

Ein Verwaltungsprogramm für eine Notfallambulanz ist zu modellieren. Dabei sind die folgende Szenarien einzuplanen:

In der Notfallambulanz sind Ärzte verschiedener Fachgebiete eingestellt: Notfallärzte, Internisten, Chirurgen und Assistenzärzte. Jeder Arzt ist durch den Eintrag seines Namens und seiner Arztnummer (LANR) in der Ambulanz registriert.

Zu Arbeitsbeginn meldet sich jeder Arzt an und zu Arbeitsende wieder ab. Beide Termine (Datum, Uhrzeit) werden in das System eingetragen.

Für jeden eingelieferten Patienten wird bei der Anmeldung eine Patientenakte mit Namen, Geburtsdatum, Krankenkasse, Einlieferungstermin (Datum, Uhrzeit) angelegt. Der Patient wird einem Assistenzarzt übergeben.

Der Assistenzarzt übernimmt den neuen Patienten und macht Standarduntersuchungen. Ein Notfallarzt setzt die Behandlung fort. Bei Bedarf wird ein Internist und ein Chirurg für die Weiterbehandlung des Patienten hinzugezogen. Ein Arzt behandelt mehrere Patienten gleichzeitig und entscheidet über die Entlassung eines Patienten.

Während der Behandlung wird ein Protokoll erstellt. Von dem jeweilig tätigen Arzt wird nach einer Maßnahme ein Protokolleintrag durchgeführt, bestehend aus Datum, Uhrzeit, Diagnose, Behandlung, Medikamentation und behandelnder Arzt.

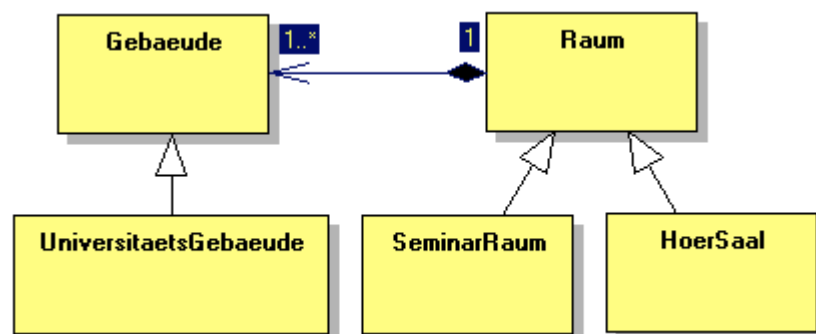
Bei der Entlassung wird der Entlassungstermin (Datum, Uhrzeit) in die Patientenakte eingetragen. Das Behandlungsprotokoll wird dem Patienten für den Hausarzt übergeben. Die Patientenakte verbleibt in der Ambulanz.

- a) Stellen Sie UML - Klassendiagramme mit den nötigen Attributen, ohne Funktionalitäten, für ein Verwaltungsprogramm einer Notfallambulanz zusammen. Achten Sie auf kleine wiederverwendbare Klassen.
- b) Ergänzen Sie Ihre Klassen zu einer Klassenhierarchie: Berücksichtigen Sie Vererbungsbeziehungen der Klassen untereinander. Stellen Sie bestehende Assoziationen, Aggregationen und Kompositionen, einschließlich der Multiplizitäten, dar. Begründen Sie Ihre Wahl.
- c) Erweitern Sie Ihre Klassenhierarchie um die folgenden Funktionalitäten:
 - i. Ein neuer Arzt wird in der Ambulanz eingestellt.
 - ii. Ein Arzt scheidet aus der Ambulanz aus.
 - iii. Ein Arzt beginnt seine Schicht.
 - iv. Ein Arzt beendet seine Schicht.
- d) Erweitern Sie Ihre Klassenhierarchie um die folgenden Funktionalitäten:
 - i. Ein Patient wird eingeliefert.
 - ii. Ein Arzt übernimmt den Patienten und behandelt ihn.
 - iii. Ein Protokolleintrag wird erstellt und in das Behandlungsprotokoll eingetragen.
 - iv. Der behandelnde Arzt entscheidet, ob die Behandlung durch einen anderen Arzt fortgesetzt wird oder die Behandlung abgeschlossen ist und der Patient entlassen wird.
- e) Welche der Collection-Klassen würden Sie zur Speicherung welcher Attribute Ihrer Klassen als Datenstruktur einsetzen? Geben Sie diese an und begründen Sie ihre Auswahl.

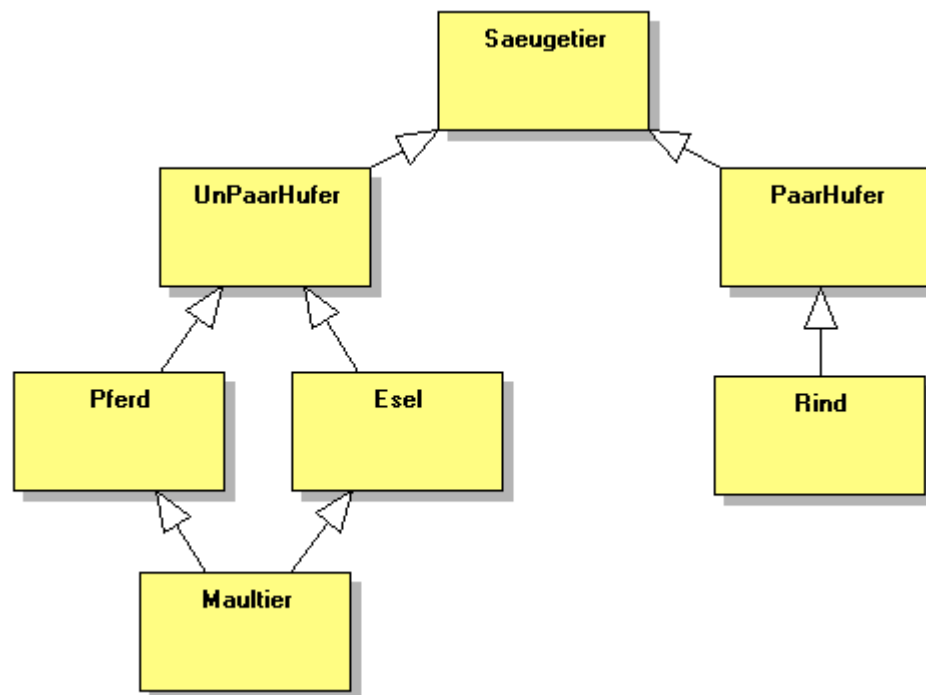
3. Beziehungen zwischen Klassen

Gegeben seien folgende Ausschnitte aus einem Klassendiagramm für eine **java**-Applikation. Suchen Sie nach strukturellen oder inhaltlichen Fehlern. Geben Sie diese an und begründen Sie, warum es sich um einen Fehler handeln muss.

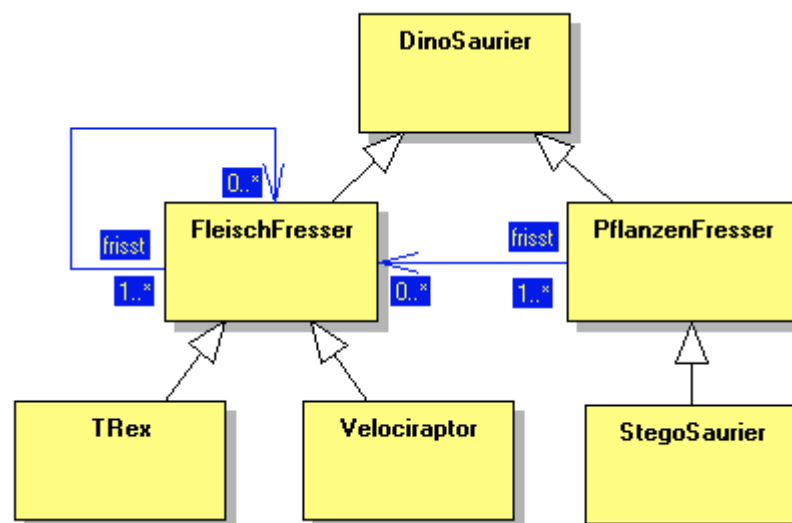
- a) i.



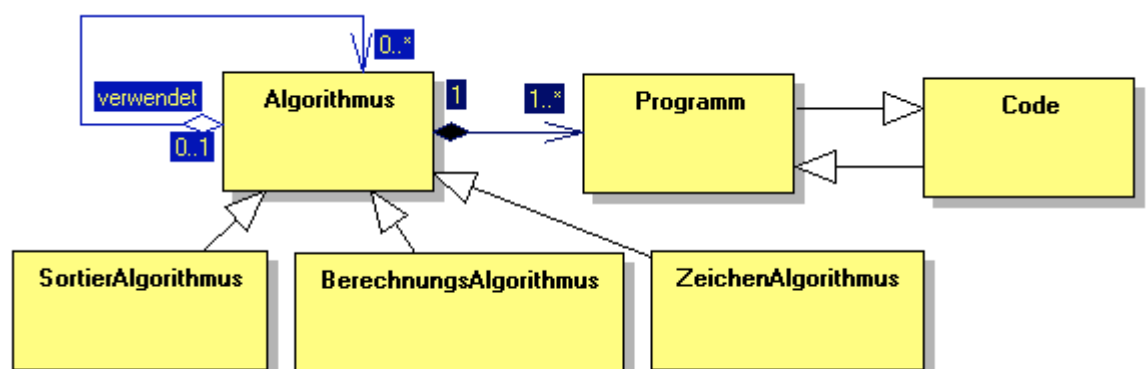
ii.



b) i.



ii.



Hinweis: Ein Programm ist ein codierter Algorithmus.