

## Übungen zu Einführung in die Informatik

*Hinweis: Nur für Diplom-BWL-Studenten. Durch erfolgreiches Bearbeiten dieses und eines weiteren, noch bekannt-zugebenden Übungsblattes können zwei zusätzliche Leistungspunkte erworben werden.*

*Abgabe: 10. 12. 2007, 12.00 Uhr*

### Aufgabe 7-1

### Denkspiel

(10 Punkte)

In dieser Aufgabe soll ein kleines Spiel programmiert werden, das der Mensch gegen den Computer spielen kann.

Die Regeln sind sehr einfach: Gegeben ist ein Haufen mit Streichhölzern. Die zwei Spieler ziehen abwechselnd Streichhölzer und zwar *mindestens 1 aber höchstens 3* von dem Haufen. Wer das letzte Streichholz zieht hat *verloren*.

Ein Beispielverlauf des Spiels:

```
java Game 12
```

```
Ich beginne das Spiel...
||||| ||||| ||
Noch 12 Streichhölzer
Ich nehme 3 Strichhölzer
||||| ||||
Noch 9 Streichhölzer
Wieviele nimmst Du?(1-3) 3
Ich nehme 1 Strichhölzer
|||||
Noch 5 Streichhölzer
Wieviele nimmst Du?(1-3) 1
Ich nehme 3 Strichhölzer
|
Noch 1 Streichhölzer
Wieviele nimmst Du?(1-1) 1
Du hast verloren! :-)
```

- Programmieren Sie eine Klasse `Heap`, die einen Streichhölzerhaufen repräsentiert und ein Attribut `private int numSticks` besitzt. Implementieren Sie eine Getter- und eine Setter-Methode für `numSticks`.
- Programmieren Sie einen Konstruktor `Heap(int n)`, der ein Spiel mit `n` Streichhölzern erzeugt.
- Erweitern Sie die Klasse `Heap` um eine Methode `public void take(int n)`, mit der `n` Streichhölzer vom Haufen genommen werden können.
- Erweitern Sie die Klasse `Heap` um eine Methode `public String toString()`, die eine String-Repräsentation des Heaps und damit der aktuell auf dem Haufen liegenden Streichhölzer erzeugt und zurückgibt. Die Repräsentation könnte dabei so aussehen (für 12 Streichhölzer): `||||| ||||| ||`.
- Laden Sie die Klasse `Game` von unserer Web-Seite herunter und implementieren Sie in der Klasse die folgenden Methoden:
  - `public static int hisMove(int max)`, die (mit Hilfe der Klasse `Eingabe`, die Sie aus Übungsblatt 1 kennen) den Benutzer auffordert, eine Anzahl zwischen 1 und `max` an Streichhölzern zu nehmen. Nur `int`-Werte aus diesem Bereich werden angenommen.
  - Die Methode `public static int myRandomMove(int size)` implementiert einen sehr einfachen Computergegner. Mit dieser Methode zieht der Computer eine `int`-Zahl zwischen 1 und 3 zufallsgeneriert, die jedoch nicht größer als `size` ist. (**Hinweis:** `Math.random()`)
  - Unter welcher Voraussetzung gibt es für das Spiel eine klare Gewinnstrategie, die der Computer (oder auch der Spieler) verfolgen kann, so dass er immer gewinnt, wenn er beginnt. Wann gibt es eine solche Strategie?
  - Die Methode `public static int myIntelligentMove(int size)` gibt für `size` Streichhölzer, die verblieben sind, die Zahl der Streichhölzer zurück, die der Computer nehmen soll, wenn er diese Gewinnstrategie verfolgt. Falls es nicht möglich ist, die Gewinnstrategie zu verfolgen, soll der Computer eine beliebige (jedoch zulässige) Anzahl an Streichhölzern nehmen.

f) Modellieren Sie die Klasse **Heap** in UML.

g) Geben Sie den Speicherzustand zu dem Zeitpunkt graphisch an, wenn nach dem Szenario

1. Am Anfang befinden sich 12 Streichhölzer im Haufen.
2. Der Computer fängt mit dem Spiel an, und nimmt 3 Streichhölzer.
3. Der Spieler gibt an, er möchte 3 Streichhölzer vom Haufen nehmen.

die Methode **take** der Klasse **Heap** (in der mit (\*) markierten Zeile in der Klasse **Game**) aufgerufen wird.