2. Aufgabenblatt für das Vertiefungstutorium

Die Türme von Hanoi

Spielprinzip

„Das Spiel besteht aus drei Stäben A, B und C, auf die mehrere gelochte Scheiben gelegt werden, alle verschieden groß. Zu Beginn liegen alle Scheiben auf Stab A, der Größe nach geordnet, mit der größten Scheibe unten und der kleinsten oben. Ziel des Spiels ist es, den kompletten Scheiben-Stapel von A nach C zu versetzen.
Bei jedem Zug darf die oberste Scheibe eines beliebigen Stabes auf einen der beiden anderen Stäbe gelegt werden, vorausgesetzt, dort liegt nicht schon eine kleinere Scheibe. Folglich sind zu jedem Zeitpunkt des Spieles die Scheiben auf jedem Feld der Größe nach geordnet.“

Aufgabenstellung

Im Folgenden werden wir da Spiel nachbauen. Hierfür werden wir zunächst die Stäbe als Stacks modellieren. Stacks sind ähnlich wie Listen, unterstützten jedoch nur

`push(x)` Das Objekt x oben auf den Stack legen

`pop()` Das oberste Objekt vom Stack entfernen

Genauer gesagt werden wir Stacks von Ganzzahlen benutzen, wobei die Zahl die Größe einer Scheibe repräsentiert.

1. Entwerfen sie eine Klasse `IntegerStack`, sowie eventuell benötigte Hilfsklassen, so mit Attributen, dass es ihnen möglich ist einen Stack darzustellen und die Methoden zu implementieren.
2. Fügen sie der Klasse `IntegerStack` eine Methode `public void push(int value)` hinzu um Zahlen auf den Stack legen zu können.
3. Um die Klasse visualisieren zu können, ergänzen sie die Klasse `IntegerStack` um eine Methode `public String toString()`. Die Werte auf dem Stack sollen dabei durch eine entsprechende Anzahl '#' visualisiert werden, also zB 3 durch "###". Die Werte sind jeweils durch einen Zeilenumbruch zu trennen.
4. Testen sie `push` und `toString`
5. Ergänzen sie die Klasse `IntegerStack` um eine Methode `public int pop()` die das oberste Element vom Stack entfernt und zuückgibt.
6. Testen sie, dass ihre Methoden sich bei beliebigen Kombinationen von `push`, `pop` und `toString` richtig verhalten
7. Was haben Stacks mit Listen zu tun?
8. Ergänzen sie `IntegerStack` um die Methoden `public int size()`, die die Anzahl der Elemente des Stacks zurückgibt sowie einer Methode `public boolean isEmpty()`, die genau dann `true` zurück gibt, wenn der Stack leer ist.

Fertig!