

Modul Verteilte Software WS2022/23

Aufgaben 2

1.

Verändern Sie das Programm der Aufgabe 1.2 wie folgt:

- a) Erstellen Sie für die Klassen Lumenus und Calorus die ToString -Methode
- b) In der main-Methode ist zum Testen der Klassen ein Array aus Objekten der Klassen Lumenus und Calorus zu erstellen. Geben Sie alle Daten der im Array gespeicherten Objekte aus.
- c) Speichern Sie die Objekte der Klassen Lumenus und Calorus statt in einem Array in einer generischen Liste.

2.

Zu erstellen ist ein Programm, das den Wissenschaftlern ermöglicht, die Merkmale einer infolge der Technoevolution entstandenen Spezi „Helmtiger“ im Vergleich zum bereits bekannten Tiger zu untersuchen.

Erstellen Sie dazu die Klasse **Tiger**, die die Angaben zu

- Registriernummer,
- Schädelumfang (reelle Zahl) und
- Schädelfestigkeit (ganze Zahl)

enthält.

Erstellen Sie weiterhin get-Methoden, die einen lesenden Zugriff auf die Attribute erlauben, die toString-Methode und zum Initialisieren der Felder einen sinnvollen Konstruktor.

Die ebenfalls zu erstellende Klasse **Helm** erhält die Angabe

- zum inneren Durchmesser und ebenfalls
- die Festigkeit.

Ermöglichen Sie in der Klasse den lesenden Zugriff auf die Felder.

Die Klasse **Helmtiger** ist von der Klasse Tiger abzuleiten.

In der Klasse ist

- ein weiteres Feld als Objekt der Klasse Helm zu definieren.

Zum Zugriff auf das Feld von außerhalb der Klasse wird außer der get-Methode die set-Methode benötigt, die beim Zuweisen des Helms sicherstellt, dass „der Helm dem Tiger passt“, d.h. dass der Helmdurchmesser größer als der aus dem Schädelumfang berechnete Wert ist, anderenfalls ist dem Helm-Feld der Wert null zuzuweisen.

Erstellen Sie in den Klassen Helm und Helmtiger ebenfalls die toString-Methoden.

In den Klassen Tiger und Helmtiger ist eine weitere Methode zu erstellen, die die gesamte Schädelfestigkeit berechnet und zurückgibt, beim „behelmteten“ Tigers als Summe der Festigkeiten des Tigers und des Helms.

In der main-Methode einer weiteren Klasse ist ein Array aus Objekten der Klassen Tiger und Helmtiger zu erstellen. Geben Sie alle Daten der im Feld gespeicherten Objekte inklusive Gesamtfestigkeit aus.

3.

Ein Mobile ist ein Zimmerschmuck aus ausbalancierten Stäbchen und Fäden, an die kleine Gegenstände geknüpft sind. Die Sterne sind unterschiedlich schwer (w = Gewicht). Die Gewichte der Fäden und Stäbchen sind gegenüber den Sternen zu vernachlässigen.

Ein Beispiel für ein Mobile mit Sternen ist auf der Abbildung 1 zu sehen.

Erstellen Sie zur Berechnung eines Mobiles **Interface Mobile** und Klassen **Stern** und **Stab**. Zu Testzwecken soll das oben skizzierte Mobile aufgebaut, ausbalanciert und dann ausgegeben werden. Dabei sollten sich die in der Skizze angegebenen Knotenpositionen ergeben. Nutzen Sie für die Umsetzung der Aufgabe das Klassendiagramm in der Abbildung 2.

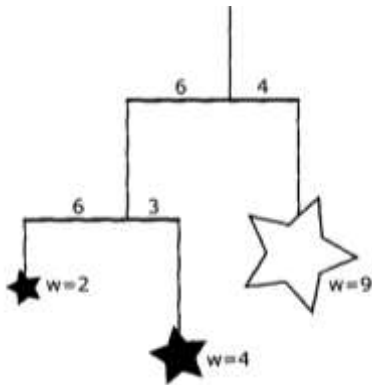


Abbildung 1

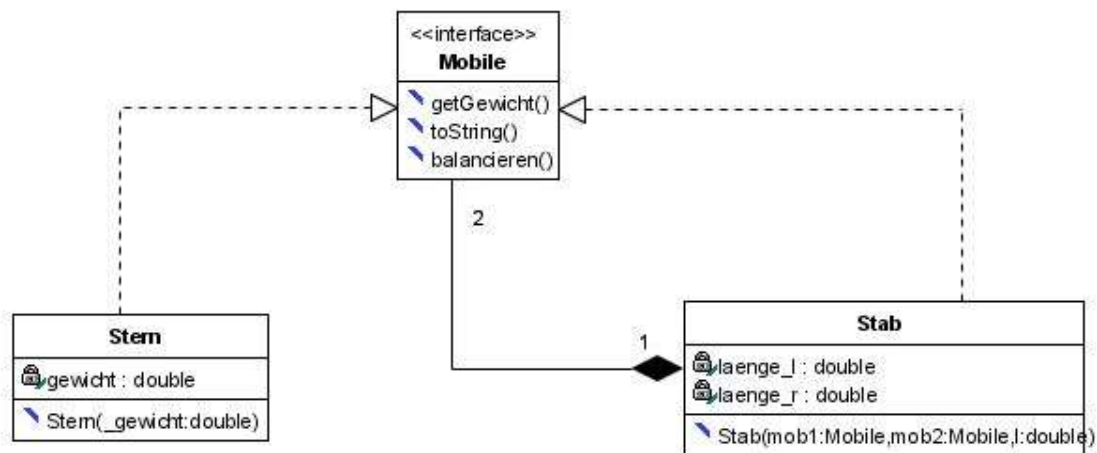


Abbildung 2

Hinweise:

- Die Methode balancieren balanciert ein Mobile aus.
- Jeder Stern hat ein Gewicht (Typ double), das an Konstruktor übergeben wird. Das Gewicht muss positiv sein, sonst wird eine **IllegalArgumentException** geworfen. Ein einzelner Stern ist immer ausbalanciert.
- Die Klasse Stab repräsentiert ein Stäbchen im Mobile. Der Konstruktor erwartet zwei andere Mobiles, die an die Enden des Stäbchens geknüpft sind, und die Länge des Stäbchens. Irgendwo zwischen den beiden Enden des Stäbchens ist der Knoten, an dem es selbst hängt. Die Position des Knotens ist zunächst noch unbestimmt, balancieren verändert die Positionen des Knotens in der Methode balancieren. Ein Mobile ist ausbalanciert, wenn jedes Stäbchen einzeln

ausbalanciert ist. Ein Stäbchen ist ausbalanciert, wenn das Produkt aus Gewicht und Armlänge auf beiden Seiten des Knotens gleich ist ($\text{gewichtL} \cdot \text{längeL} = \text{gewichtR} \cdot \text{längeR}$). Das oben skizzierte Mobile ist ausbalanciert (für das untere Stäbchen: $2 \cdot 6 = 4 \cdot 3$).

Ausgehend aus Gesamtlänge ($\text{gesamt} = \text{längeL} + \text{längeR}$) kann längeL berechnet werden:

$\text{längeL} = \text{gewichtR} \cdot \text{gesamt} / (\text{gewichtL} + \text{gewichtR})$;

Glitzersterne

Neben den normalen Sternen gibt es noch Glitzersterne. Glitzersterne können durch Aufkleben von Goldperlen verschönert werden. Allerdings steigt dabei das Gewicht. Leiten Sie von Stern eine neue Klasse Glitzerstern ab mit einer Methode dekorieren, die das Gewicht eines Sterns bei jedem Aufruf um 1 erhöht.

Ersetzen Sie im oben gezeigten Mobile den Stern mit Gewicht 4 durch einen Glitzerstern.

Verschönern Sie ihn mit drei Aufrufen von dekorieren und balancieren Sie das Mobile dann neu aus.