

Übungen zur Vorlesung
Objektorientierte Programmierung: Wintersemester 2021/2022

Nr. 9, Abgabe bis 24.01.2022

Aufgabe 9.1: PRNDL

3 Punkte

Implementieren Sie eine Methode `public static boolean isReverse(String a, String b)`. Die Methode `isReverse` soll `true` zurückgeben, wenn `String a` umgekehrt den gleichen `String` ergibt wie `b`. Sie dürfen bei Ihrer Implementierung nicht auf `.equals` oder andere Objektmethoden zurückgreifen. Verwenden Sie lediglich Methoden aus der Klasse `String` und `==` zur Überprüfung.

Schreiben Sie eine hinreichende Anzahl an sinnvollen JUnit-Tests, um Ihre Implementierung zu überprüfen.

Aufgabe 9.2: STRG+F, STRG+V

3 Punkte

Implementieren Sie die Methode `String replace(String source, String search, String replace)`, die alle (exakten) Vorkommen von `search` in `source` mit `replace` ersetzt. Verwenden Sie hierbei nur die Methoden `int indexOf(String)`, `String substring(int, int)`, `int length()` und `int indexOf(String str, int pos)` aus der `String`-Klasse. Die Verwendung von anderen in Java gebotenen Methoden ist nicht gestattet.

Testen Sie Ihre Implementierung mit einer ausreichenden Anzahl an sinnvollen JUnit-Tests.

Aufgabe 9.3: Karl Lagerregal

6 Punkte

Ein Unternehmer möchte die Lagerung seiner Warengüter automatisieren. In dem zur Verfügung stehenden Lagerraum befinden sich die verschiedensten Regalsysteme, die in einzelne Segmente unterteilt sind. Jedes Segment zeichnet sich durch dessen Höhe, Breite, Tiefe und maximale Tragkraft aus. Zudem verfügt jedes Segment über 9 Fächer mit identischen Ausmaßen, in denen jeweils nur eine Kiste verstaut werden darf.

Die einzelnen Warengüter werden in Kisten verpackt und gelagert. Implementieren Sie die Klasse `GoodsCrate`, welche die folgenden Komponenten enthalten soll:

- Vier `public double`-Variablen für die Höhe, Breite und Tiefe einer Kiste, sowie ihr Gewicht. Es soll ein Konstruktor bereitgestellt werden, mit welchem diese vier Variablen über Argumente initialisiert werden können.

1

Implementieren Sie die Klasse `GoodsSegment`, die die folgenden Komponenten enthält:

- Vier `public double`-Variablen für die Höhe, Breite und Tiefe des Warensegments, sowie ihr maximales Traggewicht. Weiterhin wird ein eindimensionales `public Array` benötigt, welches die 9 Fächer des Segmentes repräsentiert, in dem jeweils nur eine Kiste abgestellt werden darf. Es soll ein Konstruktor bereitgestellt werden, mit welchem die vier Variablen für die Kennzahlen über Argumente initialisiert werden können. Dieser Konstruktor soll auch alle weiteren Felder initialisieren.

1

- Implementieren Sie die Methode `void addCrate(GoodsCrate crate)`, welche versucht der übergebenen Kiste ein freies Fach im Segment zuzuweisen. Die Zuweisung soll dann erfolgen, wenn das zulässige maximale Traggewicht des Segmentes nicht überschritten bzw. wenn die Kiste entsprechend ihrer Ausmaße in eines der Fächer eingelagert werden kann. Bei letzterem soll von Ihnen berücksichtigt werden, dass Kisten gedreht werden dürfen, um eine geeignete Stellfläche zu finden. Sollte die Kiste nicht eingelagert werden können, soll eine passende Nachricht auf der Konsole ausgegeben werden.

2

Um die Platzverhältnisse der einzelnen Fächer eines Segmentes effizient zu nutzen, soll von Ihnen in einer Klasse `GoodsShelfSystem`, welche das Array `segmente` vom Typ `GoodsSegment` enthält. Initialisieren Sie das Array mit 5 Segmenten mit beliebigen Maßen (> 9.0) im Konstruktor und implementieren Sie die folgende Methode:

- Die Methode `public boolean findSegmentForCrate(GoodsCrate crate)`, welche den geeignetsten Lagerplatz für eine Kiste in einem Fach eines beliebigen Segments sucht. Dabei soll ein Fach gefunden werden, dessen Kapazität/Ausmaße nahezu denen der Kiste entsprechen. Wenn ein geeignetes Fach gefunden wurde, soll die Kiste eingelagert werden und `true` zurückgegeben werden. Sollte es kein geeignetes Fach geben, soll `false` zurückgegeben werden.

2

