

Dieses Übungsblatt dient der Vorbereitung auf die Klausur. Die Lösungen der Aufgaben sollen nicht abgegeben werden. Die Aufgaben werden in den Übungen der Woche vom 29.5.-2.6.2017 besprochen.

Übungsblatt 5

Aufgabe 1

Entwurfsmuster Kompositum

In einem Speditionsauftrag werden verschiedene Versandgüter transportiert. Zeichnen Sie ein Klassendiagramm für ein **Kompositum**-Muster, das die im Folgenden beschriebenen Beziehungen zwischen den beschriebenen Versandgütern geeignet umsetzt:

- Es kommen die Versandgüter Paket, Versandbox, Karton und Kiste vor.
- Ein Paket kann Versandboxen, Kartons, Kisten und andere Pakete enthalten.
- Eine Versandbox kann Pakete, Kartons, Kisten und andere Versandboxen enthalten.
- Jedes Versandgut besitzt ein Attribut `int gewicht` und ein Attribut `boolean brennbar`.

Für das oben modellierte Kompositum-Muster sollen vier Methoden angelegt werden. Skizzieren Sie deren Aufbau für jede von Ihnen vorgesehene Klasse.

Die Methode `int gewicht()` soll das Gesamtgewicht für einen mit dem Kompositum-Muster modellierten Speditionsauftrag liefern.

Die Methode `int uebergewicht(int limit)` soll für einen mit dem Kompositum-Muster modellierten Speditionsauftrag die Anzahl der Kisten liefern, deren `gewicht` über `limit` liegt.

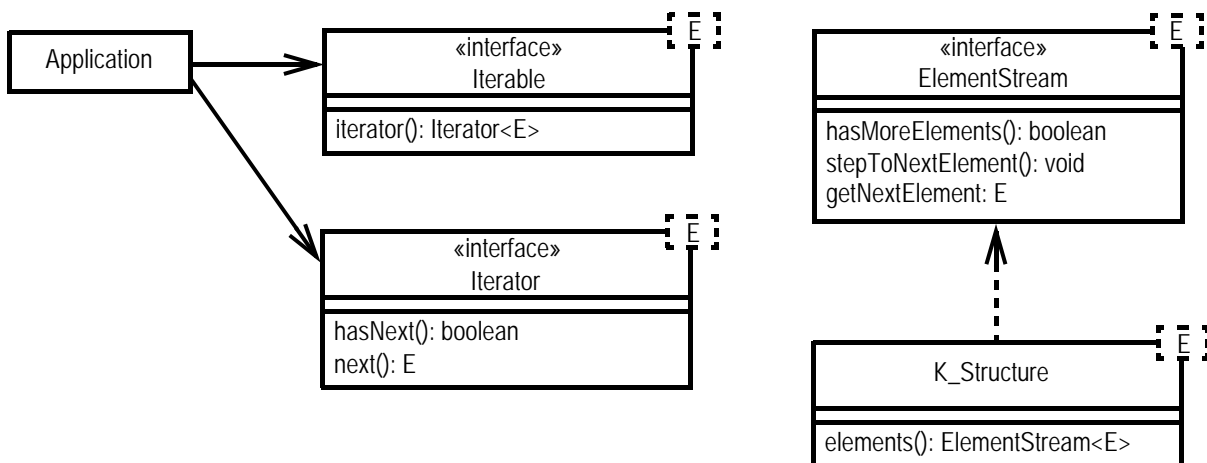
Die Methode `int anzahlBoxen()` soll für einen mit dem Kompositum-Muster modellierten Speditionsauftrag die Anzahl der enthaltenen Versandboxen liefern.

Die Methode `int gefahr()` angelegt werden. Die Methode `gefahr` soll für einen mit dem Kompositum-Muster modellierten Speditionsauftrag das Gewicht aller brennbaren Kartons und Kisten liefern.

Aufgabe 2

Entwurfsmuster – Adapter

Eine Anwendung soll mit Iteratoren arbeiten. Zugleich soll sie eine vorhandene, sehr umfangreiche und sehr kompliziert umgesetzte Datenstruktur `K_Structure` nutzen. `K_Structure` ermöglicht es, über den Aufruf der Methode `elements()` einen `ElementStream` zu erzeugen, der eine zu `Iterator` vergleichbare Funktionalität besitzt. Geben Sie an, wie Adapter eine geeignete Verbindung zwischen der Anwendung und `K_Structure` ermöglichen könnten.



Aufgabe 3

Entwurfsmuster – Iterator/Strategie

Implementieren Sie eine Methode `test` mit der unten angegebenen Signatur, die überprüft, ob es ein Objekt gibt, das sowohl in der Datenstruktur `s1` als auch in der Datenstruktur `s2` vorkommt. Der Vergleich soll mit einem geeigneten, als Argument übergebenen `Checker`-Objekt vorgenommen werden.

```
<E> boolean test(Iterable<E> s1, Iterable<E> s2, Checker<E> c)
```

```
interface Checker<T> {
    boolean isEqual(T o1, T o2);
}
```