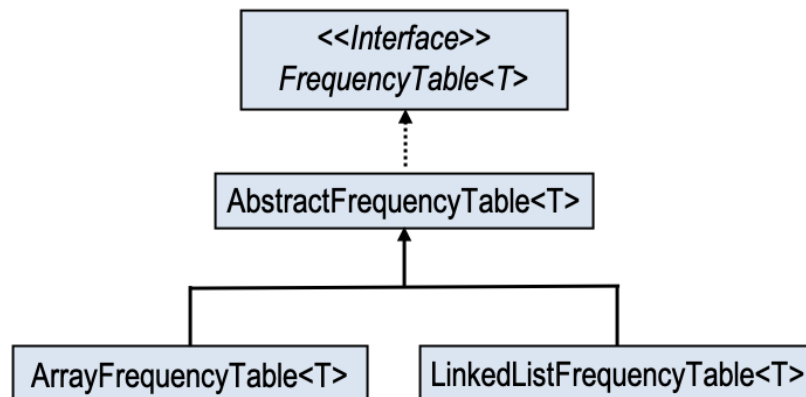


Aufgabenblatt 4

Teil 1

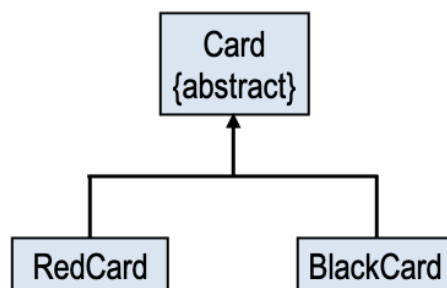
Ändern Sie die in der Aufgabe 1 und 2 realisierten Typen in generische Typen um. Anstatt die Häufigkeiten von Strings zu verwalten, soll es jetzt möglich sein, die Häufigkeiten von Elementen eines beliebigen Typs zu verwalten. Die Klasse `Word` aus Aufgabe 1 und 2 zur Speicherung von Wort-Häufigkeits-Paaren soll nun sinnvollerweise in eine generische Klasse `Element<T>` (Element-Häufigkeits-Paar) geändert werden. Berücksichtigen Sie bei der Implementierung von `ArrayFrequencyTable<T>` insbesondere auch die Folie 4-32.

Testen Sie Ihre Klasse, indem Sie die beiden bisherigen Testklassen entsprechend anpassen.



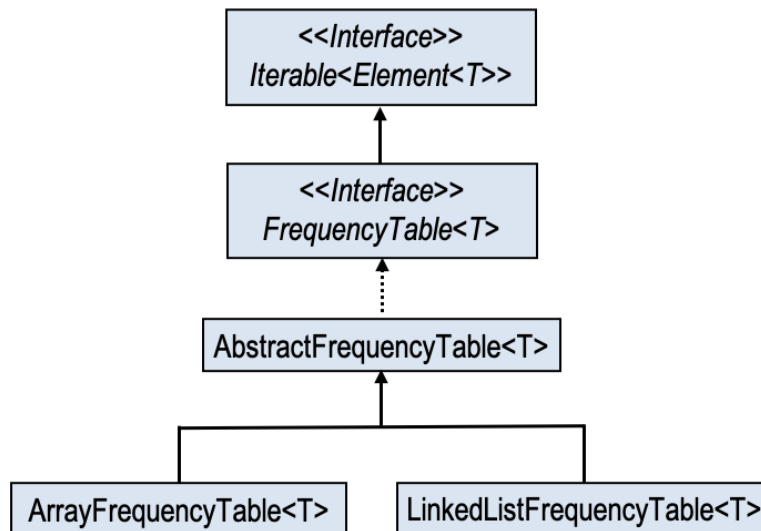
Teil 2

Das Interface `FrequencyTable<T>` soll möglichst flexibel sein. Dazu soll geprüft werden, ob das PECS-Prinzip eingehalten wird. Um das zu überprüfen, sind die Klassen `Card`, `RedCard`, `BlackCard` und `CardFrequencyTable_Test` gegeben. Lassen Sie den Test in `CardFrequencyTable_Test` laufen und korrigieren Sie gegebenenfalls die generischen Typen aus Aufgabenteil 1.



Teil 3

Das Interface `FrequencyTable<T>` soll nun das Interface `Iterable<Element<T>>` erweitern.



Erweitern Sie `ArrayFrequencyTable<T>` und `LinkedListFrequencyTable<T>` jeweils um ein Iteratorkonzept. Die `remove()`-Methode muss nicht unterstützt werden.

Ändern Sie nun die `for`-Schleifen der Methoden `addAll` und `toString` in der abstrakten Klasse `AbstractFrequencyTable<T>` zu foreach-Schleifen ab, d.h. auf Basis der neuen Iteratoren. Bei der `foreach`-Schleife kann es hilfreich sein, den Elementtyp mit Hilfe des Schlüsselworts `var` vom Compiler herleiten zu lassen.

```

public void addAll(FrequencyTable< ... > tab) {
    for (var x : tab)
        this.add(x.getElement(), x.getFrequency());
}

```

Beachte

Verwenden Sie möglichst keine *Raw Types*.

In IntelliJ können Sie über *Preferences* → *Editor* → *Inspections* in das Suchfenster *raw type* eingeben und entsprechende Warnhinweise aktivieren.