

Blatt 10

Punktetabelle

1	2	Σ
6	8	14

Betrachten Sie folgendes Interface `Multiset` zur Darstellung von Multimengen.

```
public interface Multiset<T> extends Iterable<T> {
    public void add(T elem);
    public void union(Multiset<? extends T> other);
    public void subtract(Multiset<?> other);
    public void intersect(Multiset<?> other);
    public void removeOne(Object o);
    public void removeAll(Object o);
    public boolean contains(Object o);
    public boolean subset(Multiset<?> other);
    public int frequency(Object o);
    public int cardinality();
}
```

Multimengen sind wie Mengen, wobei jedoch Duplikate erlaubt sind. Z.B. gilt $\{2, 2, 3\} = \{3, 2, 2\} \neq \{2, 3\}$. Die entsprechenden Methoden sind wie folgt spezifiziert:

- `add` fügt ein Element in das Set ein
- `union` zum Berechnen der Vereinigung zweier Sets, i.e., $A := A \cup B = \{x \mid x \in A \vee x \in B\}$. Z.B. gilt $\{1, 2\} \cup \{2, 3\} = \{1, 2, 2, 3\}$.
- `subtract` zum Berechnen der Differenz zweier Sets, i.e., $A := A \setminus B = \{x \mid x \in A \wedge x \notin B\}$. Z.B. gilt $\{1, 1, 2, 2\} \setminus \{1, 2, 3\} = \{1, 2\}$.
- `intersect` zum Berechnen der Schnittmenge zweier Sets, i.e., $A := A \cap B = \{x \mid x \in A \wedge x \in B\}$. Z.B. gilt $\{1, 1, 2, 2\} \cap \{1, 2, 2, 3\} = \{1, 2, 2\}$.
- `removeOne` zum Entfernen eines Elements aus dem Set
- `removeAll` zum Entfernen aller Elemente eines Sets
- `contains` zum Überprüfen, ob ein Set ein bestimmtes Elements enthält
- `subset` überprüft, ob das aktuelle Set eine Teilmenge des anderen ist. Z.B. ist $\{1, 2\} \subseteq \{1, 1, 2, 3\}$, aber $\{1, 1, 2\} \not\subseteq \{1, 2\}$.
- `cardinality` zum Berechnen der Größe des Sets. Z.B. ist $|\{1, 2, 2, 3\}| = 4$.
- `frequency` zum Berechnen der Häufigkeit eines Elements in einem Set

1 Unit Tests

Erstellen Sie eine TestSuite für das Interface. Achten Sie darauf, sinnvolle Testfälle zu erstellen.

2 Implementierung

Implementieren Sie das Interface `Multiset`. In Ihrer Implementierung dürfen Sie beliebig Klassen aus `java.util` verwenden. Beachten Sie jedoch unter anderem die folgenden Punkte:

- Sie entscheiden, ob Sie `null`-Werte unterstützen oder nicht.

- Sie entscheiden, ob zusätzliche Eigenschaften wie `Comparable` verlangt werden.
- Die Implementierung sollte bei Aufrufen wie `set.union(set)`, bei denen beide beteiligten Mengen identisch sind, weder eine Ausnahme liefern, noch zur Nicht-Terminierung führen.
- Der Iterator sollte `remove` unterstützen und fail-fast sein.