

Aufgaben zur Vorbereitung auf das Testat 8

Als Vorbereitung auf das Testat 8 solltest Du unbedingt diese Aufgaben bearbeiten.

Lambda-Ausdrücke

Die Grundlage für diese Aufgaben bildet die Klasse `IntIntPairs`, die eine vorgegebene maximale Zahl von Paaren von `int`-Werten verwaltet. Zwei zusammengehörende `int`-Werte werden unter jeweils dem gleichen Index in den beiden Feldern `keys` und `values` abgelegt. Die abgelegte Paare werden nur dann berücksichtigt, wenn zugleich in einem dritten boolean-Feld `valids` der Wert `true` für diesen Index eingetragen ist. Das Löschen eines Paares in der Klasse `IntIntPairs` erfolgt also durch Setzen des Werts von `valids` am entsprechenden Index auf `false`.

Mache Dich zunächst mit der Implementierung der Klasse `IntIntPairs` vertraut.

Erweitere die in der Klasse `Testumgebung` vorgegebene Testmethode schrittweise so, dass die Korrektheit der nachfolgend beschriebenen Lambda-Ausdrücke und Methoden überprüft wird.

1 - einfache Nutzung von Lambda-Ausdrücken

Wähle jeweils eine Methode der Klasse `IntIntPairs` aus, der Du zur Lösung geeignete Lambda-Ausdrücke als Argumente übergibst.

- Ermittle für alle gültigen Einträge des Wertes 0 in `keys` die Summe der zugehörigen Werte in `values`.
Die Lösung dieser Aufgabe findest Du als Beispiel in der Testumgebung.
- Bestimme die Anzahl der gültigen geraden Werte in `keys`.
- Lösche alle gültigen Paare, deren Wert in `keys` gleich dem Wert 6 ist.
- Erhöhe jeden gültigen Wert in `keys` um den Wert 10.
- Erstelle ein `IntIntPairs`-Objekt, das Kopien aller gültigen Paare enthält, deren Wert in `values` ganzzahlig durch 3 teilbar ist.
- Bestimme die Anzahl der gültigen Werte in `values`, die größer als 10 sind.
- Erhöhe die Werte in `keys` um den Wert 5, deren zugehöriger Wert in `values` größer als 3 ist.
- Lösche alle gültigen Paare, die einen negativen Wert in `values` besitzen.
- Verdopple jeden gültigen Wert in `values`.
- Erstelle eine Kopie des `IntIntPairs`-Objekts, das alle gültigen Paare enthält.
- Lösche alle gültigen Paare, deren beide Werte identisch sind.
- Bilde die Summe aller gültigen Werte in `values`.
- Erstelle ein `IntIntPairs`-Objekt, das Kopien aller gültigen Paare enthält, deren Wert in `values` größer als der Wert 5 ist.

2 - kombinierte Nutzung von Lambda-Ausdrücken

Wähle jeweils geeignete Methoden der Klasse `IntIntPairs` aus, übergib diesen geeignete Lambda-Ausdrücke als Argumente und kombiniere die Aufrufe passend in einer statischen Methode.

- Implementiere eine Methode `sumUp(IntIntPairs pairs)`, die für die gültigen Paare in `pairs`, deren Wert in `keys` gleich 0 ist, die zugehörigen Werte aus `values` aufsummiert.
Die Lösung dieser Aufgabe finden Sie als Beispiel in der Testumgebung.
- Implementiere eine Methode `addNToValue(IntIntPairs pairs, int n)`, die für die gültigen Paare in `pairs` die Werte in `values` um den Wert `n` erhöht.
- Implementiere eine Methode `uniqueKey(IntIntPairs pairs, int n)`, die `true` zurückgibt, wenn in allen gültigen Paaren in `pairs` der Wert von `n` in `keys` genau einmal auftritt.
- Implementiere eine Methode `doubleGreaterN(IntIntPairs pairs, int n)`, die für jedes gültigen Paar (k, v) in `pairs`, dessen Wert `k` in `keys` größer als der Wert `n` ist, zu `pairs` ein weiteres Paar $(k, 2*v)$ hinzufügt, sofern in `pairs` noch Einträge möglich sind.
- Implementiere eine Methode `IntIntPairs concat(IntIntPairs first, IntIntPairs second)`, die ein neues `IntIntPairs`-Objekt erzeugt, das alle gültigen Paare von `first` und `second` enthält. `first` und `second` sollen nicht geändert werden.