

## Praktikumstutorium – Blatt 8

Als Vorbereitung auf das Testat 8 sollten Sie unbedingt diese Aufgaben bearbeiten.

Sofern dabei Schwierigkeiten auftreten, sollten Sie unbedingt das **Tutorium des Programmierpraktikums** besuchen.

Tutoriumszeiten in Raum OH 12 / 4.030

Montag	Dienstag	Mittwoch	Donnerstag	Freitag
12.00 – 14.00 Uhr	10.00 – 16.00 Uhr	10.00 – 14.00 Uhr	10.00 – 14.00 Uhr	Testat

### 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 von Paaren in der Klasse `IntIntPairs` erfolgt also durch Setzen des entsprechenden Wertes von `valids` auf `false`.

Machen Sie sich zunächst mit der Implementierung der Klasse `IntIntPairs` vertraut, die Sie aus dem Ordner *Praktikumsaufgaben* im **moodle**-Kursbereich herunterladen können.

Erweitern Sie 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

**Wählen jeweils eine Methode der Klasse `IntIntPairs` aus, der Sie zur Lösung geeignete Lambda-Ausdrücke als Argumente übergeben.**

- Ermitteln Sie für alle gültigen Einträge des Wertes 0 in `keys` die Summe der zugehörigen Werte in `values`.  
Die Lösung dieser Aufgabe finden Sie als Beispiel in der Testumgebung.
- Bestimmen Sie die Anzahl der gültigen geraden Werte in `keys`.
- Löschen Sie alle gültigen Paare, deren Wert in `keys` gleich dem Wert 6 ist.
- Erhöhen Sie jeden gültigen Wert in `keys` um den Wert 10.
- Erstellen Sie ein `IntIntPairs`-Objekt, das Kopien aller gültigen Paare enthält, deren Wert in `values` ganzzahlig durch 3 teilbar ist.
- Bestimmen Sie die Anzahl der gültigen Werte in `values`, die größer als 10 sind.
- Erhöhen Sie die Werte von in `keys` um den Wert 5, deren zugehöriger Wert in `values` größer als 3 ist.
- Löschen Sie alle gültigen Paare, die einen negativen Wert in `values` besitzen.
- Verdoppeln Sie jeden gültigen Wert in `values`.
- Erstellen Sie eine Kopie des `IntIntPairs`-Objekts, das alle gültigen Paare enthält.
- Löschen Sie alle gültigen Paare, deren beide Werte identisch sind.
- Bilden Sie die Summe aller gültigen Werte in `values`.
- Erstellen Sie 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

**Wählen Sie jeweils geeignete Methoden der Klasse `IntIntPairs` aus, übergeben Sie diesen geeignete Lambda-Ausdrücke als Argumente und kombinieren Sie die Aufrufe passend in einer statischen Methode.**

- Implementieren Sie 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.

- b) Implementieren Sie eine Methode `addNToValue(IntIntPairs pairs, int n)`, die für die gültigen Paare in `pairs` die Werte in `values` um den Wert `n` erhöht.
- c) Implementieren Sie 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.
- d) Implementieren Sie 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.
- e) Implementieren Sie eine Methode `IntIntPairs concat(IntIntPairs first, IntIntPairs second)`, die ein neues `IntIntPairs`-Objekt erzeugt, dass alle gültigen Paare von `first` und `second` enthält. `first` und `second` sollen nicht geändert werden.