



W08H01 - Simple Generics

Hausaufgabe

Easy

Points: 8 of 8

Assessment: automatic ?

Submission due: 7 months ago

Complaint due: 7 months ago

100% (8 months ago) GRADED

Recent results:



Show all results

Tasks:

Simple Generics

Die Forschuine der PUM arbeiten aktuell an ihrer digitalen Datenverarbeitung. Dazu brauchen sie noch einige Utility Funktionen, die ihnen das Leben leichter machen sollen. Da sie sich mit Generics noch nicht so wohl fühlen wie du, unterstützt du sie bei der Entwicklung.

Hinweis: in der ganzen Aufgabe kannst du davon ausgehen, dass die getesteten Inputs niemals `null` sind.

Tipp: Nutze dein Wissen über Collections aus der Vorlesung/Zentralübung. Auch ein Blick in die Java Docs kann sehr hilfreich sein. Was nicht verboten ist, ist erlaubt. 🙌

• **toString** 1 of 1 tests passing

Um beim Debuggen eine einheitliche Repräsentation für Collections zu haben, benötigen die Pingine die Methode `toString(Collection)`. Der Methode wird eine `Collection` übergeben. Zurückgegeben werden soll ein `String` um diese Collection zu visualisieren. Dabei muss sich an folgende Muster gehalten werden:

1. die `Collection` selbst wird durch zwei geschweifte Klammern dargestellt: `{<Inhalt der Collection>}`.
2. die Elemente der `Collection` sollen zwischen den geschweiften Klammern stehen und jeweils von folgendem String getrennt werden: `,` .
3. die Reihenfolge, in der die Elemente im String dargestellt werden sollen, soll der Reihenfolge entsprechen, die durch den `Iterator` der gegebenen `Collection` diktiert wird.
4. Hier ein Beispiel für die `List<?> l = Arrays.asList({1, 2, 3, 5, 4})`: `{1, 2, 3, 5, 4}`

• **toIntArray** 1 of 1 tests passing

`Collections` können sehr einfach zu Arrays gecastet werden. Wollen wir jedoch aus einer `Collection<Integer>` ein primitives `int[]` casten, ist das ohne Weiteres nicht direkt machbar. Implementiere die Methode `toIntArray(Collection<Integer>)`, die das passende `int` Array zurückgeben soll. Das Array soll die passende Länge haben (nicht länger sein als die `Collection`). Die Reihenfolge der Elemente im Array soll der Reihenfolge entsprechen, die durch den `Iterator` der gegebenen `Collection` diktiert wird.

• **specialSort** 1 of 1 tests passing

Die Methode `specialSort` soll ein sortiertes Array der gegebenen `Collection` zurückgeben. Ein Array eines Generics zu erstellen, ist in Java nicht so einfach möglich (probiere es am besten einmal aus und überlege dir warum das nicht geht). Um dennoch solch ein Array erstellen zu können, findest du im Template direkt über der Methode dieser Teilaufgabe die bereits implementierte Methode `generateGenericArray(Class<T> clazz, int length)`. Benutze diese Methode um ein Array mit passender Länge zu erstellen. Die gegebene `Collection` soll anschließend in diesem Array gespeichert und vor der Rückgabe mit dem übergebenen `Comparator` sortiert werden.

• **intersection** 1 of 1 tests passing

Der Methode wird ein Array aus `Collections` übergeben. Zurückgegeben werden soll eine Collection, die alle Elemente enthält, die in jeder der gegebenen `Collections` enthalten ist. Die gegebenen `Collections` dürfen dabei nicht verändert werden. Für alle möglichen Eingaben soll eine echte/sinnvolle `Collection` (nicht `null`) zurückgeben werden. Was für eine konkrete Implementierung von `Collection` du hierbei für am sinnvollster erachtest, sollst du dir selbst überlegen.

▼ Beispiel zu intersection

```
intersection({}) = {}  
intersection({{1, 2, 3, 4, 5}, {1, 2, 3}, {1, 2, 4, 5}}) = {1, 2}
```

• **getValues** 1 of 1 tests passing

Leider arbeiten die Pinguine immer wieder mit modifizierten `Maps`, deren `values()`-Methode kaputt gehen kann. Die Methode `getValues(Map)` soll diese ersetzen. Sie soll ein `Set`, bestehend aus allen Values, die in der gegebenen `Map` gespeichert werden, zurückgeben. Da die `Map` modifiziert sein

könnte, darfst du `Map.values()` dazu nicht benutzen. Alle anderen Methoden der `Map` dürfen verwendet werden.

Tests

Du bekommst bei dieser Aufgabe von den PublicTests (den Tests, die direkt nach jedem Push ausgeführt werden), zwar kein Feedback darüber, was du falsch gemacht hast, allerdings teilen sie dir nach jedem Push mit, ob du die jeweilige Teilaufgabe korrekt gelöst hast oder nicht. Daher kannst du dir auch sicher sein, dass du alle Punkte bekommst, wenn du alle PublicTests bestehst (ausgenommen du hältst dich nicht an eine der in der Aufgabenstellung geforderten Einschränkungen). Die HiddenTests sind deckungsgleich mit den PublicTests und geben dir nach der Deadline ein Feedback, warum sie fehlgeschlagen sind.

Viel Erfolg!

[Lösungsvorschlag](#)

Tests

Ein Paar Tage nach der Deadline werden diese beiden Links zu der Musterlösung bzw. unseren PublicTests und HiddenTests führen.

Exercise details	
Release date:	Dec 8, 2022 18:30
Submission due:	Jan 8, 2023 18:00
Complaint due:	Jan 15, 2023 18:00

Every student is allowed to complain once per exercise. In total 1000 complaints are possible in this course. You still have **998** complaints left. 