

Übungsblatt 11 – Generics & Containerklassen

Abzugeben: Anfang nächsten Semesters (machen Sie es, weil wir brauchen das in PM)

1. In Übungsblatt 8 haben Sie eine Containerbibliothek angelegt, die auf dem Typ Object (Rückgrat war hier das Feld `Object [] _objs`) arbeitet. Kopieren Sie den gesamten Code in ein neues Verzeichnis und ändern Sie alle Container-Objekte so um, dass Sie auf einer generischen Typvariable `T` statt auf `Object`¹ basieren. Passen Sie dann auch den Code, der die Containerbibliothek testet so an, dass er mit der neuen Typvariable funktioniert und führen Sie dann den Code zum Testen aus.

Vergleichen Sie dann das Endergebnis mit der früheren Containerbibliothek aus Übungsblatt 8.

Wichtiger Hinweis:

Es gibt ein Problem mit dem `Object[]`-Array in `MyAbstractContainer`: Wenn Sie daraus ein Array `T[]` machen, dann benötigen Sie für die Erzeugung des Arrays auch das Klassenobjekt benötigt und sich halten muss. Deshalb empfehle ich das Array selbst weiterhin beim Typ `Object[]` zu belassen und bei den public-Methoden nach außen dann auf den Typ `T` zu casten. Das würde dann ungefähr so aussehen:

```
public abstract class MyAbstractContainer<T> {
    protected Object [] _objs;
    protected int _size;
    ...
    public T get (int index){
        ....
        return (T) _objs[index];
    }
}
```

2. Schreiben Sie eine Klasse `ListTests`. In der Klasse sollen mehrere Listen (`java.util.List`) erzeugt und getestet werden.
 - a. Schreiben Sie eine static Methode mit einer Typvariable `R` (`public <R> void printList(List<R> list)`), die den Inhalt einer übergebenen Liste auf der Konsole ausgibt. Benutzen Sie hierfür auch die neue for-Schleife.
 - b. Schreiben Sie eine Methode `void testStringList()` in der Sie zunächst eine Liste für Strings erzeugen und mehrere Strings hinzufügen.
Machen Sie dann folgende Operationen und geben Sie jeweils danach die Liste mittels der Methode `printList` aus²:
 - Fügen Sie an einem Index in der Mitte einen String ein.
 - Ersetzen Sie einen String mittels der Methode `set(...)`
 - Entfernen Sie mehrere Strings wieder, indem Sie den index des Strings ermitteln und dann `remove` aufrufen.
 - Sortieren Sie die Liste mittels `java.util.Collections`.

¹ Gehen Sie bei der Anpassung langsam vor. Ich würde ein neues Projekt machen und zunächst `MyAbstractContainer` in das Projekt nehmen und anpassen und kompilieren, dann `MyStack` und die Testklassen (kompilieren & testen), dann `MyList` (kompilieren & testen), ...

² Hinweis: Ich empfehle für die einzelnen Unterschritte eigene Untermethoden zu machen. Dann können Sie die Methode dann leichter überblicken.

- Entfernen Sie alle Einträge in der Liste.
 - c. Schreiben Sie nun eine Methode `void testIntList()` in der Sie dieselben Aktionen für den Datentyp `int` durchführen³.
 - d. Nehmen Sie nun eine beliebige Klasse, die Sie bei einer der früheren Übungen entwickelt haben. Schreiben Sie nun eine Methode `void test...List()` in der Sie dieselben Aktionen für den Datentyp dieser gewählten Klasse durchführen.
 - e. Schreiben Sie eine `main`-Methode, die die Methoden `test...List` jeweils zum Durchführen der Tests aufruft. Führen Sie dann das Testprogramm aus und überwachen Sie die Ausgaben.
3. Schreiben Sie eine Klasse `HashMapTests` in der wir `HashMap` testen. Als Schlüsselobjektyp verwenden wir `Strings` für den Wertobjektypen wählen Sie dieselbe Klasse, die Sie in 2c. verwendet haben.
- a. Erzeugen Sie ein Attribut `HashMap<String, IhrKlassenTyp>`.
 - b. Schreiben Sie eine Methode `void printKeys()`, die die in der `HashMap` vorhandenen Schlüssel auf der Konsole ausgibt.
 - c. Schreiben Sie eine Methode `void printValues()`, die die in der `HashMap` vorhandenen Werte auf der Konsole ausgibt.
 - d. Schreiben Sie eine Methode `void printKeysAndValues()`, die die in der `HashMap` vorhandenen Schlüssel-Wert-Paare auf der Konsole ausgibt.
 - e. Schreiben Sie jetzt eine Methode `void testHashMap()` in der Sie folgende Aktionen machen und danach eine der 3 `print`-Methoden ausführen⁴:
 - i. Fügen Sie mehrere Wertobjekte mit einem jeweiligen Stringschlüssel zur Hashmap hinzu.
 - ii. Versuchen Sie ein Wertobjekt mittels Schlüssel wieder zu bekommen.
 - iii. Entfernen Sie ein Wertobjekt wieder.
 - iv. Überprüfen Sie, ob ein bestimmter Schlüssel vorhanden ist.
 - v. Überprüfen Sie, ob die Hashmap ein bestimmtes Wertobjekt enthält.
 - vi. Entfernen Sie alle Einträge aus der Hashmap.
 - f. Schreiben Sie eine `main`-Methode, die die Methode `testHashMap()` zum Durchführen der Tests aufruft. Führen Sie dann das Testprogramm aus und überwachen Sie die Ausgaben.

³ Bei primitiven Datentypen müssen Sie den dazugehörigen Objektypen nehmen und in die Typvariable einsetzen (näheres dazu in den Folien).

⁴ Hinweis: Ich empfehle für die einzelnen Unterschritte eigene Untermethoden zu machen. Dann können Sie die Methode dann leichter überblicken.