

## Pflichtaufgabe 1 - Kontaktliste (Array und Testen)

Dies ist Ihre erste Pflichtaufgabe, die in Moodle *fristgerecht* abgegeben werden muss und *bewertet* wird. Bitte laden Sie als Lösung nur java-Files hoch. Bitte beachten Sie, dass Abgaben, die **mehr als 10 Min.** zu spät hochgeladen werden, leider **nicht berücksichtigt** werden.

Mit dem Anklicken der **Eigenständigkeitserklärung** bestätigen Sie, dass die Lösungen Ihre eigene (Gruppen-)Leistung sind. Bitte beachten Sie, dass (auch teilweise) identische Lösungen von Personen aus fremden Gruppen als **Betrugsversuch** ("Abschreiben") angesehen und mit **0 Punkten** bewertet werden.

### 1.1. Kontaktliste erstellen (4 Punkte)

Erstellen Sie eine Klasse für die Verwaltung Ihrer Kontakte. Erstellen Sie eine Objektklasse für die Kontakte.

Erweitern Sie die Klasse um die Getter- und Setter- Methoden.

Speichern Sie in einem Array mind. 8 Kontakte (einige mit 030-Vorwahl und einige mit anderer Vorwahl) und starten Sie die Initiierung der Objekte und die Speicherung in der Kontaktverwaltung.

### 1.2 Überschreiben der toString()-Methode (1 Punkte)

Überschreiben Sie bitte die toString()-Methode und geben Sie alle Kontakte mit einer Nummerierung aus.

### 1.3 Suchen Sie Einträge mit Berliner-030 Vorwahl (6 Punkte)

Implementieren Sie eine Java-Methode für die Rückgabe alle Berliner-Telefonnummern. Die Rückgabe erfolgt in einem Array, der nur Telefonnummern und nicht die Objekte der Kontakte beinhaltet. Der Array darf nur Daten eines Daten-Typs aufnehmen.

Zur Lösung dieser Aufgabe können weitere Hilfsmethoden und/oder vordefinierte Java-Methoden z.B. aus der Klasse Arrays verwendet werden.

### 1.4 Testen Sie Ihre Lösung (5 Punkte)

Erstellen Sie die Testmethoden für getNumber und setNumber. Und Überprüfen Sie mit mind. zwei Testmethoden (JUnit) die Arrayliste mit den Berliner-Telefonnummern. Schreiben Sie zu jedem Test im Kommentar hin, welchen Fall Sie gerade testen (Normalfall, Grenzfall, best case,...).

### 1.5 Methoden ergänzen (4 Punkte)

Ergänzen Sie die Klasse zur Verwaltung Ihrer Kontakte um eine weitere sinnvolle Methode Ihrer Wahl, wie z.B.

- search() (sucht Telefonnummer oder Name),
- delete() (Löscht einen Kontakt),
- editNumber() (ändert die Telefonnummer),

Implementieren Sie die dazugehörigen Unit Tests (2 Unit Tests). Gehen Sie dabei nach dem Prinzip des Test Driven Development vor.

Hinweis: Achten Sie bei der Implementierung aller Klassen auf einen guten Programmierstil (Kommentare, Formatierung, Namenskonventionen, eigener Konstruktor usw.).

**(Gesamt 20 Punkte)**