

Ausgangssituation: Digitales Impfreister

Zu jeder Person sollen die Impfungen gespeichert werden, die die Person erhalten hat (Reihenfolge der Impfungen spielt keine Rolle). Die Klassen für Person und Impfung (Vaccination) sind bereits vorhanden.

Beispiel mit zwei Personen:

Person	Impfungen
P1	[Covid, Influenza]
P2	[Smallpox]

Die Aufnahme von Smallpox für Person P1 und Covid für Person P2 führt zur folgenden Tabelle:

Person	Impfungen
P1	[Covid, Influenza, Smallpox]
P2	[Smallpox, Covid]

Absrakte Klasse ADigitalVaccinationRegister

```
public abstract class ADigitalVaccinationRegister {  
  
    // Anzahl an gespeicherter Personen (mit mindestens einer Impfung)  
    public abstract int size();  
  
    // Fügt zur Person person die Impfung vaccination hinzu, falls Person  
    // != null und vaccination != null ist. Die Impfung wird nur einmal hinzugefügt  
    // Rückgabe: True falls Impfung hinzugefügt, false sonst  
    // Beachten Sie: Eine Person kann nur einmal enthalten sein  
    public abstract boolean add(Person person, Vaccination vaccination);  
  
    // Anzahl Impfungen, die Person person erhalten hat. Rückgabe 0, falls Person  
    // nicht enthalten ist  
    public abstract int contains(Person person);  
  
    // True: Person hat Impfung vaccination erhalten  
    // False sonst  
    public abstract boolean contains(Person person, Vaccination vaccination);  
  
    // Rückgabe; Collection, die alle Impfungen der Person person enthält  
    // Sind zu Person person keine Impfungen enthalten, wird null zurückgegeben  
    public abstract Collection<Vaccination> get(Person person);  
  
    // Löscht die Impfung vaccination der Person person aus der zugeordneten  
    // Collection  
    // Rückgabe = True falls Impfung erfolgreich entfernt wurde, false sonst  
    public abstract boolean remove(Person person, Vaccination vaccination);  
  
    // Löscht Person person aus dem Register  
    // Rückgabe = Alle Impfungen der Person person  
    // null falls Person nicht enthalten war  
    public abstract Collection<Vaccination> remove(Person person);  
  
    // Wendet consumer.accept() auf alle gespeicherten Impfungen von allen  
    // Personen an  
    public abstract void forEach(Consumer<Collection<Vaccination>> consumer);  
}
```

Vorbereitung

1. IntelliJ
 - a. Projekt Testat-Matrikelnummer erstellen
 - b. Entpacken Sie 2022-SS-Testat-G1-Vaccination_Register_IntelliJ.zip
 - c. Kopieren des Inhaltes des src-Verzeichnis in das src-Verzeichnis des Projektes
 - d. JUnit Version 5.x.y zum Projekt hinzufügen
2. Alternativ: Starten Sie Eclipse mit einem passenden Workspace: Java Version 11 (default)
 - a. Importieren Sie das Projekt 2022-SS-Testat-G1-Vaccination_Register_eclipse.zip (Import existing project)!
3. Erstellen Sie die Klasse **DVRegisterTest#####**, die von **ADigitalVaccinationRegisterTest** erbt
4. Lesen Sie die Hinweise!

Aufgaben

- Benennen Sie das Paket G1_solution in G1_solution##### um
- Implementieren Sie die abstrakten Methoden der Klasse **ADigitalVaccinationRegister**
- ##### = Ihre Matrikelnummer

Vorgehen

- Implementieren Sie die Klasse **DVRegister#####**, die von **ADigitalVaccinationRegister** erbt.
- Passen Sie Ihre Testklasse so an, dass eine Instanz Ihrer Klasse übergeben wird.
 - a. Methode **getInstance()** – Geben Sie eine Instanz ihrer Klasse zurück

Abgabe

- Exportieren Sie Ihr Projekt (Testat-G1-#####) als Archiv in das Verzeichnis C:\InsightFiles
- Vor Abgabe melden Sie sich!

Hinweise

- Sie benötigen keine inneren Klassen
- List und Set erben von Collection
- Verwenden Sie eine Collection, die die Zuordnung von Personen zu deren Impfungen zulässt (eine Person kann eine Liste von Impfungen erhalten haben)

Hinweise zur Benotung (110 Punkte)

- **Es reichen 50 Punkte zum Bestehen!**
- **(Versuchte) Manipulation der gegebenen Klassen/Interfaces: max. 25 Punkte**
- **Kopieren der gegebenen Klassen/Interfaces in ein eigenes Paket: 0 Punkte**
- **Programm nicht lauffähig (Syntaxfehler): max. 25 Punkte**
- **Artefakte aus der anderen Gruppe tauchen auf: 0 Punkte!**
- Sowohl Unit-Tests als auch der Quelltext werden für die Benotung herangezogen
- Unit-Tests dienen nur zur Orientierung!

Collections

Interface/Klasse	Implementierende Klasse	Wichtige Methoden
Map	HashMap	put(k,v), get(k), size(), remove(k), clear() keySet(): Menge der Schlüssel values(): Menge der gespeicherten Werte
Set	HashSet	add(e), contains(e), remove(e), clear()
List	ArrayList	add(e), get(i), clear() add(i,e): Fügt e an Position i ein, vorher wird die Liste ab i nach rechts verschoben, set(i, e): Überschreibt den Eintrag an Position i mit e contains(e), remove(e): true, falls e (einmal) entfernt wurde, sonst false remove(i): Löscht den Eintrag an Position i
Date	-	new Date()// aktuelles Datum
Consumer<T>	-	void accept(T t)
String		String[] split(separator), boolean contains(string), String replace(x,y) // ersetzt x durch y int length(), String toLowerCase(), toUpperCase()
Predicate<T>		boolean test(T t)

Anhang

Die Klassen Person und Vaccination sind im Paket base enthalten. Sie benötigen keine Methoden dieser Klassen!