

Aufgabe B: Buchungssystem für Vorlesungsräume (15 Punkte)

In dieser Aufgabe soll ein Buchungssystem für Vorlesungsräume implementiert werden, um die Corona-Verordnung des Landes umzusetzen. Das KIT hat hierbei um Ihre Hilfe gebeten, da Sie in den letzten Monaten bereits in direktem Kontakt mit den relevanten Problemstellungen standen. Achten Sie bei Ihrer Implementierung auf Erweiterbarkeit, um bei eventuellen Änderungen der Corona-Verordnung das System einfach anpassen zu können.

Für die Bearbeitung der Aufgabe dürfen Sie die aus der Vorlesung bekannte Implementierung einer Liste verwenden. Diese muss dafür so verändert werden, dass sie andere Datentypen speichern kann. Den Quellcode aus der Vorlesung finden Sie auf den Vorlesungsfolien.

B.1 Genauere Beschreibung

Es gibt 3 verschiedene Personengruppen (Rollen):

- Dozierende, die Veranstaltung halten können,
- Studierende, die diese besuchen können und
- Sicherheitskräfte, die den korrekten Ablauf der Veranstaltungen sicherstellen.

Eine Veranstaltung findet an einem bestimmten Tag, in einem Vorlesungsraum statt. Gehen Sie davon aus, dass das Buchungssystem nie mehr als 100 Veranstaltungen verwaltet. Eine Veranstaltung wird von einer oder einem Dozierenden gehalten und Studierende haben die Möglichkeit sich für diese Veranstaltung anzumelden, vorausgesetzt die Kapazität des Raumes ist noch nicht überschritten, die Studierenden besitzen den nötigen und gültigen 2G/3G Nachweis und die Veranstaltung hat genügend Sicherheitskräfte für die Anzahl an Teilnehmenden. Standardmäßig muss mindestens eine Sicherheitskraft bei der Veranstaltung dabei sein, jedoch steigt der Bedarf an Sicherheitskräften mit der Anzahl der Teilnehmenden. Das Verhältnis hierbei ist 5 zu 1, also pro 5 Studierende eine Sicherheitskraft. Bei der Erstellung sollte ebenfalls auf die aktuelle Corona-Verordnung geachtet werden, die sich durch 2G (Geimpft, Genesen) oder durch 3G (Geimpft, Genesen, Getestet) auszeichnet. Zusätzlich besitzt die Veranstaltung noch eine eindeutige Identifikationsnummer. Gehen Sie davon aus, dass im Buchungssystem nie mehr als 1000 Personen verwaltet werden.

Ein Datum besteht aus einer ganzen Zahl aus dem Intervall $[0, 364]$ (um somit das Jahr 2022 zu repräsentieren). Mithilfe von dem unten spezifizierten `set-date`-Befehl, soll das aktuelle Datum gespeichert werden. Wurde das Datum noch nicht gesetzt, ist es standardmäßig 0.

Es soll zudem möglich sein, 3G-Nachweise einer Person hinzuzufügen. Diese Nachweise sind von einem der drei Typen: Geimpft, Genesen oder Getestet. Jeder Nachweis hat ein Ausstellungsdatum und ist für eine gewisse Zeit gültig. „Getestet“-Nachweise sind hierbei nur für denselben Tag gültig, „Genesen“-Nachweise für 180 Tage (beginnend mit dem Tag der Ausstellung) und „Geimpft“-Nachweise laufen nie aus. [Gehen Sie davon aus, dass einer Person nie mehr als 500 3G-Nachweise hinzugefügt werden.](#)

Ein Vorlesungsraum zeichnet sich durch einen eindeutigen alphanumerischen Raumnamen aus und hat eine begrenzte ganzzahlige Kapazität aus dem Intervall $[1, 2^{31} - 1]$, welche nicht überschritten werden darf. Zur Kapazität zählen sowohl Studierende als auch Dozierende und Sicherheitskräfte.

Auch Kontaktnachverfolgung soll möglich sein. Mithilfe des „report-case“-Befehls (siehe unten) soll ein positiver Fall gemeldet und dann alle Kontakte der betroffenen Person in den letzten 14 Tage, inklusive Dozierende und Sicherheitskräfte, ausgegeben werden. Hierbei soll ebenfalls ausgegeben werden, in wie vielen Veranstaltungen die Personen (in dem Zeitraum) gemeinsam waren.

B.2 Interaktive Benutzerinteraktion

Nach dem Start nimmt das Programm über die Konsole die nachfolgenden Befehlen unter Verwendung der Standardeingabe³ entgegen. Alle Befehle werden auf dem aktuellen Zustand des Programms

³<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/System.html#in>

ausgeführt. Sie können zu diesem Zweck die Klasse `Scanner`⁴ aus dem Paket `java.util` verwenden.

Nach der Abarbeitung eines Befehls wartet das Programm auf weitere Befehle, bis es durch Eingabe des `quit`-Befehls beendet wird. Die Ausführung der Befehle dürfen nicht gegen die definierten Spezifikationen verstoßen. Sie können in dieser Aufgabe davon ausgehen, dass bei den interaktiven Befehlen keine Fehler auftreten, diese also syntaktisch und semantisch den definierten Spezifikationen entsprechen. Für die Werte der Platzhalter `<...>` orientieren Sie sich an den angegebenen Beispielinteraktionen.

B.2.1 Aktuelles Datum setzen

Das aktuelle Datum kann mit dem `set-date`-Befehl eingestellt werden.

Eingabe `set-date <Datum>`

Ausgabe Bei erfolgreicher Eingabe wird `OK` ausgegeben.

➤ Beispielinteraktion

```
1 > set-date 42
2 OK
```

B.2.2 Personen hinzufügen

Der `add-Person`-Befehl registriert eine neue Person und teilt ihr eine eindeutige Personen-Identifikationsnummer (Personen-Id) zu. Die Ids werden aufsteigend und beginnend mit 0 vergeben. Die möglichen Rollen sind `student`, `lecturer` oder `security`.

Eingabe `add-person <Rolle>;<Vorname>;<Nachname>`

Ausgabe Die eindeutige Personen-Id.

➤ Beispielinteraktion

```
1 > add-person student;Max;Mustermann
2 0
3 > add-person student;Maximiliane;Musterfau
4 1
5 > add-person lecturer;Hans;Jürgen
6 2
7 > add-person student;Anna;Bauer
8 3
9 > add-person student;John;Doe
10 4
```

⁴<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/Scanner.html>

B.2.3 3G-Nachweis hinzufügen

Der `add-certificate`-Befehl fügt einer Person einen 3G-Nachweis hinzu. Die möglichen Nachweise sind `vaccinated`, `recovered` oder `tested`.

Eingabe `add-certificate <Personen-Id>;<3G-Nachweis>;<Datum>`

Ausgabe Bei erfolgreicher Eingabe wird `OK` ausgegeben.

➤ Beispielinteraktion

```
1 > add-certificate 2;tested;10
2 OK
3 > add-certificate 0;recovered;5
4 OK
5 > add-certificate 1;tested;42
6 OK
7 > add-certificate 2;vaccinated;15
8 OK
```

B.2.4 Person ausgeben

Der `print-person`-Befehl gibt die eingepflegten Daten der jeweiligen Person auf die Konsole aus. Zu den Daten gehören Personen-Id, Vorname, Nachname, Rolle und der 3G-Nachweis. Es sollen nur zum aktuellen Zeitpunkt gültige Nachweise ausgegeben werden. Wenn für eine Person kein gültiger 3G-Nachweis hinterlegt ist, wird stattdessen `no certificate` ausgegeben. Sind mehrere gültige Nachweise vorhanden, wird nur der am längsten gültige ausgegeben.

Eingabe `print-person <Personen-Id>`

Ausgabe `<Personen-Id>, <Vorname> <Nachname>, <Rolle>, <3G-Nachweis>`

➤ Beispielinteraktion

```
1 > print-person 0
2 0, Max Mustermann, student, recovered
3 > print-person 2
4 2, Hans Jürgen, lecturer, vaccinated
5 > print-person 1
6 1, Maximiliane Musterfrau, student, tested
7 > print-person 4
8 4, John Doe, student, no certificate
```

B.2.5 Personenart ausgeben

Der `print-people`-Befehl gibt eine Liste aller Personen im System mit der gleichen Rolle aus.

Eingabe `print-people <Rolle>`

Ausgabe Die Personen werden mit der Formatierung des `print-person`-Befehls jeweils in einer eigenen Zeile und ~~alphabetisch nach dem Vornamen~~aufsteigend nach der Personen-Id sortiert ausgegeben. Wenn im System keine Personen mit der gegebenen Rolle vorliegen, wird stattdessen `No <Rolle> in the system` auf der Konsole ausgegeben.

➤ Beispielinteraktion

```
1 | > print-people student
2 | 0, Max Mustermann, student, recovered
3 | 1, Maximiliane Musterfrau, student, tested
4 | 3, Anna Bauer, student, no certificate
5 | 4, John Doe, student, no certificate
6 | > print-people lecturer
7 | 2, Hans Jürgen, lecturer, vaccinated
8 | > print-people security
9 | No security in the system
```

B.2.6 Veranstaltung hinzufügen

Der `add-event`-Befehl erstellt eine neue Veranstaltung in einem Gebäude mit einer bestimmten Kapazität. Die Veranstaltung beinhaltet eine:n Dozent:in und eine Corona-Verordnung (2G oder 3G). Zudem wird der Veranstaltung eine eindeutige Veranstaltungs-Identifikationsnummer (Event-Id) zugeteilt. Die Ids werden aufsteigend und beginnend mit 0 vergeben.

Eingabe `add-event <Personen-Id>;<Gebäudenname>;<Kapazität>;<Verordnung>;<Datum>`

Ausgabe Die eindeutige Event-Id.

➤ Beispielinteraktion

```
1 | > add-event 2;Audimax;4;3G;42
2 | 0
```

B.2.7 Sicherheitskräfte hinzufügen

Der `increase-security`-Befehl fügt einer bereits erstellten Veranstaltung eine Sicherheitskraft hinzu. Jede Sicherheitskraft kann nur ein Mal an demselben Event teilnehmen.

Eingabe `increase-security <Event-Id>;<Personen-Id>`

Ausgabe Falls die Sicherheitskraft noch nicht für dieses Event hinzugefügt ist, sie die Veranstaltungsregeln erfüllt und es noch genug Platz in der Veranstaltung gibt, wird OK ausgegeben, andernfalls `Could not add security`.

➤ Beispielinteraktion

```
1 > add-person security;Bill;Baumeister
2 5
3 > add-certificate 5;vaccinated;0
4 OK
5 > increase-security 0;5
6 OK
```

B.2.8 Studierende einchecken

Der `book-spot`-Befehl ermöglicht es Studierenden sich einen Platz für eine Veranstaltung zu buchen. Alle Studierenden können nur ein Mal an derselben Veranstaltung teilnehmen.

Eingabe `book-spot <Event-Id>;<Personen-Id>`

Ausgabe Falls die Person noch keine Platz für dieses Event gebucht hat, sie die Veranstaltungsregeln erfüllt, es noch genug Platz in der Veranstaltung gibt und genug Sicherheitskräfte anwesend sind, wird `<Restkapazität> spot(s) left` ausgegeben, anderenfalls `Could not book spot`.

➤ Beispielinteraktion

```
1 > book-spot 0;0
2 1 spot(s) left
3 > book-spot 0;1
4 0 spot(s) left
5 > book-spot 0;3
6 Could not book spot
7 > book-spot 0;4
8 Could not book spot
```

B.2.9 Kontaktnachverfolgung

Der `report-case`-Befehl meldet einen positiven Fall und gibt daraufhin alle Personen aus, die in den letzten 14 Tagen mit der positiven Person in mindestens einer Veranstaltung waren. Außerdem wird pro Kontaktperson die Anzahl der gemeinsamen Veranstaltungen ausgegeben.

Eingabe `report-case <PersonId>`

Ausgabe Eine aufsteigend nach der Personen-Id sortierte Liste mit allen Personen in einer eigenen Zeile und im Format von `print-person`. Am Ende jeder Zeile wird die Anzahl der Veranstaltungen umgeben von eckigen Klammern hinzugefügt.

➤ Beispielinteraktion

```
1 | > report-case 0
2 | 1, Maximiliane Musterfrau, student, tested [1]
3 | 2, Hans Jürgen, lecturer, vaccinated [1]
4 | 5, Bill Baumeister, security, vaccinated [1]
```

B.2.10 Der quit-Befehl

Der parameterlose `quit`-Befehl beendet das Programm. Beachten Sie, dass hierfür keine Methoden wie `System.exit()`, `Runtime.exit()` oder vergleichbar verwendet werden dürfen.

Eingabe `quit`

Ausgabe Es findet keine Ausgabe statt.

➤ Beispielinteraktion

```
1 | > quit
```