



Datenstrukturen Übung – SS2020

Abgabe: 17.05. bis 24Uhr.

Abgabeort: Bitte laden Sie Ihre Klasse im OPAL-Bereich „Abgabe_Pflichtaufgabe_2“ hoch. Ergänzen Sie bei Ihrer hochgeladenen Klasse die Namen (Vor- und Nachname) aller beteiligten Personen. Sie können Ihre Abgabe bis zur Abgabefrist beliebig oft löschen und erneut einreichen.

Wichtig: Da es sich um eine Prüfungsvorleistung handelt, ist es wichtig, dass jeder Teilnehmer Zugang zu allen notwendigen Informationen hat. Deswegen bitten wir Sie Fragen direkt ins Forum, in den Thread „Fragen zur Pflichtaufgabe 2“ zu stellen.

Prüfungsvorleistung 2 – Die Studentenverwaltung

Schreiben Sie ein Programm zur Verwaltung von Studenten. Implementieren Sie dafür die Klasse "PVL2_studentAdministration_Group\$X" in Java oder Python, wobei \$X für ihre Gruppe steht.

Ein Student wird durch seine Matrikelnummer (1-999) eindeutig gekennzeichnet. Weiterhin soll ein Student eine Sammlung (Datenstruktur Ihrer Wahl) von Kursen besitzen, die er bereits belegt hat. Diese können dabei bestanden oder nicht bestanden sein. Es soll dabei auch gezählt werden, wie viele Fehlversuche stattgefunden haben. Sollten Übergabeparameter „null“ sein, geben Sie „null“ zurück.

Schreiben Sie die Methoden:

java: **public Integer matriculate(String firstName, String lastName)**

python: **def matriculate(self, firstName : str, lastName : str) -> int**

- Diese soll, mittels den gegebenen Übergabeparametern einen Studenten erstellen, und ihn in die Verwaltung einfügen.
- Weisen Sie ihm eine einmalige Matrikelnummer hinzu. Diese soll der ersten freien Matrikelnummer entsprechen.
- Geben Sie die Matrikelnummer zurück, sollte der Student angelegt wurden sein.
- Sollte ein Fehler auftreten (zum Beispiel keine gültigen Matrikelnummern mehr), geben sie -1 zurück.

java: **public Boolean deregister(Integer matriculationNumber)**

python: **def deregister(self, matriculationNumber: int) -> bool**

Diese Methode soll einen Studenten anhand der gegebenen Matrikelnummer exmatrikulieren. Gebe „false“ zurück, wenn es keinen Studenten mit dieser Matrikelnummer gab.

java: **public String find(Integer matriculationNumber)**

python: **def find(self, matriculationNumber: int) -> str**

Diese Methode soll einen Studenten anhand seiner Matrikelnummer finden (in maximal $O(n \log n)$) und seine Daten in einen String schreiben und zurückgeben.

Der Rückgabe-String soll folgendes Format besitzen. Wenn er k Kurse besitzt:

```
"$firstName\n
$lastName\n
$matriculationNumber\n
$Course[1]\n
$...
$Course[k]\n"
```

Ein Kurs soll folgendermaßen formatiert sein:

```
"$CourseName\t$NumberOfTries\t$passed"
```

Mit NumberOfTries in ["1", "2", "3"] und mit passed in ["true", "false"].

Sollte die Gegebene Matrikelnummer nicht vergeben sein, so lautet der Befehl return null.

java: **public String takeExam(Integer matriculationNumber, String courseID, Boolean passed)**

python: **def takeExam(self, matriculationNumber: int, courseID : str, passed : bool) -> str**

Diese soll, bei einem Studenten mit der passenden Matrikelnummer den Kurs hinzufügen.

Gib den Kurs zurück im folgenden Format:

```
"$CourseName\t$NumberOfTries\t$passed"
```

Mit NumberOfTries in ["1", "2", "3"] und mit passed in ["true", "false"].

Sollte er schon vorhanden sein wird der Versuchszähler erhöht. Anhand der gegebenen Flag wird eingetragen, ob die Prüfung bestanden oder nicht bestanden ist. Sollte der 3. Versuch ein Fehlversuch sein, so soll der Student exmatrikuliert werden. Sollte eine Prüfung geschrieben werden, die bereits bestanden ist, so soll nichts im Studenten verändert werden. Sollte die Matrikelnummer nicht vergeben sein, geben Sie "" zurück. Wenn der Student exmatrikuliert wird, geben Sie seine Daten trotzdem aus.

java: **public String dataBase()**

python: **def dataBase(self) -> str**

Diese soll, alle Studenten, nach Matrikelnummer geordnet zurückgeben.

Das Format soll wie folgt sein, wenn k Studenten eingeschrieben sind:

```
"$student[0]\n\n
...
$student[k]\n\n"
```

Lassen Sie ihre Klasse das Interface "StudentAdministration" implementieren, Sie finden es auch im Ordner „Materialien“ im OPAL:

[interface]

```
public interface StudentAdministration {
```

```
    Integer matriculate(String firstName, String lastName);
```

```
    Boolean deregister(Integer matriculationNumber);
```

```
    String find(Integer matriculationNumber);
```

```
    String takeExam(Integer matriculationNumber, String courseID, Boolean passed);
```

```
    String dataBase();
```

```
}
```