

Datenstrukturen - SoSe 23

PVL 1



Professur Softwaretechnik

4| 2023

Administratives

Abgabe Termin

24.04.2023 - 23:59

Abgabe

Ihre Abgabe darf entweder in Java oder Python geschehen. Nutzen Sie für die Abgabe das Code-Template aus dem OPAL-Bereich "Prüfungsvorleistung 1", dieses gibt eine Ordnerstruktur vor. Passen Sie lediglich den Platzhalter für Ihren Namen an. Abseits der Struktur ist es Ihnen erlaubt, weitere Dateien anzulegen. Sollten Sie sich nicht an diese Vorgaben halten, werden Ihnen Punkte abgezogen. Laden Sie Ihre Lösung als ZIP-Archiv im OPAL-Bereich "Prüfungsvorleistung 1" hoch.

Erlaubte Klassen

Sie dürfen Klassen und Methoden aus der Standardbibliothek von Java bzw. Python nutzen. Bitte benutzen Sie keine Bibliotheken von Dritten.

Information zu Plagiaten

Es ist **erlaubt**, Lösungen aus dem Internet in Ihre Abgabe zu integrieren. Bitte **markieren** Sie jeglichen Code, der nicht von Ihnen selbst stammt, indem Sie einen Kommentar mit der Quelle des Codes (URL ist ausreichend) an die entsprechende Stelle setzen. Dies soll verhindern, dass Ihre Lösung fälschlicherweise als Plagiat erkannt wird.

Da es sich hierbei um eine Prüfungsvorleistung handelt, ist es notwendig, dass Sie die Aufgaben selbständig bearbeiten. Daher werden alle Abgaben untereinander auf Plagiate überprüft. Sollten Plagiate erkannt werden, gilt die Abgabe von **sämtlichen** beteiligten Parteien als ungültig.

Kompilierbarkeit und Clean Code

Es werden nur Abgaben gewertet, welche sich in einem ausführbaren Zustand befinden. Achten Sie außerdem auf die Leserlichkeit des Codes. Unverständliche oder unangemessene Bezeichner von Klassen, Variablen, etc. können zu Punkteabzügen führen.

Fragen

Bitte stellen Sie ihre Fragen im Forum im Thread PVL1, damit alle Studierenden Zugang zu allen notwendigen Informationen erhalten.

PVL 1: Wetterbericht

Erstellen Sie eine Klasse **Forecast**, welche die Daten beliebig vieler Wetterstationen auswerten und daraus eine Wettervorhersage erstellen kann.

KONSTRUKTOR(**int**[][] *rainfall*, **String**[] *descriptors*)

Ein Konstruktor in der entsprechenden Programmiersprache, welchem die Daten der Wetterstationen in zwei Parametern übergeben werden. Zum einen die Niederschlagsdaten als zweidimensionales integer Array *rainfall* und zum anderen den allgemeinen Wetterbericht des Tages in Form eines beschreibenden Strings, im String-Array *descriptors*.

In dem Array *rainfall* entspricht jede Zeile einer Wetterstation und jede Spalte einem Tag.

Beispiel:

Weather Stations	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Station 1	-10	22	33	19	45	75	20
Station 2	35	-6	57	8	10	-100	10
Station 3	15	20	29	39	30	75	20

Table 1: Niederschlagsdaten von 3 Wetterstationen über einen Zeitraum von 7 Tagen

Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
sunny	rainy	thunderstorm	sunny	sunny	thunderstorm	sunny

Table 2: Wetterberichte der entsprechenden 7 Tage

dataPreparation()

Die Niederschlagsdaten können fehlerhaft sein und negative Werte enthalten. Diese Methode soll die Niederschlagsdaten so bearbeiten, dass Sie später weiterverwendet werden können. Dazu müssen mögliche fehlerhafte (negative) Daten entsprechend der nachfolgenden Spezifikationen entfernt werden:

- Ist der descriptor des entsprechenden Tags "sunny", wird der negative Wert mit 0 ersetzt.
- Ist der descriptor des entsprechenden Tags "rainy", wird der negative Wert mit dem Durchschnitt aller (positiven) Werte der anderen Wetterstationen von diesem Tag ersetzt. Rechnen Sie mit ganzen Zahlen. Sollten die Daten aller Stationen an diesem Tag fehlerhaft sein, tragen Sie bei allen Stationen 0 ein.
- Ist der descriptor des entsprechenden Tags "thunderstorm", wird der negative Wert durch den entsprechenden absoluten Wert ersetzt.

Beispiel: So sollen die oben angegebenen Beispieldaten nach der Modifizierung durch die Methode `dataPreparation` aussehen. Veränderungen sind rot markiert.

Weather Stations	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
Station 1	0	22	33	19	45	75	20
Station 2	35	21	57	8	10	100	10
Station 3	15	20	29	39	30	75	20

Table 3: Niederschlagsdaten nach `dataPreparation()`

`totalRainfall()`

Diese Methode berechnet den Gesamtniederschlag, den die Wetterstationen in Messzeitraum gemessen haben.

Beispiel:

Mit den gegebenen Beispieldaten beträgt der Gesamtniederschlag **683**.

`trend(int n)`

Diese Methode erstellt eine Vorhersage beruhend auf dem durchschnittlichen Niederschlag pro Station und Tag der letzten n Tage. Die Vorhersage wird in Form eines Strings, der einem der etablierten descriptors entspricht, zurückgegeben.

- Ist der durchschnittliche Niederschlag der letzten n Tage unter 50, gebe "sunny" zurück.
- Ist der durchschnittliche Niederschlag der letzten n Tage 50 oder höher, gebe "rainy" zurück.
- In dem besonderen Fall, dass der durchschnittliche Niederschlag der letzten n Tage *genau* 75 beträgt, gebe "thunderstorm" zurück.

Beispiel:

$n = 3$

Durchschnitt der letzten 3 Tage = $(45+75+20+10+100+10+30+75+20)/9 = 42$

→ Es wird "sunny" zurückgegeben.

$n = 2$

Durchschnitt der letzten 2 Tage = $(75+20+100+10+75+20)/6 = 50$

→ Es wird "rainy" zurückgegeben.