Aufgabenblatt #3

In Ihrer Übungsumgebung unter Linux am Informatik-Labor finden Sie das Projekt Aufgabenblatt3. Dieses Projekt besteht aus fünf Aufgaben, von denen Sie vier Aufgaben erweitern müssen. Die fünfte wird als Ad-hoc-Aufgabe in der Übung gestellt und darf bis dahin nicht verändert werden.

Abgabe: Die Abgabe muss bis Montag, den 01.12. um 15:59 erfolgen. Alle Änderungen am Projekt in der Übungsumgebung, die bis zu diesem Zeitpunkt von Ihnen vorgenommen wurden, werden von uns als Abgabe betrachtet. Verspätete Abgaben werden nicht akzeptiert. Bitte beachten Sie folgende Punkte:

- Dieses Aufgabenblatt wird benotet.
- Verändern Sie bitte nicht die Ordnerstruktur oder die Dateinamen!
- Ändern Sie nur die Inhalte der benötigten Dateien!
- Die Antworten zu den Zusatzfragen können bei den jeweiligen Aufgaben als Kommentare geschrieben werden. Achten Sie dabei auf korrekte Kommentare!

Aufgabe 1: Klassen und Objekte

Die Klasse Aufgabe1 soll einen Zähler implementieren, der bei 0 beginnt und bis zu einer maximalen Zahl `max` inkrementiert werden kann. Wird `max` erreicht, dann wird der Zähler auf 0 zurückgesetzt und ein Überlaufzähler auf 1 gesetzt. Danach kann der Zähler immer wieder bis `max` erhöht werden und bei einem Überlauf wird der Überlaufzähler wieder um 1 erhöht. Implementieren Sie folgende Methoden:

- `getCounter`: Liefert den aktuellen Zählerstand zurück.
- `getOverflowCounter`: Liefert den aktuellen Wert des Überlaufzählers zurück.
- `reset`: Setzt den Zähler und den Überlaufzähler auf den Anfangszustand zurück.
- `increment`:
  - Erhöht den Zähler um 1.
  - Wird dabei der maximale Wert erreicht, dann wird der Zähler auf 0 gesetzt und der Überlaufzähler um 1 erhöht.

Es sollte möglich sein, mehrere Zähler erzeugen zu können. Dazu müssen Sie einen Konstruktor implementieren, der den maximalen Wert als Parameter übergeben bekommt. Außerdem muss die Klasse so implementiert werden, dass die Methoden von außen aufgerufen werden können.

Zusatzfragen:

- Wozu benötigt man Klassen?
- Wie unterscheiden sich Objekte von Klassen?
- Warum benötigt man einen Konstruktor?
- Wann verwendet man `private` und wann `public`?
- Wie unterscheiden sich this und this() voneinander?

Aufgabe 2: Mehrere Klassen, Klassen verwenden

Implementieren Sie eine Klasse `Data`, die folgende Informationen verwaltet:

- Name einer Stadt als String gespeichert.
- Durchschnittliche Niederschlagshöhe pro Jahr (in mm) der Stadt als Ganzzahl gespeichert. Stellen Sie beim Erzeugen eines Objekts sicher, dass der Wert immer >=0 ist.

Die Klasse sollte einen entsprechenden Konstruktor anbieten. Danach wird der Inhalt eines Objekts nicht mehr verändert, sondern nur mehr ausgelesen. Verwenden Sie dazu die Methoden `getCity` und `getValue`.

Die Klasse Aufgabe2 verwaltet nun Datenobjekte (Objekt der Klasse Data) in einem Speicher. Diese Klasse bietet folgende Methoden an:

- `addData`: Mit dieser Methode wird ein Datenobjekt in den Speicher gegeben. Dazu werden der Methode zwei Parameter übergeben: Ein String (Name der Stadt) und der Wert für die durchschnittliche Niederschlagshöhe.
- `getData`: Mit dieser Methode wird ein Datenobjekt an den Aufrufer zurückgegeben. Dazu wird der Methode der Stadtname übergeben. Sind keine Daten für diese Stadt vorhanden, dann wird `null` zurückgeliefert.

Die Anzahl der Städte ist nicht fix vorgegeben. Ihre Implementierung sollte mit einer beliebigen Anzahl von Städten umgehen können. Für eine Stadt muss nur ein Eintrag abgespeichert werden.

Zusatzfragen:

- Sowohl Klassen als auch Arrays beinhalten Daten. Wie unterscheiden sich aber Klassen von Arrays?
- Wie unterscheidet sich ein Array von einer Map?

Aufgabe 3: Character-Stack

Implementieren Sie in Aufgabe 3 eine private Methode `check`, die einen String als Übergabeparameter besitzt. Diese Methode gibt `true` zurück, wenn sich im String eine korrekt geklammerte Sequenz mit runden Klammern befindet.

- Korrekte Klammerungen sind zum Beispiel (Beispiele durch Beistrich getrennt): (), (()), (())()((())). Auch der leere String ist korrekt geklammert.
- Nicht korrekte Klammerungen sind (Beispiele durch Beistrich getrennt): ((),), (, (((())(,)(

Verwenden Sie einen Stack zur Überprüfung. Immer wenn eine öffnende Klammer gelesen wird, dann wird diese auf den Stack gelegt. Wird eine schließende Klammer gelesen, dann wird die oberste öffnende Klammer vom Stack genommen. Findet man für eine schließende Klammer keine entsprechende Klammer am Stack oder befindet sich nach dem Durchlaufen des Strings noch eine Klammer am Stack, dann ist die Sequenz nicht richtig geklammert.

Hinweise: Sie müssen die Klasse `IntStack` aus der Vorlesung für die Verwendung mit Zeichen adaptieren. Dazu wird schon der Rumpf der Klasse `CharStack` zur Verfügung gestellt und muss von Ihnen erweitert werden. Weiters ist auch eine Klasse für einen Knoten in der verketteten Liste für `CharStack` vorhanden und muss von Ihnen implementiert werden.

Zusatzfragen:

- Erklären Sie den Unterschied zwischen dem LIFO und dem FIFO-Prinzip.
- Muss man bei einer Klasse immer einen Konstruktor implementieren? Wenn nein, was passiert dann?
- Kann man einen Stack auch mit einem Array implementieren? Wenn ja, welche Nachteile ergeben sich daraus?

Aufgabe 4: Queue

Um das Auswahlverfahren für Präsentationen in Übungen alternativ zu lösen hat sich ein Professor die folgende Methode ausgedacht: Alle n Studierenden werden in einem Kreis angeordnet. Dann wird beginnend mit einer Nummer m, jeder m-te Studierende entfernt, wobei der Kreis immer wieder geschlossen wird. Am Ende bleibt ein Studierender für die Präsentation übrig. Implementieren Sie in Aufgabe4 eine private Methode `assign`, die genau den obigen Sachverhalt für beliebige n und m ($n \geq m \geq 1$) implementiert und dabei eine Queue verwendet. Verwenden Sie die Queue-Implementierung aus Java und dabei nur die Methoden `offer` und `poll`. Jeder Studierende wird durch eine Nummer repräsentiert. Am Ende wird von der Methode `assign` eine Nummer zurückgegeben.

Hinweis: Befüllen Sie zunächst die Queue mit allen Nummern.

Zusatzfrage:

- Kann man eine Queue auch mit einem Array implementieren? Wenn ja, wie würden Sie dabei vorgehen?

Aufgabe 5: Ad-hoc-Aufgabe

Aufgabe5 wird als Ad-hoc-Aufgabe in den Übungen gestellt.