Projekt: Kontaktdatenbank Programmierung mit Java

Aufgabe

Es soll eine Anwendung in Java entstehen, welche sich mit den Personen und deren Kontakten beschäftigt. Die genauen Anforderungen an das Programm werden im folgenden Dokument beschrieben. Es soll auf ein objektorientiertes Datenmodell geachtet werden und es sollen Tests für die wichtigsten Funktionen vorhanden sein. Es dürfen außer JUNIT und dem JDK keine weiteren Bibliotheken verwendet werden. Es sind ebenfalls keine anderen Abhängigkeiten wie Builds-Tools oder ähnliches erlaubt.

Zusätzlich zu dem Programm soll ein Klassendiagramm angefertigt werden, welche sowohl die Beziehung der Klassen beschreibt, wie auch deren Methoden. Dieses Klassendiagramm darf nicht generiert werden und muss im PDF oder PNG Format abgegeben werden. Im Klassendiagramm müssen alle im Projekt verwendeten Klassen aufgeführt werden.

Unnötig komplexer Code muss vermieden werden.

Abgabe

Bitte sendet euer Projekt als Zip-Datei (.zip!) an unterstein@me.com. In dieser Zip-Datei müssen sowohl Quelltext, als auch Klassendiagramm enthalten sein. Der Name der Zip-Datei muss dem Muster "Kontaktprojekt_Kurs_Matrikelnummer.zip" entsprechen, wobei Kurs und Matrikelnummer durch den realen Wert ersetzt werden muss. Abgabeende ist der 31.07.2021 23:59:59.

Datenmodell

Das Datenmodell dieser Anwendung enthält Personen, Orte und Besuche.

Eine Person hab eine ID und einen Namen. Ein Ort hat eine ID, einen Namen und eine Angabe ob es drinnen oder draußen ist. Ein Besuch hat ein Startdatum, ein Enddatum, eine Person und einen Ort. Ein Besuch spiegelt den Aufenthalt einer Person an einem Ort für eine gewisse Dauer wieder, zum Beispiel "Johannes war am 15.05.2021 von 14:00-16:00 Uhr auf dem Spielplatz."

Daten laden

Es muss möglich sein die Datei http://wwwlehre.dhbw-stuttgart.de/~unterstein/contacts2021.db zu importieren und in das Datenmodell des Programmes zu überführen. Eine Zeile, welche mit 'New_Entity:' beginnt, leitet ein neues Objekt des Datenmodells ein. Es gibt folgende 3 Datensätze:

1. Personen
2. Orte
3. Besuche

In der Zeile, die mit 'New_Entity:' beginnt, sind Beschreibungen für die Spalten der folgenden Zeilen enthalten. Die Werte sind jeweils mit Komma von einander getrennt und jeweils von Anführungszeichen eingegrenzt. Die trennenden Kommas und Anführungszeichen sollen nicht ins Datenmodell importiert werden.

Der Import soll beim Starten des Programmes erfolgen, soll aber von einer lokalen Datei laden und nicht die Datei jedes mal neu abfragen. Bitte ladet die Datei herunter und speichert sie bei euch auf der Festplatte. Eine absolute Angabe des Pfades zu der Datei in eurem Code ist ok.

Beim importieren der Daten sollen überflüssige Leerzeichen am Anfang und Ende von Feldern (z.B. Name der Person) entfernt werden. Weiterhin kann das Datenmodell Dubletten enthalten, zum Beispiel ein Ort der mehrfach vorkommt. Sollte ein Datensatz mit einer ID mehrfach vorkommen, sollte der zweite ignoriert werden.

Personen suchen
Für den Benutzer des Programmes muss es möglich sein die ID und Details einer Person anzuzeigen. Dabei soll die Suche nach 'ila', sowohl "Mila" als auch "Milan" zurückgeben. Groß- und Kleinschreibung ist zu ignorieren.

Ort suchen
Für den Benutzer des Programmes muss es möglich sein die ID und Details eines Ortes anzuzeigen. Dabei soll die Suche nach 'Markt' unter anderem "Supermarkt" und "Großmarkt" zurück liefern. Groß- und Kleinschreibung ist zu ignorieren.

Kontaktpersonen anzeigen
Für den Benutzer des Programmes muss es möglich sein für eine gegebene Personen ID die Kontaktpersonen der Person anzuzeigen. Unter Kontaktpersonen einer Person verstehen wir in diesem Kontext die Liste aller Personen, welche zur gleichen Zeit an dem gleichen Ort waren.
Hierbei ist zu beachten, dass die Dauer des gemeinsamen Aufenthaltes egal ist und nur Besuche in geschlossenen Räumen "in_door" gezählt werden.

Die Ausgabe muss alphabetisch aufsteigend in folgendem Format geschehen:
Name1, Name2, Name3, …

Besucher und deren Kontakte anzeigen
Für den Benutzer des Programmes muss es möglich sein für eine gegebene Ort ID und Zeitpunkt (Format wie in der Datei) eine Liste aller Personen zu erhalten, welche zu diesem Zeitpunkt an diesem Ort waren. Zusätzlich zu diesen Personen sollen ebenfalls deren Kontakte aufgelistet werden.
Hierbei ist zu beachten, dass die Dauer des gemeinsamen Aufenthaltes egal ist und nur Besuche in geschlossenen Räumen "in_door" gezählt werden.
Sollte der gegebene Ort "out_door" sein, müssen nur die direkten Besucher zurückgegeben werden.

Die Ausgabe muss alphabetisch aufsteigend in folgendem Format geschehen:
Name1, Name2, Name3, …

Statischer Modus
Es wird keine Oberfläche für dieses Programm benötigt. Die Kommunikation mit dem Programm erfolgt mittels Übergabe von Argumenten. Argumente können keine Listen sein, sondern immer nur konkrete Suchbegriffe bzw. Ids. Das Programm soll folgende Argumente verstehen können:

personensuche: z.B.: --personensuche="Ila"
ortssuche: z.B.: --ortssuche="Markt"

kontaktpersonen: z.B.: --kontaktpersonen=1
besucher: z.B.: --besucher=1,"2021-05-15T14:16:00"

Unittests und Test auf Korrektheit

Die wichtigste Funktionalität (zum Beispiel die genannten vier Anforderungen, Import und so weiter) muss in einem Unittest geprüft werden. Zusätzlich zu den eigenen Unittests werden von mir Tests auf Korrektheit des Projektes durchgeführt.

Bewertung

Regeln für die Bewertung

1.) Kleine Abweichungen später noch möglich, z.B. falls ich was vergessen habe.
2.) Viele Punkte setzen voraus, dass das Programm kompiliert.
3.) Es gibt Abzüge falls gegen die oben genannten Anforderungen verstoßen wird, z.B. externe Abhängigkeiten, unnötig komplexer Code oder wenn das Dateiformat mißachtet wird.

Aufgabe	Prozentpunkte
OO Datenmodell	20
UML Diagramm	15
Import aus File	20
Unittests	5
Personensuche	5
Ortssuche	5
Kontaktpersonen anzeigen	7
Besucher und deren Kontakte anzeigen	7
Code verständlich und sinnvolle Docs	10
Test auf Korrektheit	6
	100