

Aufgabe

Es soll eine Anwendung in Java entstehen, welche sich mit den Personen und deren gekauften Produkten 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 "Produktprojekt_Kurs_Matrikelnummer.zip" entsprechen, wobei Kurs und Matrikelnummer durch den realen Wert ersetzt werden muss. Abgabeende ist der 26.07.2020 23:59:59.

Datenmodell

Das Datenmodell dieser Anwendung enthält Personen, Firmen und Produkte.

Eine Person hat eine ID, einen Namen und ein Geschlecht. Eine Firma hat eine ID und einen Namen. Ein Produkt hat eine ID und einen Namen. Ein Produkt wird von einer Firma hergestellt. Ein Person ist mit weiteren Personen befreundet. Freundschaft ist immer zweiseitig.

Daten laden

Es muss möglich sein die Datei <http://www.lehre.dhbw-stuttgart.de/~unterstein/movieproject2020.db> zu importieren und in das Datenmodell des Programmes zu überführen.

Eine Zeile, welche mit 'New_Entity:' beginnt, leitet ein neues Objekt des Datenmodelles ein.

Es gibt folgende 7 Datensätze:

1. Personen
2. Produkte
3. Firmen
4. Personen sind befreundet
5. Person hat Produkt gekauft
6. Firma stellt Produkt her

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.

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 Produkt der mehrfach vorkommt. Sollte ein Datensatz mehrfach vorkommen, muss das zweite Auftreten erneut erfasst werden als eigenständiger Datensatz, da auch eine eigene ID vorliegt. Sollte ein Datensatz (Zeile) ein Problem beim Importieren erzeugen, weil die Formatierung nicht korrekt ist, kann dieser Datensatz ignoriert werden. Ich wäre über eine kurze Rückmeldung über diese defekten Datensatz sehr dankbar.

Personen suchen

Für den Benutzer des Programmes muss es möglich sein die ID und Details einer Person anzuzeigen. Dabei soll die Suche nach 'Roger', sowohl "Roger Walker" als auch "Jamie Rogers" zurückgeben. Groß- und Kleinschreibung ist zu ignorieren.

Produkt suchen

Für den Benutzer des Programmes muss es möglich sein die ID und Details eines Produktes anzuzeigen. Dabei soll die Suche nach 'iPad' unter anderem "iPad" und "iPad mini" zurück liefern. Groß- und Kleinschreibung ist zu ignorieren.

Produktnetzwerk anzeigen

Für den Benutzer des Programmes muss es möglich sein für eine gegebene Personen ID das Produktnetzwerk der Person angezeigt zu bekommen. Unter Produktnetzwerk einer Person verstehen wir in diesem Kontext die Liste alle Produkte welche Freunde der gegebenen Person gekauft haben. Es ist darauf zu achten, dass ausgegebene Namen nicht mehrfach vorkommen, die Reihenfolge ist dabei alphabetisch aufsteigend. Es ist weiterhin darauf zu achten, dass Produkte, welche die gegebene Person bereits gekauft hat, nicht in dieser Liste vorkommt.

Die Ausgabe muss in folgendem Format geschehen:
Name1, Name2, Name3, ...

Firmennetzwerk anzeigen

Für den Benutzer des Programmes muss es möglich sein für eine gegebene Personen ID das Firmennetzwerk der Person angezeigt zu bekommen. Unter Firmennetzwerk einer Person verstehen wir in diesem Kontext die Liste alle Firmen, welche Produkte herstellen, welche Freunde der gegebenen Person gekauft haben. Es ist darauf zu achten, dass ausgegebene Namen nicht mehrfach vorkommen, die Reihenfolge ist dabei alphabetisch aufsteigend. Es ist weiterhin darauf zu achten, dass Firmen, von denen die gegebene Person bereits Produkte gekauft hat, nicht in dieser Liste vorkommt.

Die Ausgabe muss 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="Roger"
produktsuche: z.B.: --produktsuche="Matrix"
produktnetzwerk: z.B.: --produktnetzwerk=4899
firmennetzwerk: z.B.: --firmennetzwerk=17562

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

Aufgabe	Prozentpunkte
OO Datenmodell	20
UML Diagramm	15
Import aus File	20
Unittests	5
Personensuche	5
Produktsuche	5
Produktnetzwerk	7
Firmennetzwerk	7
Code verständlich und sinnvolle Docs	10
Test auf Korrektheit	6
	100

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.