Arbeitsstand

Das Bank-Projekt soll Transaktionen zwischen Konten von Kunden nach den Geschäftsregeln der Wossi-Bank simulieren. In Analogie zur Datenbankentwicklung wurden bislang die Klassen **`Kunde`** und **`Konto`** entworfen.

Beim Schließen des Projekts gehen jedoch alle Daten und Transaktionen verloren. In der Realität werden die Objektdaten extern in Dateien oder Datenbanken gespeichert. Dies soll im Projekt mit Hilfe der Klasse **`Verwalter`** für die Verwaltung der Objekte der Klassen **`Kunde`** und **`Konto`** sowie der Klasse **`DatenZugriff`** für das Speichern und Einlesen der Daten geschehen.

Klasse Datenzugriff

1) Kopieren Sie Klasse **`DatenZugriff`** sowie die CSV-Dateien **`Konten`** und **`Kunden`** in ihren Projektordner.
2) Öffnen Sie die CSV-Dateien mit einem Editor. Beschreiben Sie den Inhalt der beiden CSV-Dateien. Vergleichen Sie mit den Eigenschaften eines Kunden bzw. Kontos. Merken Sie sich den Saldo von Konto 1. Schließen Sie die Dateien.
3) Öffnen Sie das BlueJ-Projekt. Die Klasse **`Datenzugriff`** erscheint nun im Projekt. Übersetzen Sie sie. Beseitigen Sie ggf. auftretende Fehler durch Anpassung der Bezeichner, des Konstruktors, der Methodenarten oder der Parameterreihenfolge in Ihren Klassen **`Kunde`** und **`Konto`**. Die Klasse **`DatenZugriff`** benötigt in den Klasse **`Kunde`** und **`Konto`** zwingend jeweils eine Methode **`getDatensatz`**, welcher die Daten als kommagetrennte Zeichenkette in der richtigen Reihenfolge liefert.
4) Erstellen Sie ein Objekt der Klasse **`DatenZugriff`** und lesen sie darüber Daten ein. Inspizieren Sie das Objekt. Beschreiben Sie die Attribute des Objekts. Erstellen Sie eine Klassenkarte der Klasse. Markieren Sie Aufträge/Anfragen sowie Attribute unterschiedlich.
5) Schließen Sie das BlueJ-Projekt.

Klasse `Verwalter`

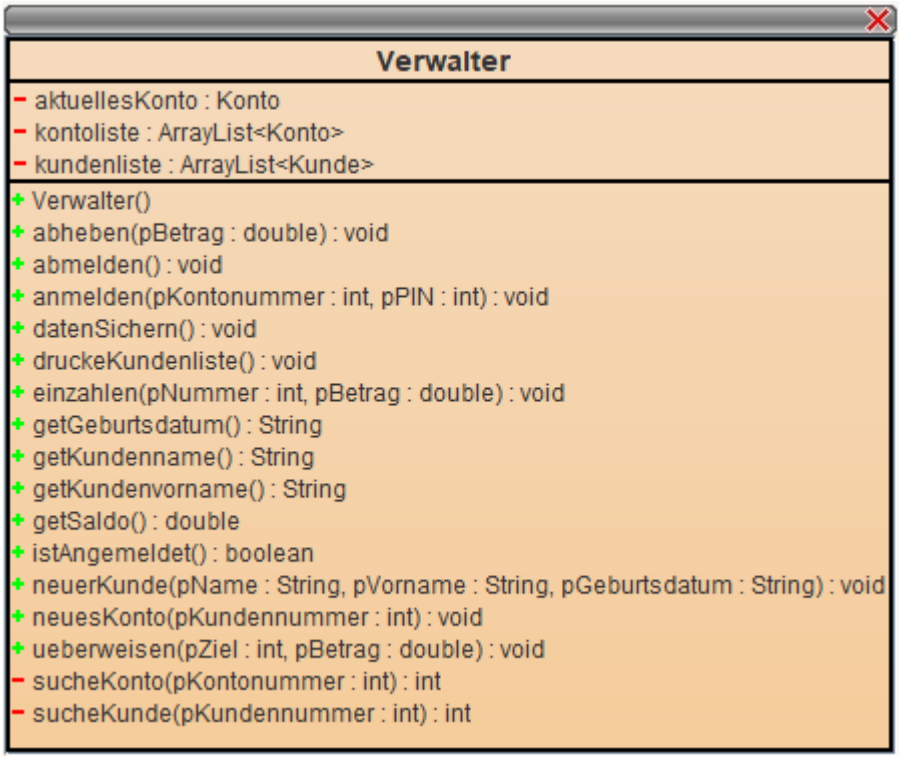
6) Fügen Sie über das Menü „Bearbeiten – Klasse aus Datei hinzufügen …“ die Klasse **`Verwalter`** in das Projekt ein.
7) In der Klasse **`Verwalter`** werden – wie bereits in der Klasse **`Datenzugriff`** – Listen eingesetzt. Informieren Sie sich auf der Tafelwerksergänzung Listen über
   a) die Voraussetzungen für die Verwendung von Listen,
   b) den Durchlauf durch Listen.
8) Prüfen Sie, ob die Voraussetzungen für die Verwendung von Listen vorliegen. Korrigieren Sie ggf.
9) Erzeugen Sie ein neues Verwalterobjekt und inspizieren Sie es. Beschreiben Sie das Objekt. Zahlen Sie etwas Geld auf das Konto mit der Nummer 1 ein, Die PIN lautet 1234. Speichern Sie die Daten über die Methode **`sichereDaten`**. Kontrollieren Sie die beiden CSV-Dateien. Vergleichen Sie mit dem notierten Saldo von Konto 1.
10) Analysieren Sie die Methode zum Einzahlen eines Geldbetrags auf ein beliebiges Konto. Beschreiben Sie die Arbeitsweise der Methode knapp.

CC BY-SA 4.0 T. Hempel · Version vom 21.11.2024

11) Entwickeln Sie das neue Klassendiagramm mit allen Beziehungen. Ermitteln Sie die Kardinalität der Beziehungen.

12) Erläutern Sie Unterschiede zwischen den Beziehungen der Klassen Konto-Kunde und Verwalter-Konto.

13) Entwickeln Sie auf Basis der vorhandenen Methoden **`einzahlen`** und **`sucheKonto`** die restlichen Methoden gemäß Klassendiagramm.



14) Vervollständigen Sie alle Java-Doc-Kommentare.

15) Informieren Sie sich unter http://www.oop-uml.de/drei-schichten-architektur.php oder/und http://www.mrknowing.com/2013/11/08/wie-funktioniert-die-3-schichten-architektur/ über die Drei-Schicht-Architektur von Software und fassen Sie die Aussagen geeignet zusammen. Geben Sie an, welche Schichten im Projekt realisiert wurden. Begründen Sie.

CC BY-SA 4.0 T. Hempel · Version vom 21.11.2024