

Kontoverwaltung

Eine Bank möchte eine Software zur Verwaltung ihrer Kundenkonten erstellen lassen. Es wird festgelegt, dass jedes Konto den Namen des Inhabers, eine zufällig gewählte neunstellige Kontonummer (ohne führende Nullen), den Kontostand sowie den Guthabenzinssatz in Prozent beinhalten muss.



Es sind zwei verschiedene Konten im Angebot, die durch je eine eigene Klasse repräsentiert werden sollen. Der Kontostand des Girokontos darf bis zu einem vereinbarten Betrag negativ werden (Überziehungsrahmen). Dafür muss der Kunde Überziehungszinsen bezahlen.

Das Guthaben des Sparkontos darf 0EUR nicht unterschreiten. Bei beiden Kontotypen ist es möglich, Ein- und Auszahlungen zu verbuchen.

Teil 1



(T1A1)

Stellen Sie die Beziehungen zwischen den Klassen Bankverwaltung, Konto, Girokonto und Sparkonto in einem Klassendiagramm dar. Die Klassen sollen nur den Klassennamen, keine Attribute und Methoden enthalten. Abstrakte Klassen müssen als solche gekennzeichnet sein.

Die Grafik zeigt das UML-Klassendiagramm der Klasse Konto.



(T1A2)

- Entwerfe passend zur Klasse Konto ein UML-Klassendiagramm für die Klasse Girokonto. Gib dabei die Sichtbarkeit aller Attribute und Methoden, die Parameterlisten und eventuelle Rückgabewerte an.
- Erstelle jeweils einen Konstruktor für die Klassen Konto und Girokonto.

Um Geld ein- oder auszahlen zu können, enthält die Klasse Konto eine abstrakte Methode buchen (betrug: double): boolean. (d) Implementieren Sie diese Methode in der Klasse Girokonto unter Beachtung des Überziehungsrahmens. Der Rückgabewert true bedeutet, dass die Buchung ausgeführt werden konnte. (7 VP) Zu Testzwecken soll die Anzahl der Konten zunächst auf 100 beschränkt sein. Die Klasse Bankverwaltung soll Spar- und Girokonten erzeugen und verwalten können. Die folgenden Aufgaben beziehen sich alle auf die Klasse Bankverwaltung. In jeder Teilaufgabe dürfen Methoden aus

anderen Teilaufgaben verwendet werden. (a) Geben Sie die Deklaration und Initialisierung einer geeigneten Datenstruktur zur Speicherung der Konten in der Bankverwaltung an. (b) Implementieren Sie eine Methode `sucheKonto (kontonummert int) : Konto`, die das Konto-Objekt mit der angegebenen Kontonummer zurückliefert bzw. `null` (bzw. `None`), wenn kein solches Konto existiert. © Implementieren Sie eine Methode `neueKontonummer () int`, die eine neue neunstellige Kontonummer zurückgibt. Diese darf noch keinem existierenden Konto zugewiesen sein. Verwenden Sie als Hilfsmethode `geteufalls zahl (min: int, mak: int) : int`, die eine ganzzahlige Zufallszahl z mit $\min \leq z \leq \max$ zurückgibt. (d) Implementieren Sie die Methode `erstelleGirokonto (inhabername: String, ueberziehungsrahmen: double) int`, die ein neues Konto mit zufälliger Kontonummer erzeugt und in der von Ihnen angelegten Datenstruktur speichert. Der Rückgabewert soll die Nummer des neu erzeugten Kontos sein bzw. `0`, falls kein neues Konto angelegt werden konnte. (e) Implementieren Sie eine Methode `ueberweise (vonKontonummer: int, aufKontonummer: int, betrag: double) boolean`. Diese Methode soll den angegebenen Geldbetrag vom Konto mit der Kontonummer `vonKontonummer` auf das Konto mit der Kontonummer `aufKontonummer` überweisen, sofern die Konten beide existieren, das Absenderkonto ausreichend gedeckt ist und der Betrag positiv ist. Falls die Überweisung durchgeführt werden konnte, soll `true` zurückgegeben werden, andernfalls `false`.

From:
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:modellierung:2020.a:start?rev=1639642397>

Last update: 16.12.2021 09:13

