Beziehungen von Klassen in C++

Lernziele

Hier sollen die Themen *Assoziation*, *Aggregation* und *Komposition* vertieft und angewendet werden.

Aufgabe 9: (Bank, Kunde, Konto)

Erweitern Sie die Funktionalität der Klassen `Sparkonto` und `Girokonto` aus Aufgabe 8, und realisieren Sie eine Bank, die Kunden und Konten verwaltet. Die parametrisierte Klasse `Liste` aus der Vorlesung kann zur Speicherung der Kontobewegungen innerhalb der Klasse `Konto` verwendet werden. Außerdem werden die Kunden und die Konten in einer solchen Liste gespeichert. Ein mögliches Klassendiagramm ist in Abbildung 1 dargestellt.

Die Angabe des Datums bei Ein-/Auszahlungen und Überweisungen wird in einer echten Bank normalerweise automatisch vom System mit dem aktuellen Datum vorbelegt. Damit wir aber die Zinsberechnung testen können, erlauben wir auch eine explizite Angabe des Datums. Die Klasse `Datum` stellen wir zur Verfügung.

Wir stellen Ihnen eine Benutzeroberfläche zur Verfügung, die wie folgt aussieht:

```
+--------------------------------+
|           Hauptmenü            |
+--------------------------------+
  ( 1) Einzahlung
  ( 2) Auszahlung
  ( 3) Überweisung
  ( 4) Kontoauszug
  ( 5) Kundenliste anzeigen
  ( 6) Kontoliste anzeigen
  ( 7) Stammdaten-Dialog
  ( 8) Daten einlesen
  ( 9) Daten speichern
  (10) Zinsgutschrift

  (11) Programm beenden
---------------------------------
  Ihre Auswahl?
```

```
+--------------------------------+
|      Stammdaten-Dialog         |
+--------------------------------+
  (1) Kunde anlegen
  (2) Kunde entfernen
  (3) Kundendaten ändern
  (4) Konto anlegen
  (5) Konto entfernen

  (6) Abbruch
---------------------------------
  Ihre Auswahl?
```

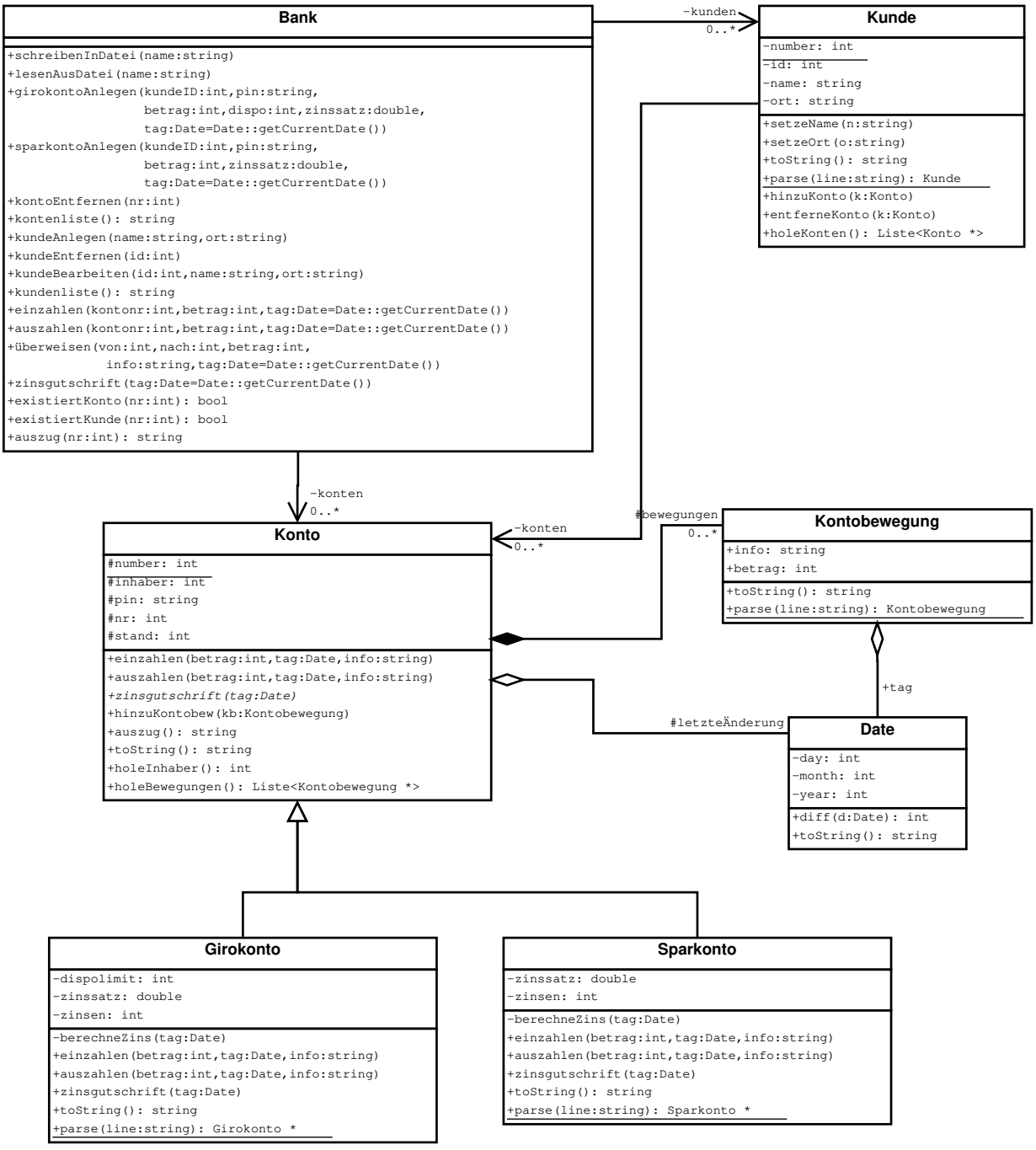

Abbildung 1: UML-Klassendiagramm für das Bank-Beispiel

Bei einer Einzahlung (Auswahl 1) wird zunächst die Kontonummer eingegeben und geprüft, ob ein entsprechendes Konto vorhanden ist. Anschließend wird der Betrag und das aktuelle Datum eingegeben.

```
+------------------------------+       +------------------------------+
|         Einzahlung           |       |         Einzahlung           |
+------------------------------+       +------------------------------+
    Kontonummer?  42                       Kontonummer?  43

         Betrag?  1000
          Datum?  5.4.2021             Fehler: Konto existiert nicht!!!
```

Zusatz-Aufgabe Z9: (Heizung, Thermostat)

In dieser Aufgabe soll ein Heizungssystem entsprechend der folgenden Modellierung umgesetzt werden: Zentrales Element ist eine `Heizung`, die in jedem beheizten Raum ein `Thermostat` besitzt. Jedes Thermostat wiederum besitzt einen `Sensor` zur Erfassung der momentanen Temperatur und ein `Ventil`, über das die Heizleistung eingestellt werden kann.

Die Implementierung der Klasse `Heizung` ist bereits gegeben. Implementieren Sie nun die weiteren Klassen unter Berücksichtigung der folgenden Anforderungen:

- Der `Sensor` soll mit einer zufälligen Temperatur zwischen 15 und 25 Grad initialisiert werden. Die Methode `gibIstTemperatur` soll die aktuelle Temperatur zurückliefern. Die Methode `beeinflusse` verändert die Temperatur entsprechend der in der Vorlage gegebenen Formel.
- Das `Ventil` muss zunächst mittels `verknuepfeSensor` mit einem Sensor verknüpft werden, damit bei der Ausführung eine Änderungen des Ventils der entsprechende Sensor beeinflusst werden kann. Die Methode `stelle` soll einfach den entsprechenden Sensor beeinflussen.
- Für das `Thermostat` existieren zwei Varianten. Das `DiskreteThermostat` stellt das Ventil entweder auf zu (0.0) oder auf (1.0). Sinkt die Temperatur 1 Grad unter den Sollwert, wird das Ventil aufgedreht. Steigt die Temperatur 1 Grad über den Sollwert, wird das Ventil abgedreht. Andernfalls wird die momentane Ventilstellung beibehalten. Das `AnalogeThermostat` dreht das Ventil ab, sobald die Solltemperatur überschritten wird. Andernfalls wird das Ventil pro Grad Differenz um 0.1 geöffnet - bei 10 Grad unter der Zieltemperatur ganz aufgedreht.

Das Klassendiagramm in Abbildung 2 zeigt nochmal die Beziehungen zwischen den einzelnen Klassen.

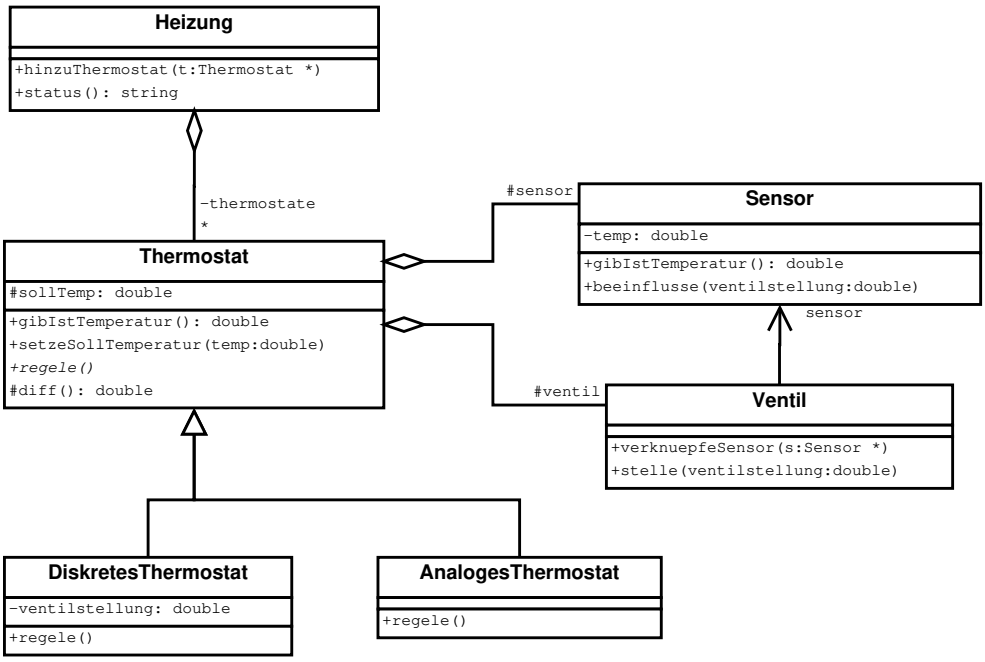

Abbildung 2: UML-Klassendiagramm für die Gebäudesteuerung