

Aufgabe 1 (Klasse Konto):

Erstellen Sie eine Klasse „Konto“ mit den privaten Eigenschaften:

- Kontonummer (int)
- Kontostand (float)

Beim Anlegen eines neuen Kontos (Konstruktor) soll die Kontonummer automatisch ab einem Startwert (z.B. 111100) fortlaufend vergeben werden. Definieren Sie hierfür zunächst eine globale Variable mit diesem Startwert und erhöhen diesen Wert bei jedem Aufruf des Konstruktors.

Der Kontostand darf NIE ins Soll kommen, darf also niemals negative Werte annehmen, der Kontostand 0,00 sei jedoch erlaubt.

Bei einer „Kontoeröffnung“ kann der Benutzer sogleich einen beliebigen Betrag auf das Konto einzahlen, muss aber nicht.

Selbstverständlich kann man auf das Konto mit den Methoden „**einzahlen**“ und „**auszahlen**“ einen Betrag einzahlen oder abheben, worauf der Kontostand natürlich aktualisiert wird – und wie gesagt kein Soll aufweisen darf.

Außerdem kann man jederzeit den Kontostand und seine Kontonummer abfragen, schreiben Sie hierfür jeweils 2 Methoden, die einmal nur den Wert zurückgeben bzw. die Auskunft am Bildschirm ausgeben. Vorschlag für die Methodennamen: **getKontonummer**, **zeigeKontonummer**, **getKontostand**, **zeigeKontostand**.

Sehen Sie auch einen Destruktor vor, der beim Auflösen eines Kontos das verbleibende Guthaben auf dem Konto auf dem Bildschirm anzeigt.

Schreiben Sie ein geeignetes Test-Szenario für Ihre Klasse.

Aufgabe 2 (Kundenkonto):

Entwickeln Sie nun eine Klasse „Kunde“ mit den privaten Eigenschaften:

- Name (string)
- Kundenkonto (Konto)

Beim Anlegen eines neuen Kunden (Konstruktor) soll sogleich der Name und evtl. ein erster Einzahlungsbetrag angegeben werden. Dabei wird auch gleich ein neues Konto für diesen Kunden eröffnet, das für diesen Kunden natürlich ebenfalls die Möglichkeiten des Ein- und Auszahlens und der Abfrage von Kontostand und Kontonummer bereit stellt. Wenn der Kunde kündigt (Destruktor) wird auch das Konto aufgelöst...

TIPP: Das Konto für den neuen Kunden sollte „dynamisch“ mit der new-Anweisung angelegt werden, damit es beim Kunden-Destruktor gelöscht werden kann („delete“).

Schreiben Sie ein geeignetes Test-Szenario für Ihre Klasse.

Beachten Sie auch die zum Beginn der Übung „live“ vorgetragenen Hilfestellungen!

Aufgabe 3 (static-Variablen):

Die in Aufgabe 1 verwendete globale Variable für die laufend fortgeschriebene Kontonummer kann natürlich auch im main-Programm manipuliert werden...
Probieren Sie es aus!

Im „Cyberwar“ könnte ein Bösewicht diesen Wert auf die höchste Integer-Zahl setzen (dann könnten keine funktionierenden Konten mehr eröffnet werden...) oder den Wert auf den Startwert zurücksetzen (dann gibt es die folgenden Kontonummern doppelt...) Das wollen wir aber nicht!!!

Legen Sie die Variable als statische Eigenschaft der Klasse „Konto“ an und probieren es erneut. Hurra! Ein Angriffspunkt ist beseitigt...

Merksatz:

Statisch deklarierte Klassen-Eigenschaften werden nur einmal angelegt und gelten für ALLE erzeugten Objekte dieser Klasse gemeinsam.