

VL09, Aufgabe 1 (Übung)

Beantworten Sie folgende Fragen:

1. Was ist der Zweck des Geheimnisprinzips?
2. Was ist der Unterschied zwischen einem Objektattribut und einem Klassenattribut?
3. Worin besteht der Unterschied, wenn eine Methode innerhalb bzw. außerhalb der deklarierenden Klasse aufgerufen wird?
4. Worin besteht der Unterschied zwischen call by reference und call by value?

VL09, Aufgabe 2 (Übung)

Gegeben seien die Klasse `Zaehler` und `ZaehlerTest`.

```
class Zaehler
{
    //Objektattribute
    private int zaehlerstand;

    //Konstruktoren
    public Zaehler(int wert)
    {
        this.zaehlerstand = wert;
    }

    //Methoden
    public void uebertrage(Zaehler zweiterZaehler)
    {
        zweiterZaehler.setZaehlerstand(this.zaehlerstand);
    }

    public void setZaehlerstand(int wert)
    {
        this.zaehlerstand = wert;
    }

    public int getZaehlerstand()
    {
        return this.zaehlerstand;
    }

    public boolean equals(Zaehler zweiterZaehler)
    {
        return this.zaehlerstand == zweiterZaehler.getZaehlerstand();
    }
}

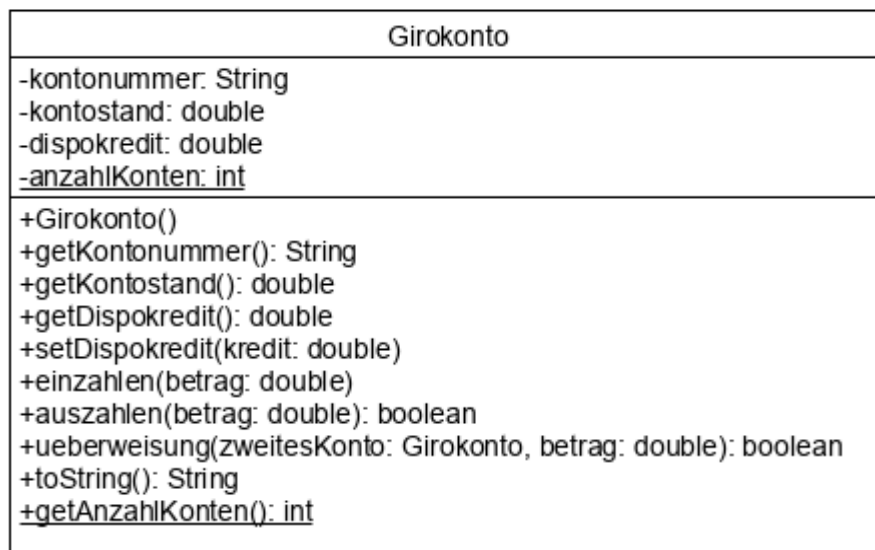
01 class ZaehlerTest
02 {
03     public static void main(String[] args)
04     {
05         Zaehler einKmZaehler = new Zaehler(1);
06         Zaehler nochEinKmZaehler = new Zaehler(42);
07
08         nochEinKmZaehler.uebertrage(einKmZaehler);
09         System.out.println(einKmZaehler == nochEinKmZaehler);
10         System.out.println(einKmZaehler.equals(nochEinKmZaehler));
11     }
12 }
```

- a) Zeichnen Sie ein Speicherbild während der Ausführung von Zeile 8 (vor dem Ende der Methode `setZaehlerstand`, die in `uebertrage` aufgerufen wird).
- b) Welche Ausgabe erzeugen die Zeile 9 und 10? Begründen Sie Ihre Antwort.

VL09, Aufgabe 3 (Praktikum)

Erstellen Sie in Eclipse ein neues Projekt mit dem Namen `EidP-VL09-Aufgabe3`. Fügen Sie dem Projekt zwei Klassen `Girokonto` und `GirokontoTest` hinzu.

- a) Programmieren Sie die Klasse `Girokonto` entsprechend dem folgenden UML-Klassendiagramm:



- Der Konstruktor erzeugt für jedes Konto eine fortlaufende zehnstellige Kontonummer, die mit 0822 beginnt. Das erste Konto erhält die Kontonummer 0822000001, das zweite die Kontonummer 0822000002, usw.
 - Die Funktion `auszahlen` darf den Kontostand nur ändern, wenn Kontostand und Dispokredit dies erlauben. Bei erfolgreicher Auszahlung wird `true` als Ergebnis zurückgegeben, ansonsten `false`.
 - Die Funktion `ueberweisung` überträgt den Betrag vom ersten auf das zweite Konto. Nutzen Sie dazu die Methoden `auszahlen` und `einzahlen`. War die Überweisung erfolgreich wird `true` zurückgegeben, ansonsten bleiben die Kontostände unverändert und es wird `false` zurückgegeben.
 - Die Methode `toString` soll eine Zeichenkette bestehend aus Kontonummer, Kontostand und Dispokredit durch Komma getrennt als Zeichenkette zurückgeben.
- b) Programmieren Sie die Klasse `GirokontoTest`, in deren Hauptprogramm (`main`) mehrere Objekte der Klasse `Girokonto` erzeugt werden, alle Methoden mindestens einmal und die Methoden `auszahlen` und `ueberweisung` mindestens einmal mit und einmal ohne Erfolg aufgerufen werden. Fügen Sie nach dem

Aufruf jeder Methode Bildschirmausgaben auf das Konsolenfenster ein, um die Kontoänderungen zu überprüfen.

Hinweis: Da in der Klasse `Girokonto` die öffentliche Methode `toString` implementiert ist, können Sie die Attributwerte von `einKonto` direkt wie folgt auf der Konsole ausgeben: `System.out.println(einKonto);`