

VL09, Aufgabe 1 (Übung)**(☆)**

Beantworten Sie folgende Fragen:

1. Was ist das Geheimnisprinzip in der Programmierung, und in welchen Situationen wird es wichtig?
2. Worin besteht der Unterschied zwischen Objekt- und Klassenattributen, und wie wird er in Java technisch realisiert?
3. Wie unterscheidet sich der Aufruf einer Methode innerhalb der sie deklarierenden Klasse in Java vom Aufruf derselben Methode außerhalb der Klasse?
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 (Übung)

(☆☆☆)

Stellen Sie sich vor, sie sollten ein Onlinespiel programmieren, bei dem Spielende das Rätsel der vier Wanderer lösen müssen.

Vier Wanderer müssen über eine unbeleuchtete Hängebrücke gehen. Die Brücke trägt immer nur zwei Personen gleichzeitig. Für jede Überquerung brauchen sie unbedingt eine Taschenlampe, die insgesamt nur 60 Minuten brennt. Die vier Wanderer brauchen für den Weg über die Brücke unterschiedlich lange Zeit:

- *Wanderer A braucht 5 Minuten.*
- *Wanderer B braucht 10 Minuten.*
- *Wanderer C braucht 20 Minuten.*
- *Wanderer D braucht 25 Minuten.*

Achtung: Die Gehzeit zählt für jede Überquerung, egal ob hin oder zurück. Gehen zwei Wanderer zusammen, zählt immer die Gehzeit des langsamsten.

Wie kommen die Wanderer in 60 Minuten über die Hängebrücke?

Die Spielenden lösen das Rätsel, in dem Sie Spielzüge eingeben, dabei wechselt bei jedem Spielzug die Richtung. Das Onlinespielprogramm kontrolliert nach jedem Zug, ob das Rätsel gelöst ist.

Beispiel:

Spielzug 1 (links-nach-rechts) : A, B

Spielzug 2 (rechts-nach-links): A

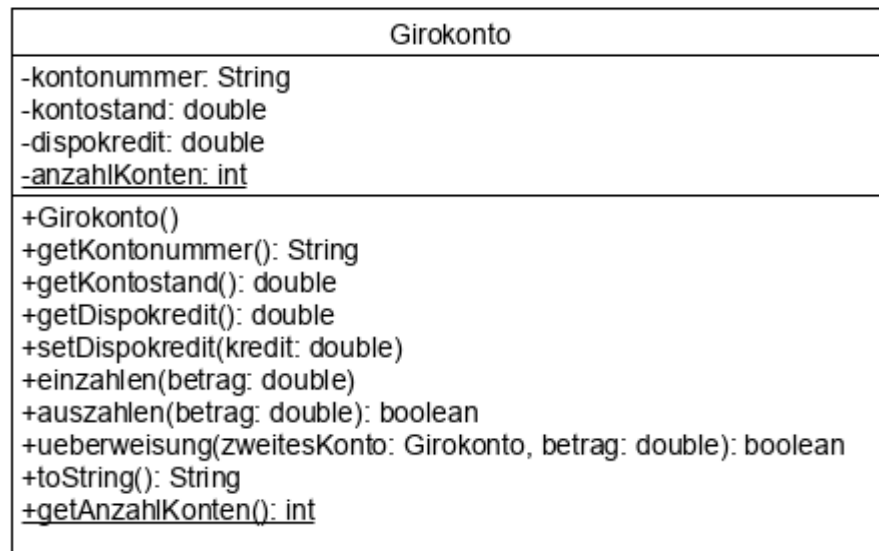
...

- a) Welche „Dinge“ kommen in dem Spiel vor, die als Objekte modelliert werden können? Legen Sie fest, welche Klassen programmiert werden sollten.
- b) Welche Informationen müssen in Zuständen der von Ihnen in Teilaufgabe a) gewählten Objekte gespeichert werden, um den aktuellen Spielstand zu erfassen?
- c) Welche Methoden werden in den Klassen benötigt, damit die Objekte bei der Ausführung von Spielzügen, ihren Zustand ändern?
- d) Entwerfen Sie ein UML-Diagramm für die Klassen ihres Online-Spiels und tragen Sie in das Diagramm, die Attribute und Methoden der benötigten Objekte ein.

VL09, Aufgabe 4 (Praktikum)**(☆☆)**

Erstellen Sie in Eclipse ein neues Projekt mit dem Namen `EidP-VL09-Aufgabe4`. 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 die Kontonummer und die Werte von Kontostand und Dispokredit durch Kommata getrennt als Zeichenkette zurückgeben.
- b) Programmieren Sie die Klasse `GirokontoTest`, in deren Hauptprogramm (`main`) mehrere Objekte der Klasse `Girokonto` erzeugt werden. Außerdem sollen in der `main` Methoden alle Methoden mindestens einmal aufgerufen werden. Die Methoden `auszahlen` und `ueberweisung` sollen sogar 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);`