

Programmierung 2 — Übungsblatt 1

Abgabe: Dienstag, 5. Mai 2009, 11.45 Uhr

Aufgabe 1.1: Objekte, Klassen und die reale Welt

Betrachten Sie den folgenden vereinfachten Ausschnitt der realen Welt: Es gibt Bankinstitute, Kunden und Girokonten.

1. Machen Sie sich für diesen Weltausschnitt klar: Welche Objekte existieren und welche Charakteristiken (Zustand, Verhalten, Identität) besitzen sie? Welche Klassen existieren und welche Charakteristiken (Attribute und Verhalten) besitzen sie? Welche Beziehungen gibt es zwischen den Objekten und Klassen, die von Ihnen identifiziert worden sind?
2. Identifizieren Sie für jede Klasse einige wesentlichen Attribute und Verhaltensformen (Anzahl: jeweils 3 bis 4)!
3. Analog den Beispielen aus der Vorlesung („Ein Auto“, „Ein GUI-Element“) geben Sie die Objekte (jeweils Zustand, Verhalten und Identität) für ein Szenario an, in dem drei Kunden bei zwei verschiedenen Banken in der Summe fünf Girokonten besitzen!
4. Notieren Sie die Klassen, die dem „3 Kunden - 2 Bankinstitute - 5 Girokonten“-Szenario zugrunde liegen, als Java-Klassen. Hinweis: Aus der Vorlesung bekannt: Klassen sind Typen, somit sind Klassennamen Typnamen, die in Klassenattributen verwendet werden dürfen. Neben den Klassen, die von Ihnen identifiziert wurden, verwenden Sie nur die String-Klasse, um die restlichen Attribute von Klassen zu spezifizieren.

Aufgabe 1.2: Objekte, Klassen: Formalisierung

1. Formalisieren Sie Ihre Java-Beschreibung des Szenarios aus der Aufgabe 1.1 (→ Menge der Klassen und Objekte, Abbildungen *kls*, *eig* und *attr*)!
2. Zeigen Sie für ein Kundenobjekt und zwei verschiedene Namen durch Anwenden dieser Abbildungen, dass ihre festgelegten Abbildungen die „Diagramm - kommutiert“-Eigenschaft erfüllen (analog für ein Bankinstitutobjekt und zwei von Ihnen gewählte Namen; analog für ein Girokontoobjekt und zwei von Ihnen gewählte Namen)!

Aufgabe 1.4.1: Nutzung von Backus-Naur-Form

1. Welche der folgenden Namen sind laut der Grammatik aus der Vorlesung gültige Bezeichner (Begründung falls ungültig)?
 - Informatik
 - 4711_Girokonto
 - kUNDE_10
 - _
Hinweis: ein Unterstrich
 - www.uni-saarland.de
 - anna@File_Server
 - _____
Hinweis: vier Unterstriche

- 0_03_2008_Korrek56TUR_

2. Welche der folgenden Zeichenfolgen sind laut der Grammatik aus der Vorlesung syntaktisch gültig bzw. ungültig (Begründung!)?
 - `class kunde { }`
 - `class kunde }`
 - `class { }`

Aufgabe 1.4.2: Erstellung von Backus-Naur-Form

1. Setzen Sie folgende Beschreibung in BNF um: Eine Liste besteht aus einem oder mehreren Nichtterminalen R , welche durch Kommata getrennt sind. Am Ende der Sequenz darf sich kein Komma befinden.
2. Setzen Sie folgende Beschreibung für eingeschränkte Bezeichner in BNF um: Ein gültiger Bezeichner besteht aus einem bis acht Zeichen. Erlaubte Zeichen sind Großbuchstaben und Ziffern. Das erste Zeichen muss ein Großbuchstabe sein. Wie lautet die Grammatik, wenn Bezeichner genau acht Zeichen lang sein müssen?
3. Schreiben sie eine Grammatik, die alle möglichen Palindromwörter erzeugt. Palindromwörter sind Wörter, die von vorne und hinten gelesen gleich lauten, z.B. LAGERREGAL, ROTOR.
4. Setzen Sie folgende Beschreibung für korrekt geklammerte Ausdrücke in eine Grammatik um: Ein Ausdruck ist korrekt geklammert, falls er genau so viele schließende wie öffnende Klammern besitzt und wenn – von links gelesen – sich stets vor einer schließenden Klammer eine zugehörige öffnende Klammer befindet. So ist `((()))` korrekt geklammert, `((())` und `)()()` sind es jedoch nicht.

Aufgabe 1.5: Methoden-Syntax, Sichtbarkeit

1. Ist die folgende Klassendeklaration syntaktisch korrekt? Falls nicht, begründen Sie Ihre Antwort!

```
class Klasse {
    void Methode_1(String par1, String par2) {
        System.out.println(par1 + par2);
    }
    String Methode_2() {
        System.out.println("Ich bin Methode 2!");
        return "Methode 2 kehrt zurück";
    }
    void Methode_3() {
        System.out.println("Ich bin Methode 3!");
    }
}
```

2. Nennen sie mindestens vier Syntaxfehler im folgenden Programmfragment:

```
class Klasse {{
    void_Methode_1(String par1_String_par2) {
        System.out.println(par1_String_par2);
    };
    String Methode_2 {
        System.out.println("Ich bin Methode 2!");
        return "Methode 2 kehrt zurück";
    }
    void Methode_3()
        System.out.println("Ich bin Methode 3!");
}
```

3. Was wird beim Aufruf von *aufruf()* ausgegeben?

```
class Hallo {
    String x = "Text 1";

    void textAusgabe(String x) {
        System.out.println(x);
    }
    void aufruf() {
        textAusgabe("Text 2");
    }
}
```

4. Was wird beim Aufruf von *aufruf2("Text2")* ausgegeben?

```
class Hallo {
    String x = "Text 1";

    void textAusgabe(String xx) {
        System.out.println(x);
    }
    void aufruf2(String x) {
        textAusgabe(x);
    }
}
```