

Vorlesung „Programmieren“

Wintersemester 2007/2008

8. Übungsblatt

1. Syntaxbaum (5 Punkte)

Zeichnen Sie den Syntaxbaum für $(3*5*6+7)/9+10$

2. Gültigkeitsbereich von Bezeichnern (10 Punkte)

Gegeben sei folgendes Programm:

```
class A
{
    int A;
    int aA;
    void A (int A, char a)
    {
        System.out.print ('=');
        this.A = 2;
        while (this.A < A)
        { System.out.print (a); this.A = this.A+1; }
        System.out.println ('=');
    }

    void a (int A, int a, char b)
    {
        A (A, '=');
        int aA = 2;
        while (aA < a)
        { A (A, b); aA = aA+1; }
        A (A, '=');
    }

    public void aA ()
    {
        A A=new A();
        A.a(10, 15, '?');
    }
}
```

Aufgabe:

Benennen Sie die Bezeichner für Variablen, Parameter, Methoden und Klassen so um, dass eindeutige Bezeichner verwendet werden, aber die Ausführungssemantik nicht verändert wird.

3. Statische Komponenten (Programmieraufgabe) (10 Punkte)

Aufgaben:

- Schreiben Sie eine Klasse **SparKonto**. Modellieren Sie dabei die Kontonummer, die letzte vergebene Kontonummer, den Inhaber, den Kontostand und den aktuellen Sparzinssatz als Komponenten. Überlegen Sie dabei, welche Größen statisch und welche dynamisch zu deklarieren sind, sowie gegebenenfalls welche als konstant. Initialisieren Sie statische und oder

konstante Größen geeignet und implementieren Sie mindestens einen Konstruktor, um die dynamischen Komponenten zu initialisieren. (4 Punkte)

- b) Implementieren Sie die Methoden **void setzeAktuellerSparZinssatz(double zinssatz)**, **gibKontoStand()**, **einzahlen(double betrag)**, **abheben(double betrag)** und **zahleZinsen()**. Welche Methoden sind statisch und welche nicht? (6 Punkte)

Bemerkungen:

- Dieses ist eine Programmieraufgabe. Bitte geben Sie die Lösung *sowohl* auf **Papier** als auch den Java-Quelltext in elektronischer Form per **Email** an Ihren Tutor ab.

4. Modellierung (Programmieraufgabe) (20 Punkte)

Wir wollen ein Programm schreiben, das die Arbeit einer Bibliothek simuliert.

Die Bibliothek verwaltet Bücher und Zeitschriften. Die Klasse **Buch** speichert folgende Informationen ab: Signatur, Autor, Titel, Verlag, Erscheinungsjahr, Regalnummer und Regalposition. Eine erste Version der Klasse könnte wie folgt aussehen:

```
public class Buch {
    private String signatur;
    private String autor;
    private String titel;
    private String verlag;
    private int erscheinungsjahr;
    private int regalnummer;
    private int regalposition;
    //...
}
```

Für eine Zeitschrift speichern wir beim Buch die folgenden Informationen ab: Signatur, Titel, Verlag, Erscheinungsjahr, Regalnummer und Regalposition. Die Information Autor ist irrelevant; dafür hat eine Zeitschrift eine Zeitschriftennummer (ganze Zahl).

- a) Implementieren Sie eine Klasse **AusleihGegenstand**, die die Gemeinsamkeiten von Buch und Zeitschrift implementiert. Schreiben Sie die Klassen **Buch** und **Zeitschrift** als Unterklassen dieser Klasse. Implementieren Sie geeignete Konstruktoren in allen 3 Klassen, um Datenfelder zu initialisieren und Methoden, um auf die Daten zuzugreifen. (6 Punkte)
- b) Fügen Sie der Klasse **AusleihGegenstand** eine Methode **String gibInfo()** hinzu, die alle Informationen zu einem Objekt ausgibt. Überschreiben Sie die Implementierung in den Unterklassen, um spezifische Daten zu einem Buch oder einer Zeitschrift ausgeben zu können. Für ein Buch sollte das Ergebnis etwa so aussehen: "Buch; Titel = Thinking in Java; Verlag = Prentice Hall; Jahr = 1998; Signatur = Prog 2001-93; Regalnummer = 5; Regalposition = 12; Autor: B. Eckel". (4 Punkte)
- c) Schreiben Sie eine Klasse **Bibliothek**, die die Ausleihgegenstände verwaltet. Überlegen Sie sich, welche Datenstruktur am besten geeignet ist, die Ausleihgegenstände zu speichern. Implementieren Sie zunächst folgende Methoden:
- **void add(AusleihGegenstand ag)**: nimmt ein Buch oder eine Zeitschrift in die Bibliothek auf. (2 Punkte)
 - **AusleihGegenstand gibAusleihGegenstand(String signatur)**: sucht nach dem Buch oder der Zeitschrift mit der gegebenen Signatur. Das gefundene Objekt oder null wird zurückgegeben je nach Ergebnis der Suche. (2 Punkte)
- d) Schreiben Sie eine Methode **void gibAlleInfo()**, die alle Informationen zu allen Ausleihgegenständen einer Bibliothek auf die Konsole ausgibt. (4 Punkte)
- e) Implementieren Sie eine Testmethode in der Klasse **Bibliothek**, die ein Bibliothek-Objekt, 3 Buch-Objekte und 2 Zeitschrift-Objekte erzeugt. Diese Ausleihgegenstände werden zur Bibliothek hinzugefügt sowie mit der Methode **gibAlleInfo()** ausgegeben. (2 Punkte)

Bemerkungen:

- Dieses ist eine Programmieraufgabe. Bitte geben Sie die Lösung *sowohl* auf **Papier** *als auch* den Java-Quelltext in elektronischer Form per **Email** an Ihren Tutor ab.
-

Bemerkungen:

- Jede Seite soll oben rechts den Namen der Abgebenden und die Übungsgruppennummer (wichtig!) enthalten.
- Lösungen für die Übungsaufgaben sind (in der Regel) zu zweit abzugeben.
- Kommentieren Sie Ihre Lösungen! Besteht eine Lösung aus mehreren Zetteln, so sind diese zusammen zu heften. Bitte keine Hüllen, Mappen, o.ä..
- Bitte schicken Sie *Programmieraufgaben zusätzlich zur Abgabe auf Papier in elektronischer Form per Email* an Ihren jeweiligen Tutor.
- Kommentieren Sie ihren Quelltext bei Programmieraufgaben. Dabei sollen keine Trivialitäten kommentiert werden, also bitte keine Kommentare wie

~~x=5; // Wir weisen nun der Variablen x den Wert 5 zu~~

sondern sinnvolle Kommentare, die Ideen des Quelltextabschnittes beschreiben oder auf Unteraufgaben (z. B. a), b), ...) hinweisen.

- **Hinreichende Bedingung für die Zulassung zur Klausur:** 50% der erreichbaren Punkte bei jedem Übungszettel (bis auf zwei) und einmaliges Vorrechnen in der Übung
- **Zertifikatskriterium:** Das Bestehen der Klausur am Ende des Semesters

Abgabetermin: Freitag, 11.1.2008, nach der Vorlesung