

Hochschule Pforzheim	
Fakultät für Technik	
Bereich: Informationstechnik	
Fach: Technische Informatik	Studiensem.:
SS 09	Datum: 8.7.09
Prüfer: Prof. Dr. Th. Greiner	
Bearbeitungszeit: 40 Minuten	20 Punkte
Hilfsmittel: Darstellung der Sprachsyntax ohne Diagramme	
Hinweis: Geben Sie auf allen Blättern Matrikelnummer und Name an.	
Beginnen Sie für jede Aufgabe auf einer neuen Seite	
Name, Vorname:	
Matrikel-Nr.:	

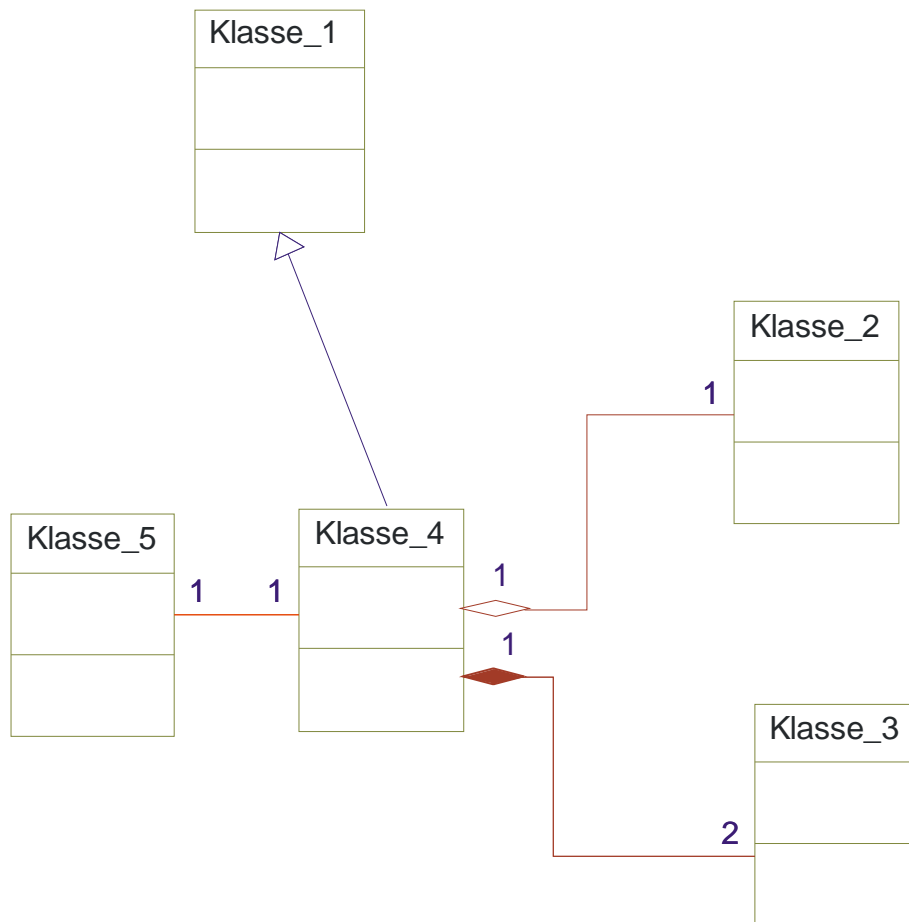
1. Aufgabe (2 Punkte)

Erklären Sie die folgenden Begriffe:

- a) static in Verbindung mit Variablen
- b) IndexOutOfBoundsException
- c) Deadlock
- d) Integer.parseInt

2. Aufgabe (6 Punkte)

Es ist folgendes UML-Klassendiagramm gegeben:



- a) Beschreiben Sie mit einigen Sätzen den Zusammenhang der Klassen.
- b) Erstellen Sie Java-Klassen, sodass dieser Zusammenhang wiedergegeben wird.

3. Aufgabe (4 Punkte)

Schreiben Sie ein Java-Programm, das 2 Threads nutzt, um auf einer gemeinsamen Integer-Variablen `a` zu arbeiten. Der erste Thread erhöht die Variable um 2, der zweite Thread erniedrigt die Variable um 1. Ist der Zählerwert 10 erreicht, werden die Threads beendet.

- a) Welche grundsätzliche Schwierigkeit kann beim Verändern der Variable auftreten?
- b) Erstellen Sie das Java-Programm unter Verwendung der Klasse „Thread“, sodass sichergestellt ist, dass beide Threads alle Veränderungen mitbekommen.

4. Aufgabe (5 Punkte)

Es soll ein Java-Programm erstellt werden, das eine „sichere“ Division durchführt. Erstellen Sie eine entsprechende Klasse „Division“ mit der zugehörigen Methoden „Dividieren“, welche zwei beliebige Integerzahlen `a` und `b` zur Division nutzt und das Divisionsergebnis sowie den Divisionsrest in einem einzigen Datentyp zurückgibt.

5. Aufgabe (3 Punkte)

Erläutern Sie kurz jede Zeile des nachfolgenden Programmtextes:

```
class StopThread1 implements Runnable
{
    private boolean running = true;
    public StopThread1()
    {
        // Hier Starten über den Konstruktor
        Thread t = new Thread(this);
        t.start();
    }
    public synchronized void stopRunning()
    {
        running = false;
    }

    public synchronized boolean isRunning()
    {
        return running;
    }

    public void run()
    {
        int i = 0;
        while(isRunning())
        {
            i++;
            System.out.println("Hello World (" + i + ")");
        }
    }
}
```