

Technische Universität Braunschweig

Institut für Programmierung und Reaktive Systeme

Programmieren II

Dr. Werner Struckmann

3. September 2007

Name:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Vorname:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Matrikelnummer:

--	--	--	--	--	--	--

Kennnummer:

Anrede: Frau Herr

angestrebter Abschluss: Bachelor Master Diplom Magister

Fachrichtung: Informatik Wirtschaftsinformatik IST Mathematik

Mobilität u. Verkehr Physik Maschinenbau Sonstiges: _____

Die Bearbeitungszeit beträgt 120 Minuten. Die Klausur besteht aus 5 Aufgaben. Sie haben die Klausur bestanden, wenn Sie mindestens 40 von 80 möglichen Punkten erreicht haben.

Aufgabe	1	2	3	4	5	Σ
max. Punkte	20	12	12	20	16	80
Punkte						

Note:

Bitte prägen Sie sich Ihre Kennnummer gut ein. Aus Datenschutzgründen wird das Klausurergebnis nur unter dieser Kennnummer bekannt gegeben. Aus den gleichen Gründen können Ergebnisse weder telefonisch noch per E-Mail mitgeteilt werden.

Die Ergebnisse der Klausur erfahren Sie ab dem 14. September 2007 auf der WWW-Seite zu dieser Veranstaltung. Ihre Klausur können Sie am

Dienstag, den 18. September 2007,

von 9:00–11:00 Uhr und von 13:00–15:00 Uhr im Raum 251 des Informatikzentrums einsehen.

Aufgabe 1: Unter dem *Modalwert* einer Folge versteht man denjenigen Wert, der in der Folge am häufigsten auftritt. Beispielsweise ist 5 der Modalwert der Folge

3, 12, 12, 5, 5, 0, -3, 0, 5, -5, 12.

Der Modalwert einer Folge ist i. Allg. nicht eindeutig bestimmt.

Schreiben Sie eine Methode `static int modal(int[] a)`, die ein Feld ganzer Zahlen als Parameter erhält und den Modalwert des Felds als Rückgabewert liefert. Falls der Modalwert der Folge nicht eindeutig bestimmt ist, kann ein beliebiger Modalwert zurückgegeben werden. Erläutern Sie kurz die Idee Ihres Algorithmus.

20 Punkte

Lösung:

Aufgabe 2: Bitte kreuzen Sie an. Für jede richtige Antwort erhalten Sie einen Punkt, für jede falsche Antwort wird ein Punkt abgezogen. Kein Kreuz bzw. zwei Kreuze bedeuten 0 Punkte. Die minimale Gesamtpunktzahl für diese Aufgabe beträgt 0 Punkte. Alle Fragen dieser Aufgabe beziehen sich auf Java.

	wahr	falsch
Die Klasse <code>String</code> ist <code>final</code> .	<input type="checkbox"/>	<input type="checkbox"/>
Werte einer Enum-Klasse können nicht verändert werden.	<input type="checkbox"/>	<input type="checkbox"/>
Enum-Klassen können nicht instanziiert werden.	<input type="checkbox"/>	<input type="checkbox"/>
Wrapper-Klassen sind unveränderbar (immutable).	<input type="checkbox"/>	<input type="checkbox"/>
Von anonymen Klassen lassen sich keine Objekte erzeugen.	<input type="checkbox"/>	<input type="checkbox"/>
Von lokalen Klassen lassen sich keine Objekte erzeugen.	<input type="checkbox"/>	<input type="checkbox"/>
Ein Konstruktor darf nicht den Modifikator <code>private</code> besitzen.	<input type="checkbox"/>	<input type="checkbox"/>
Serialisierung ist eine Möglichkeit, Persistenz von Daten zu erreichen.	<input type="checkbox"/>	<input type="checkbox"/>
Typebounds schränken die möglichen Parameter generischer Klassen ein.	<input type="checkbox"/>	<input type="checkbox"/>
Klassen und Schnittstellen können gleichzeitig als Typebound auftreten.	<input type="checkbox"/>	<input type="checkbox"/>
Ein Semaphore kann den Wert 0 annehmen.	<input type="checkbox"/>	<input type="checkbox"/>
Die Methoden <code>wait</code> und <code>notify</code> können nur Threads ausführen, die sich in einem Monitor befinden.	<input type="checkbox"/>	<input type="checkbox"/>

12 Punkte

Aufgabe 3: Beschreiben Sie kurz die Aufgaben der folgenden Komponentenklassen.

JLabel: Darstellung von Texten und Bildern

JButton: _____

JCheckBox: _____

JRadioButton: _____

JTextArea: _____

JTextField: _____

JComboBox: _____

12 Punkte

Aufgabe 4: Gegeben sei die folgende Klasse `LinkedList` zur verketteten Implementierung linearer Listen ganzer Zahlen.

```
public class LinkedList {  
  
    private int item = -1;  
  
    private LinkedList next;  
  
    public void insert(int x) {  
        LinkedList l = new LinkedList();  
        l.item = x;  
        l.next = next;  
        this.next = l;  
    }  
  
    public String toString() {  
        return next == null ? "." : " ! " + next.item + next;  
    }  
  
}
```

a) Was gibt das folgende Programmfragment aus?

```
LinkedList l = new LinkedList();  
l.insert(-8);  
l.insert(3);  
l.insert(-5);  
l.insert(3);  
l.insert(0);  
System.out.println(l);
```

- b) Ändern Sie die Methode `void insert(int x)` so ab, dass die Elemente in absteigend sortierter Reihenfolge eingefügt werden.
- c) Schreiben Sie eine Methode `int summe()`, die die Summe der eingefügten Elemente der aktuellen Liste liefert. Wenn keine Elemente eingefügt wurden, soll das Ergebnis 0 sein. Durch die Berechnung der Summe darf die Liste nicht verändert werden.

Schreiben Sie Ihre Lösung zu dieser Aufgabe bitte auf Seite 6.

20 Punkte

Aufgabe 5: Gegeben sei die folgende Java-Methode:

```
static int f(int n) {
    assert n >= 1;          // Vorbedingung P
    int s = 1,
        i = 0;
    assert ...;           // Schleifeninvariante Q
    while (s < n) {
        s = 2*s;
        i = i + 1;
        assert ...;       // Schleifeninvariante Q
    }
    assert ...;           // Nachbedingung R
    return i;
}
```

- a) Welchen Wert berechnet diese Methode? Formulieren Sie eine entsprechende Nachbedingung R .
- b) Beweisen Sie die partielle Korrektheit des Programms bezüglich P und R , indem Sie eine geeignete Schleifeninvariante Q angeben.
- c) Formulieren Sie Q und R als Java-Ausdrücke, sodass diese in den obigen assert-Anweisungen verwendet werden können.

16 Punkte

Lösung:

Lösung zu Aufgabe 4