

Klausur		WS 99/00		Bearbeitungsdauer					Prof. Dr. Franzen	
Programmieren 3 (Java)		28. Jan. 2000		110 min					FB MNI	
Aufgabe	1	2	3	4	5	6	7	8	Summe	Note
Max. Pkt.	10	8	8	9	8	7	39	11	100	
Err. Pkt.										

Name, Vorname:

Matrikelnummer:

Ich bestätige die Richtigkeit der oben gemachten Angaben.

Unterschrift:

1. Programmstruktur.

- Warum kann ein Java Programm auf jeder Plattform ausgeführt werden?
- Wieso kann ein Applet mit fast jedem Browser ausgeführt werden?
- Warum braucht ein Applet keine Methode `void main(String[] args)`?
- Welche Methode eines neuen Applets ruft ein Browser zuerst, welche beim Verlassen der HTML-Seite?
- Warum verwendet man eine `import`-Anweisung?

(2 + 2 + 2 + 2 + 2 P.)

2. Typen.

- Wenn eine Klasse *C* eine Schnittstelle *I* implementiert, welche Methoden von *I* müssen dann in *C* oder einer Vorgängerklasse von *C* implementiert sein?
- Kann man einen Feldtyp definieren, dessen Element-Typ ein Schnittstellentyp ist? Ist also z.B. `ActionListener[]` ein zulässiger Typ?
- Welche Typen werden in Java unterschieden? Zu welcher Kategorie gehört ein Feldtyp?

(2 + 1 + 5 P.)

3. Speicherbereinigung.

- Was bedeutet automatische Speicherbereinigung (*garbage collection*)?
- Warum sollte ein Java Laufzeitsystem eine automatische Speicherbereinigung besitzen?
- Warum ist ein Laufzeitsystem mit automatischer Speicherbereinigung für ein Echtzeitsystem fast immer ungeeignet?

(3 + 2 + 3 P.)

4. Bibliotheken.

- Wozu dient ein `LayoutManager`, wie z.B. die Klasse `java.awt.BorderLayout`?
- Wie erkennen Sie bei einer Unterklasse von `java.awt.Component`, welche Ereignisse eines ihrer Objekte erzeugen kann?
- Wozu dient die Methode `String toString()`?
- Besitzt jedes Objekt eine Methode `String toString()`?

(3 + 3 + 2 + 1 P.)

5. Threads.

- a. Jeder Thread besitzt die Methoden `start()` und `run()`. Was bewirken Sie? Wann wird die Methode `run()` aufgerufen?
- b. Können mehrere nebenläufige Threads auf ein einzelnes `String`-Objekt beliebig zugreifen, oder müssen sie diese Zugriffe synchronisieren, um Wettbewerbssituationen (*race conditions*) auszuschliessen? Begründen Sie Ihre Antwort! Ändert sich etwas, wenn statt auf ein `String`-Objekt auf ein `StringBuffer`-Objekt zugegriffen wird?

(3 + 5 P.)

6. Ausnahmen.

- a. Welche Ausgabe bewirkt ein Aufruf von `foo()` im untenstehenden Programm?

```
public static foo(){
    char c = 'a';
    int i = 0;
    try {
        i = i/i;
        c = 'b'
    }
    catch (Error e)      { System.out. println("Fehler");}
    catch (ArithmeticException e){System.out. println("Teiler gleich 0");}
    catch (Exception e) { System.out. println("Ausnahme");}

    finally { System.out.println(c);}
}
```

- b. Kann man im folgenden Programmfragment Anweisungen für den `try`-Block finden, so dass die letzte `catch`-Anweisung ausgeführt wird, also ein Aufruf von `foo()` zur Ausgabe von `werfbar` führt? Wenn ja, wie lauten sie, wenn nein, begründen Sie Ihre Antwort.

```
public static foo(){
    try{
        .....
        .....
    }
    catch (Error e)      { System.out. println("Fehler");}
    catch (Exception e) { System.out. println("Ausnahme");}
    catch (Throwable t) { System.out. println("Werfbar");}
}
```

(3 + 4 P.)

7. Gegeben sei die folgende Schnittstelle:

```
interface Liste {
    // fügt ein Objekt am Anfang der Liste ein
    public void einfügen (Object obj);

    // hängt ein Objekt an das Ende der Liste
    public void anhängen (Object obj);

    // liefert neue Liste mit denen gleichen Elementen, aber in umgekehrter Reihenfolge
    public Liste reverse();

    // liefert true, falls Liste leer, false sonst
    public boolean istLeer();

    // liefert das letzte Element, wenn existent, null sonst
    public Object letztesElement();

    // liefert das erste Element, wenn existent, null sonst
    public Object erstesElement();

    public boolean enthältElement(Object o);
    public int länge();

    // Darstellung als String: beginnt mit "<", gefolgt von einer String-Darstellung
    // für jedes Element, jede getrennt durch Leerräume, endet mit ">"
    public String toString();
}
```

- a. Schreiben Sie eine Klasse `DoppeltVerketteteListe`, die diese Schnittstelle implementiert.

Hinweis: Führen Sie eine geschachtelte Top-Level Klasse `Element` mit Attributen `inhalt`, `nachfolger` und `vorgänger` ein.

- b. Ergänzen Sie die Klassendefinition um ein `main`-Programm, in dem eine Liste mit den 4 Elementen 1, 4, 'a', "Element" erzeugt wird und diese Liste in umgekehrter Reihenfolge mittels der Methode `toString()` auf der Standardausgabe als `<Element a 2 1>` ausgegeben wird.

Hinweis: Beachten Sie, dass `int`- und `char`-Werte nicht vom Typ `Object` sind! Abhilfe?

(34 + 5 P.)

Lösungsvorschlag

-
8. Schreiben Sie ein Programm namens `Head`, das die ersten zehn Zeilen einer auf der Kommandozeile angegebenen Datei ausgibt.

(11 P.)

Lösungsvorschlag