



# Klausur Einführung in die Informatik II für Elektrotechniker

27. Februar 2006

Name: .....

Matr.-Nr. ....

**Bearbeitungszeit: 120 Minuten**

**Bewertung** (bitte offenlassen :-)

Aufgabe	Punkte	Erreichte Punkte
1	7	
2	7	
3	5	
4	9	
5	8	
6	8	
Summe	44	

Spielregeln (**Jetzt lesen!**):

- Benutzen Sie für die Lösung der Aufgaben **nur** das mit diesem Deckblatt ausgeteilte Papier. Lösungen, die auf anderem Papier geschrieben werden, können **nicht** bewertet werden. Schreiben Sie ihre Lösung auch auf die Rückseiten der Blätter; benötigen Sie für eine Lösung mehr als ein Blatt, finden Sie am Ende der Klausur Leerblätter. Zusätzliches Papier können Sie von den Tutoren bekommen.
- Tragen Sie jetzt (vor Beginn der eigentlichen Bearbeitungszeit!) auf **allen** Blättern ihren Namen und ihre Matrikelnummer ein.
- Schreiben Sie deutlich! Unleserliche oder zweideutige Lösungen können nicht gewertet werden.
- Schreiben Sie **nicht** mit Bleistift und **nicht** mit rotem oder grünem Stift (das sind die Farben für die Korrektur).
- Lesen Sie die Aufgaben jeweils bis zum Ende durch; oft gibt es hilfreiche Hinweise!
- Kommentare kosten Zeit; kommentieren Sie ihr Programm nur da, wo der Code alleine nicht verständlich wäre.
- Wir weisen noch einmal darauf hin, dass die Benutzung von Taschenrechnern und anderen elektronischen Hilfsmitteln nicht gestattet ist.

**Viel Erfolg!**

---

**Aufgabe 1 (7 Punkte) Theorie.**

1. (2 Punkte) Nennen Sie je einen *Vorteil* und einen *Nachteil* der Verwendung

- (a) eines *sortierten Arrays* und
- (b) eines *Suchbaumes*

zur Verwaltung von Daten.

2. (3 Punkte) Beschreiben Sie **kurz** das Grundprinzip des Sortierverfahrens *Heapsort*. Welchen (Zeit-)Aufwand und welches Speicherplatzverhalten (*in-situ* bzw. *nicht in-situ*) hat Heapsort?

---

3. (1 Punkt) Was verstehen Sie unter dem Begriff *Gültigkeitsbereich* in einer Programmiersprache?

4. (1 Punkt) Wozu dienen in Programmiersprachen definierte *Konstanten*, wie sie z.B. in Java mit dem Schlüsselwort `final` deklariert werden können?

**Aufgabe 2 (7 Punkte) Java.**

1. (2 Punkte) Was wird auf dem Bildschirm ausgegeben, wenn folgendes Java-Programm mit dem Kommando  
java ListProg  
ausgeführt wird?

```
1 class ListProg {
2     public static void main(String[] args) {
3         L l = new L();
4         l.f();
5     }
6 }
7 class L {
8     void f() {
9         g(new Cell(3, new Cell(2, new Cell(1, new Cell(0, null))));
10    }
11    void g(Cell c) {
12        Terminal.println(c.next.next.value);
13    }
14 }
15 class Cell {
16     int value;
17     Cell next;
18     Cell(int value, Cell next) {
19         this.value = value; this.next = next;
20     }
21 }
```

2. (1 Punkt) Weshalb wird im Konstruktor der Klasse Cell aus der vorigen Teilaufgabe das Schlüsselwort `this` benutzt?

3. (1 Punkt) Beschreiben Sie den Unterschied zwischen den folgenden beiden Methoden:

```
void m1() {}  
abstract void m2();
```

4. (3 Punkte) Welche Fehler enthält folgende Java-Klasse? Geben Sie jeweils die Zeilennummer an und beschreiben Sie den Fehler. Folgefehler (also Fehler, die aus anderen Fehlern resultieren) sollen ignoriert werden.

```
1  interface Atmend {  
2      public void keuchen();  
3  }  
4  abstract class Hund implements Atmend {  
5      abstract void hecheln();  
6      public void keuchen() {  
7          int x = hecheln();  
8      }  
9  }  
10 class Dackel extends Hund implements Hund {  
11 }
```

**Aufgabe 3 (5 Punkte) Numerik.**

Die Exponentialfunktion  $e^x$  kann folgendermaßen definiert werden:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

1. (5 Punkte) Schreiben Sie eine Methode

```
double exponential(double x)
```

welche die Exponentialfunktion anhand der oben angegebenen Formel berechnet. Brechen Sie den Approximationsprozess ab, sobald eine Genauigkeit von 4 Nachkommastellen erreicht ist.

*Hinweis: Sie dürfen die Methode `double pow(double x, double n)` aus der Klasse `Math` benutzen um die Potenz  $x^n$  zu berechnen. Die Fakultätsfunktion*

$$\begin{aligned} 0! &= 1 \\ (n+1)! &= (n+1) \cdot n! \end{aligned}$$

*müssen Sie aber selbst definieren.*

**Aufgabe 4 (9 Punkte) Abstrakte Datentypen.**

In dieser Aufgabe soll ein abstrakter Datentyp `KundenListe` zum Speichern von Kundendaten erstellt werden, welcher Objekte der folgenden Klasse verwalten kann:

```
class Kunde {  
    String kdnr;           // Kundennummer  
    String name;          // Name des Kunden  
    String tel;           // Telefonnummer  
}
```

Der abstrakte Datentyp soll sich immer in einem der folgenden beiden Zuständen befinden:

1. Modus „Nummer“. In diesem Modus kann nach Kundennummern gesucht werden.
2. Modus „Name“. In diesem Modus kann nach Namen gesucht werden.

Sie sollen die Klasse `KundenListe` mit Methoden zum Einfügen und Suchen schreiben, wobei die Suchmethode immer den aktuellen Zustand der Liste berücksichtigen soll und entweder nach Kundennummern oder Namen suchen soll.

1. (3 Punkte) Schreiben Sie die Klasse `KundenListe`. Definieren Sie die benötigten Attribute, um die oben beschriebenen Operationen schreiben zu können. Schreiben Sie einen geeigneten Konstruktor, mit dem man eine leere Kundenliste erzeugen kann.

*Hinweis: Die Art und Weise, wie die Kundenobjekte intern von der Klasse `KundenListe` verwaltet werden, ist Ihnen überlassen.*

2. (2 Punkte) Schreiben Sie (in der Klasse `KundenListe`) eine Methode

```
void einfuegen(Kunde k)
```

welche das Kundenobjekt `k` an einer beliebigen Stelle in die Liste einfügt.

3. (1 Punkt) Schreiben Sie (ebenfalls in der Klasse `KundenListe`) eine Methode

```
void setModus(int m)
```

welche die Kundenliste in den Modus setzt, der durch den Parameter `m` bestimmt wird. Dabei soll folgende Kodierung realisiert werden:

<code>m</code>	Modus
0	„Nummer“
1	„Name“

*Hinweis: Gehen Sie davon aus, dass die Methode immer mit einem gültigen Parameter aufgerufen wird.*

4. (3 Punkte) Schreiben Sie (ebenfalls in der Klasse `KundenListe`) eine Methode

`String suchen(String s)`

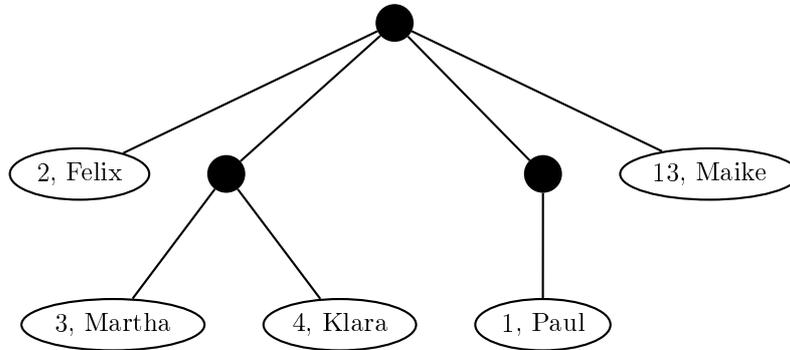
welche in der Kundenliste nach einem Kunden sucht, dessen Kundennummer oder Name mit dem übergebenen String `s` übereinstimmt. Welches der beiden Attribute verglichen wird, wird vom aktuellen Zustand der Liste festgelegt. Wenn die Suche erfolgreich ist, soll die dazugehörige Telefonnummer zurückgegeben werden, andernfalls geben Sie den Leerstring ("") zurück.

*Hinweis: Strings müssen mit der Methode `boolean equals(String other)` aus der Klasse `String` verglichen werden, nicht mit dem Operator `==`.*

**Aufgabe 5 (8 Punkte) Bäume.**

In dieser Aufgabe soll eine Datenstruktur für Bäume mit beliebiger Verzweigungsstruktur geschrieben werden. Jeder innere Knoten im Baum kann beliebig viele Kinder haben (auch gar keine). Werte werden nur an den Blättern gespeichert.

Die folgende Abbildung zeigt, wie solch ein Baum konzeptionell aussieht. Die inneren Knoten (schwarz dargestellt) besitzen beliebig viele Nachfolger, die Blätter (oval) besitzen eine Zahl und einen String als Attribute.



1. (3 Punkte) Schreiben Sie eine Klasse **Baum** zur Darstellung von Baumknoten. Jeder Knoten hat sowohl beliebig viele Referenzen auf Kindknoten als auch einen Schlüssel vom Typ **int** sowie einen Dateneintrag vom Typ **String**. Bei inneren Knoten werden nur die Referenzen auf Kindknoten benutzt, bei Blättern nur der Schlüssel und der String.

Schreiben Sie zwei geeignete Konstruktoren, einen für innere Knoten und einen für Blätter.

2. (3 Punkte) Schreiben Sie (in der Klasse `Baum`) eine Methode

```
String suche(int s)
```

welche nach dem Schlüssel `s` in dem Knoten und allen Kindknoten sucht und das dazugehörige Datenelement zurückliefert. Sollte der Schlüssel nicht im Baum vorkommen, soll `null` zurückgegeben werden.

3. (2 Punkte) Schreiben Sie (ebenfalls in der Klasse `Baum`) eine Methode

```
void tauschen(int s, String n)
```

welche im Baum den Eintrag mit dem Schlüssel `s` so ändert, dass hinterher der Parameter `n` das Datenelement ist. Wenn kein Element mit dem gegebenen Schlüssel existiert, soll der Baum nicht verändert werden.



---

3. (2 Punkte) Schreiben Sie eine Klasse `BildNachricht`, welche von `Nachricht` abgeleitet ist. Diese Klasse soll ein Bild enthalten, das als `byte`-Array dargestellt ist. Schreiben Sie einen geeigneten Konstruktor. Die Zusammenfassung einer Bildnachricht besteht einfach aus dem String `"Ein Bild"`.

4. (2 Punkte) Schreiben Sie eine Klasse `MultiNachricht`, welche von `Nachricht` abgeleitet ist. Diese Klasse soll beliebig viele andere Objekte der Klasse `Nachricht` enthalten können. Schreiben Sie einen geeigneten Konstruktor. Die Zusammenfassung einer Multinachricht soll entweder die Zusammenfassung der ersten Textnachricht sein, falls eine vorhanden ist. Wenn keine Nachricht oder nur Bildnachrichten enthalten sind, soll einfach der String `"Eine Multinachricht"` zurückgegeben werden.



Fak. ET/Inform.  
Klausur InfET II  
27. Februar 2006

Name: .....

Matr.-Nr. ....

---



Fak. ET/Inform.  
Klausur InfET II  
27. Februar 2006

Name: .....

Matr.-Nr. ....

---