

# Einführung in die Informatik I Informatik I/A

Wintersemester 2004/05

## 12. Übungsblatt(Probeklausur)

### Aufgabe 1: Kurzfragen zu Java und Objekt-Orientierter Programmierung

- Geben Sie einen Java-Ausdruck an, der ein zweidimensionales Array anlegt, welches 20 mal 20 Werte vom Typ `double` fassen kann. (3 Punkte)
- Nennen Sie zwei wesentliche Nachteile als `public` bzw. öffentlich deklariertes Instanzvariablen. (3 Punkte)

### Aufgabe 2: Programmverständnis

Finden Sie die Fehler in der folgenden Java-Methode und korrigieren Sie diese, indem Sie die Methode korrigiert abschreiben. Was macht die korrigierte Methode? (8 Punkte)

```
public void methode(Char[] args){
    boolean b=1;
    Ausgabe ausgabe;
    for(i=0; i<args.size(); i++){
        if(args[i]=args[args.size()-i-1])
        {
        } else {
            b=false;
            ausgabe.println("Funktioniert_nicht!");
            Break
        }
    }
    if(b)ausgabe.println("Funktioniert!");
}
```

### Aufgabe 3: Algorithmen - allgemein

Bei der Kommunalwahl 2003 gab es in Lübeck folgende Ergebnisse:

Partei	Stimmen	Anteil
CDU	42669	50.0 %
SPD	27637	32.4 %
GRÜNE	7870	9.2 %
FDP	4448	5.2 %
sonstige	2645	3.2 %
gültige Stimmen	85269	100%
ungültige Stimmen	1063	

Um die Wahlbeteiligung zu erhöhen, soll ein computergestütztes Wahlsystem erstellt werden, welches einen Stimmzettel als Java-Objekt der folgenden Klasse darstellt:

```
public class Stimmzettel{
    public boolean cdu = false;
    public boolean spd = false;
    public boolean gruene = false;
    public boolean fdp = false;
}
```

Der Einfachheit halber werden hier die *sonstigen* Parteien vernachlässigt. Eine Partei erhält eine Stimme, indem die entsprechende boolesche Komponente auf *true* gesetzt wird. Es ist möglich eine ungültige Stimme abzugeben, indem nichts oder mehr als eine Partei 'angekreuzt' werden.

- Legen Sie einen Beispiel-Stimmzettel an, welcher eine **ungültige** Stimme repräsentiert.  
(4 Punkte)
- Schreiben Sie eine statische Java-Methode, die ein Array von Stimmzettel-Objekten erhält und die Anzahl der gültigen Stimmen errechnet.  
Die Methode soll anschliessend die prozentuale Verteilung der gültigen Stimmen auf die Parteien berechnen und ( auf dem Bildschirm ) ausgeben. Zudem soll die Anzahl der ungültigen Stimmen ausgegeben werden.  
(16 Punkte)

#### Aufgabe 4: Algorithmen - rekursiv

- a) Schreiben Sie eine rekursive Java-Methode `div`, die die ganzzahlige Division zweier ganzer Zahlen berechnet. Es kann dabei vorausgesetzt werden, dass die Zahlen positiv sind und der Teiler jeweils ungleich Null ist. (6 Punkte)
- b) Zeichnen Sie den Aufrufbaum für `div(9, 3)`. (2 Punkte)
- c) Schreiben Sie eine rekursive Java-Methode `istPrim`, die prüft, ob eine positive ganze Zahl eine Primzahl ist. (8 Punkte)
- d) Zeichnen Sie den Aufrufbaum für `istPrim(5)`. (4 Punkte)

#### Aufgabe 5: Vererbung

Tische und Stühle sollen als Gegenstände in Räumen modelliert werden. Sie besitzen eine feste Inventarbezeichnung(String) und einem aktuellen Wert(int).

- a) In der Java-Klasse `Raum` sollen Gegenstände zusammen in einem Sammlungsobjekt gespeichert werden.

```
public class Raum{
    private ArrayList gegenstaende;

    public Raum(){
        this.gegenstaende = new ArrayList();
    }
}
```

Damit die `Raum`-Klasse funktioniert, soll eine Klassenhierarchie entworfen werden, die aus der Superklasse `Gegenstand` und den Subklassen `Tisch` und `Stuhl` besteht. Zeichnen Sie ein Klassendiagramm, das auch `Raum` mit beinhaltet und in Beziehung setzt. Definieren Sie die drei Klassen in Java. Vergessen Sie die Konstruktoren nicht! Schreiben Sie für die `Raum`-Klasse eine geeignete Methode `registriere`, die Gegenstände in einem Raum registrieren kann. (12 Punkte)

- b) Schreiben Sie für die Klasse `Raum` eine Methode, die den Wert aller registrierten Gegenstände zurückliefert. Entwickeln Sie dafür (an geeigneter Stelle) eine Methode `gibAktuellenWert`, die den aktuellen Wert eines Gegenstandes zurückliefert. Zeichnen Sie die neuen Methoden auch in das Klassendiagramm aus Teilaufgabe a) ein. (5 Punkte)
- c) Überlegen Sie sich für die Klasse `Tisch` sowie `Stuhl` jeweils eine besondere Instanzvariable und zeichnen Sie diese auch in das Klassendiagramm ein. Nun soll in der Klasse `Raum` eine Methode `ausgeben` implementiert werden, die alle registrierten Gegenstände ausgibt. Entwickeln Sie hierfür an geeigneter Stelle eine Methode `ausgeben` für Gegenstände. (9 Punkte)

## Hinweise

- Kommentieren Sie Ihre Lösungen !
- Die erste Seite soll oben rechts die Namen der beiden Abgebenden und die Übungsgruppennummer (wichtig!) enthalten. Besteht eine Lösung aus mehreren Zetteln, so sind diese zusammen zu heften. Bitte keine Hüllen, Mappen, o.ä.
- Die Programmieraufgaben können in den Rechnerräumen der Häuser 21 und 64 auf dem Universitätsgelände bearbeitet werden (siehe Unix-Anleitung).
- Fragen technischer Art können in der '**betreuten Rechnerzeit**' Mittwochs von 18h bis 20h in den entsprechenden Rechnerräumen gestellt werden.
- Zertifikat/Scheinkriterium: Für den Schein/das Zertifikat sind 50% der erreichbaren Punkte bei den Übungsaufgaben, zweimaliges Vorrechnen in der Übung und das Bestehen der Klausur am Ende des Semesters erforderlich.

---

**Abgabetermin:** Freitag, 28. Januar 2005, nach der Vorlesung.