

Einführung in die Informatik für Ingenieure und Naturwissenschaftler
Einführung in die Informatik für Wirtschaftsmathematiker
Einführung in die Informatik für Mediziner

Klausur am 7. Februar 2019

Bitte kreuzen Sie hier an, nach welcher Prüfungsordnung Sie studieren:

- Bachelor Logistik
 Bachelor Wirtschaftsingenieurwesen
 Bachelor Wirtschaftsmathematik
 Bachelor Medizinphysik

 andere: _____

- Semester: erstes
 höheres: _____

Hinweise

- Diese Klausur besteht aus 10 Aufgaben. Diese Klausur ist bestanden, wenn 40 der möglichen 100 Punkte erreicht werden.
- Für die Bearbeitung dieser Klausur stehen insgesamt 120 Minuten zur Verfügung.
- **Bevor** Sie mit der Bearbeitung dieser Klausur beginnen, **müssen** Sie auf allen Blättern **Ihren Namen und Ihre Matrikelnummer** eintragen.
- Bei der Bearbeitung der Klausur dürfen **keine Hilfsmittel** verwendet werden.
- Schreiben Sie mit einem dokumentenechten **blauen oder schwarzen Stift**. Korrekturflüssigkeiten und ähnliche Materialien dürfen nicht benutzt werden. Bitte trennen Sie die Blätter nicht.
- Bearbeiten Sie jede Aufgabe möglichst auf der Seite, auf der sie steht. Wenn der Platz nicht reicht, können Sie die Zusatzseite 17 nutzen. Notieren Sie in jedem Fall, wo die Lösung zu finden ist.
- Wenn Sie zusätzliche Blätter benötigen, wenden Sie sich an die Aufsicht. Die Nutzung von eigenem Papier ist nicht gestattet.
- Da es verschiedene Sprachgebräuche und unterschiedliche Ausprägungen einzelner Definitionen gibt, werden hier ausdrücklich die Ihnen aus der Vorlesung und den Übungen bekannten Ausdrucksweisen und Realisierungen zu Grunde gelegt.

Viel Erfolg!

Aufgabe	1	2	3	4	5	6	7	8	9	10	Σ
Mögliche Punkte	8	10	8	8	10	10	8	10	8	20	100
Erreichte Punkte											

Aufgabe 1 – Binärsystem

(8 Punkte)

- a) In dieser Aufgabe sollen Sie Zahlen vom Binär- ins Dezimalsystem und umgekehrt umrechnen. In jeder Zeile der unten stehenden Tabelle soll der äquivalente Wert in Binär- und Dezimalschreibweise stehen. Ergänzen Sie die freien Felder der Tabelle entsprechend (es kommen keine negativen Zahlen vor).

(6 Punkte)

Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Dezimalzahl
1	0	0	0	1	0	
1	0	1	0	0	0	
0	0	0	1	1	0	
						63
						50
						14

- b) Vervollständigen Sie die Wahrheitstabelle für die logische **und**-Funktion mit den beiden Eingängen **x** und **y** und dem Ausgang **f**.

(1 Punkt)

x	y	f
false	false	
false	true	
true	false	
true	true	

- c) Vervollständigen Sie die Wahrheitstabelle für die logische **oder**-Funktion mit den beiden Eingängen **x** und **y** und dem Ausgang **f**.

(1 Punkt)

x	y	f
false	false	
false	true	
true	false	
true	true	

Aufgabe 2 – Signatur von Methoden und Deklaration von Arrays

(10 Punkte)

a) Bestimmen Sie die Signaturen für folgende öffentliche, statische Methoden. Eine Implementierung der Methoden ist dabei nicht erforderlich. (je 1 Punkt)

i) Die Methode `mult` multipliziert drei Integerzahlen miteinander und gibt deren Produkt zurück.

ii) Die Methode `equals` vergleicht zwei Strings auf ihre Gleichheit und gibt das Ergebnis zurück.

iii) Die Methode `a` gibt ein 'a' zurück.

iv) Die `main`-Methode.

v) Die Methode `round` rundet eine reelle Zahl, sodass sie keine Nachkommastelle mehr besitzt und gibt die gerundete Zahl zurück.

vi) Die Methode `matrix` erstellt eine $m \times n$ Matrix, die aus Integerzahlen besteht, wobei m und n ganzzahlig sind. Die erstellte Matrix wird zurückgegeben.

b) Deklarieren Sie folgende Arrays.

(je 1 Punkt)

i) Deklarieren Sie ein `int`-Array namens `arr`.

ii) Deklarieren Sie ein Array `b` vom Typ `double` mit der Länge 10.

iii) Deklarieren und initialisieren Sie ein `int`-Array `five` mit den Zahlen von 1 bis einschließlich 5.

iv) Deklarieren Sie ein Array `cake` und initialisieren Sie das Array mit den drei Strings `apple`, `banana` und `cherry`.

Aufgabe 3 – Methodenaufrufe

(8 Punkte)

In dieser Aufgabe sollen Sie die folgenden Methoden realisieren. Hierzu stehen Ihnen folgende Hilfsmethoden zur Verfügung. Verwenden Sie zur Realisierung der Methode eine geeignete Auswahl aus den Hilfsmethoden, die alle öffentlich und statisch sind.

- `int max(int x, int y)` gibt das Maximum zweier Zahlen zurück.
- `int max(int[] a)` gibt die größte Zahl eines Arrays zurück.
- `int min(int[] a)` gibt die kleinste Zahl eines Arrays zurück.
- `int sum(int[] a)` gibt die Summe aller Elemente eines Arrays zurück.
- `double sqrt(int x)` berechnet die Wurzel von x und gibt das Ergebnis zurück.
- `void printMax(int[] a)` gibt die größte Zahl eines Arrays aus.

a) Schreiben Sie eine öffentliche, statische Methode `max`, welche das Maximum aus den zwei `int`-Arrays `a` und `b` berechnet (3 Punkte)

b) Schreiben Sie eine öffentliche, statische Methode `average`, welche den Durchschnitt eines `int`-Arrays `a` berechnet. (3 Punkte)

c) Wie nennt sich der Mechanismus in Java, mit dem es möglich ist, in einer Klasse mehrere Methoden mit dem Namen `max` zu deklarieren? (2 Punkte)

Aufgabe 4 – Schleifen

(8 Punkte)

- a) Schreiben Sie das folgende Programmfragment so um, dass anstelle der `for`-Schleife eine `while`-Schleife verwendet wird. Die Funktionalität des Programmfragmentes muss erhalten bleiben. (2 Punkte)

```
int a = 0;
for (int z = 10; z >= 0; z--) {
    a = a + 2;
}
System.out.println("a: " + a);
```

- b) Schreiben Sie das folgende Programmfragment so um, dass anstelle der `do-while`-Schleife eine `while`-Schleife verwendet wird. Die Funktionalität des Programmfragmentes muss erhalten bleiben.

(3 Punkte)

```
int x = 7;
int p = 256;
do {
    p = p - 20;
    x--;
} while (x > 0);
System.out.println("p: " + p);
```

- c) Schreiben Sie das folgende Programmfragment so um, dass anstelle der `while`-Schleife eine `for`-Schleife verwendet wird. Die Funktionalität des Programmfragmentes muss erhalten bleiben.

(3 Punkte)

```
int b = 0;
int n = 7;
int e = 1;
while (b < 15 || n < 9) {
    e = e + b * n - 2;
    b = b + 2;
}
System.out.println("e: " + e);
```

Aufgabe 5 – Felder

(10 Punkte)

Schreiben Sie eine öffentliche, statische Methode `minOf2`, die das Minimum aus zwei `int`-Arrays berechnet und zurückgibt. Beide Arrays sind nicht leer, können aber unterschiedliche Größen besitzen. Greifen Sie **nicht** auf die Hilfsmethoden aus Aufgabe 3 zurück.

Aufgabe 6 – Rekursion

(10 Punkte)

- a) Programmieren sie eine öffentliche, statische Methode `f`, die die unten abgebildete mathematische Definition für ganzzahlige x und ganzzahlige $y > 0$ umsetzt:

$$f(x, y) = \begin{cases} 1 & \text{falls } y = 0 \\ f(x, y/2) \cdot f(x, y/2) & \text{falls } y \text{ gerade} \\ x \cdot f(x, y - 1) & \text{sonst} \end{cases}$$

Wählen Sie für die Methode einen geeigneten Argument- und einen geeigneten Rückgabotyp. (5 Punkte)

- b) Berechnen Sie die Funktion f mit folgenden Werten: (je 2 Punkte)

i) $f(3, 2) =$ _____

ii) $f(2, 3) =$ _____

- c) Was berechnet die Funktion f ? (1 Punkt)

Aufgabe 7 – Bäume

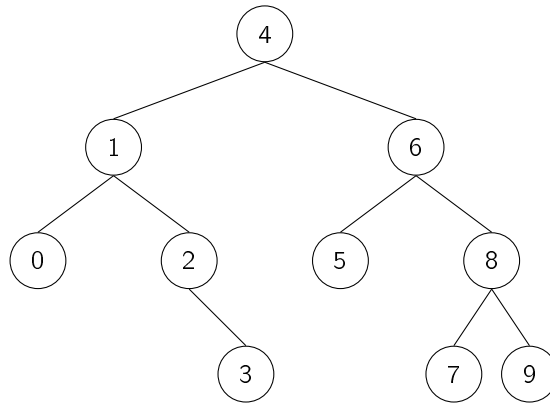
(8 Punkte)

- a) Erstellen Sie einen binären Suchbaum, indem Sie die Zahlen 12, 17, 21, 22, 8, 14, 10, 23, 19, 5, 24, 9, 2, 11, 25 in der angegebenen Reihenfolge in einen binären Suchbaum eintragen, der zu Beginn leer ist.

(4 Punkte)

b) Betrachten Sie folgenden Binärbaum:

(je 1 Punkt)



i) Geben Sie die Pre-Order-Reihenfolge für obigen Baum an:

ii) Geben Sie die In-Order-Reihenfolge für obigen Baum an:

iii) Geben Sie die Post-Order-Reihenfolge für obigen Baum an:

iv) Warum liefert der In-Order-Durchlauf eine aufsteigend sortierte Folge der Bauminhalte?

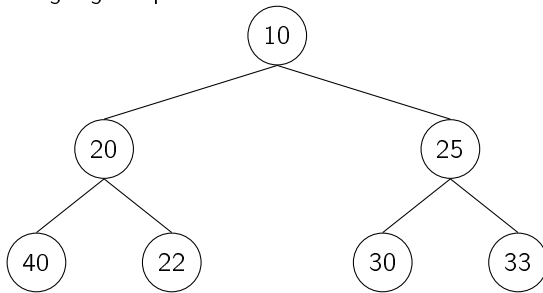
Aufgabe 8 – Heaps

(10 Punkte)

- a) Führen Sie auf den folgenden Minimum-Heaps jeweils die angegebene Operation aus. Zeichnen Sie in den Teilaufgaben i) und ii) jeweils den Heap auf, der nach der Durchführung der entsprechenden Operation entsteht. (je 2 Punkte)

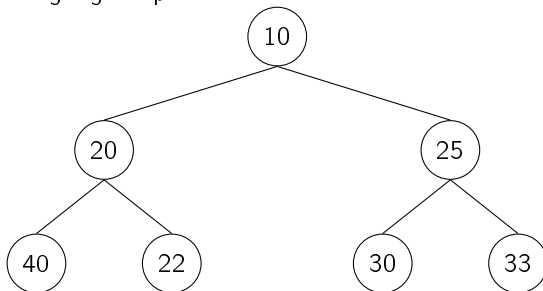
- i) Entfernen des Minimums

Ausgangsheap:



- ii) Einfügen von 5

Ausgangsheap:



b) Zeichnen Sie den Heap für folgendes Array:

(2 Punkte)

[0, 1, 8, 6, 12, 9, 10, 22, 19, 17, 20, 11, 23]

c) Kreuzen Sie an, ob die folgenden Aussagen über Minimum-Heaps wahr oder falsch sind. Es wird die in der Vorlesung gegebene Heapdefinition angenommen. (Richtige Antwort: 0,5 Punkte, falsche Antwort: -0,5 Punkte, keine Antwort 0 Punkte; die Aufgabe gibt insgesamt mindestens 0 Punkte.) (2 Punkte)

Aussage

wahr falsch

Ein Heap ist ein rechtsvollständiger binärer Baum.

Ein Feld, das einen Minimum-Heap repräsentiert, muss immer aufsteigend sortiert sein.

Die Markierung eines Knotens ist immer größer als die Markierungen seiner Kinder.

Die Markierung des rechten Kindes eines Knotens ist immer größer als die des linken Kindes.

d) Welchen Vorteil besitzt Heapsort gegenüber dem naiven Sortieralgorithmus aus der Vorlesung (Insertionsort)? (2 Punkte)

Aufgabe 9 – Klassenvariablen und -methoden

(8 Punkte)

Ergänzen Sie die Klasse `Account` um einen privaten Instanzenzähler `accountCounter`, der die Anzahl der instantiierten Konten einer Bank zählt. Ergänzen Sie die Klasse zudem mit einer `getter`-Methode, die den Wert des Zählers zurückgibt.

```
public class Account{

    private int balance;
    private String name;

    public Account(String name) {
        this.name = name;
        balance = 0;
    }

    public String getName() {
        return this.name;
    }

    public int getBalance() {
        return this.balance;
    }

    public int depositMoney(int money) {
        this.balance = this.balance + money;
        return this.balance;
    }

    public int withdraw(int money) {
        if(this.balance - money >= 0) {
            this.balance = this.balance - money;
        }
        return this.balance;
    }
}
```

Aufgabe 10 – Parkplatz

(20 Punkte)

- a) Schreiben Sie eine öffentliche Klasse `Parkplatz`, die einen Parkplatz darstellt. Die Klasse soll folgende Eigenschaften erfüllen: (10 Punkte)
- i) Alle Methoden (auch der Konstruktor) sind öffentlich.
 - ii) Zu einem Parkplatz gehören die Attribute `plaeetze` (für die Anzahl der vorhandenen Plätze) und `belegt` (für die Anzahl der Plätze, die belegt sind). `belegt` ist nur für die Klasse zugreifbar; `plaeetze` ist nur für die Klasse und Unterklassen zugreifbar.
 - iii) Dem Konstruktor wird die Anzahl der vorhandenen Plätze übergeben. Es sind anfangs keine Plätze belegt.
 - iv) Die Klasse hat eine Methode `parken`, die die Anzahl der belegten Plätze um eins erhöht, wenn noch freie Plätze vorhanden sind. Ist der Parkplatz bereits voll, liefert die Methode `false` zurück, ansonsten `true`.
 - v) Die Klasse hat eine Methode `ausparken`, die die Anzahl der belegten Plätze um eins verringert. Ist kein Platz belegt, so geschieht nichts. Die Methode liefert die Anzahl der Plätze zurück, die nach dem Ausparken noch frei sind.
 - vi) Die Klasse hat eine Methode `druckeBelegung`, die die aktuelle Belegung und die Anzahl der vorhandenen Plätze ausgibt (Beispiel: "3 von 14 belegt.")

- b) Schreiben Sie eine öffentliche Klasse `Parkhaus`, die von der Klasse `Parkplatz` erbt. Die Klasse soll folgende weitere Eigenschaften haben: (7 Punkte)
- i) Alle Methoden (auch der Konstruktor) sind öffentlich.
 - ii) Die Klasse hat ein zusätzliches privates Attribut `einfahrtFrei`, in dem gespeichert wird, ob die Einfahrt in das Parkhaus möglich ist.
 - iii) Die Klasse hat einen Konstruktor: der Konstruktor ruft zunächst den Konstruktor der Oberklasse auf und setzen dann das Attribut `einfahrtFrei` auf `true`. Der Konstruktor hat als Parameter die Anzahl der vorhandenen Plätze.
 - iv) Die Klasse hat eine Methode `sperrern`, die das `einfahrtFrei`-Attribut auf `false` setzt.
 - v) Die Methode `parken` wird überschrieben. Die neue `parken`-Methode implementiert dieselbe Funktionalität wie die `parken`-Methode der Basisklasse mit dem einzigen Unterschied, dass das Einparken nur dann möglich ist, wenn die Einfahrt frei ist. Falls nicht, liefert die Methode `false` zurück.

- c) Was wird durch die Ausführung des folgenden Quelltextfragments auf dem Bildschirm ausgegeben?
(3 Punkte)

```
Parkplatz p = new Parkplatz(2);
p.parken();
System.out.println(p.ausparken());
System.out.println(p.ausparken());
p.parken();
p.druckeBelegung();

Parkhaus ph = new Parkhaus(500);
ph.sperren();
System.out.println(ph.parken());
ph.druckeBelegung();
```


Platz für Nebenrechnungen und Aufgabenlösungen.

Platz für Nebenrechnungen und Aufgabenlösungen.