

Vorname: XXXXXXXXXX

Diese Klausur ist mein dritter Prüfungsversuch (bitte ankreuzen): Ja  Nein   
 Ich studiere im Diplom- bzw. Bachelorstudiengang Medien: Diplom  Bachelor

Schreiben Sie jede Lösung auf die Vorderseite eines neuen Blattes (und lassen Sie die Rückseiten Ihrer Lösungsblätter leer). Die Aufgaben 5 und 6 stehen auf der Rückseite dieses Blattes!  
 (15/12 Punkte) bedeutet: 15 Punkte für Studiengang MB (Bachelor), 12 für MD (Diplom).

**Aufgabe 1 (15/12 Punkte):** Betrachten Sie die folgenden drei Klassenvereinbarungen:

```

1 class K1 {
2     private int wert;
3     public int getWert() {return wert;}
4     public void setWert(int wert) {
5         if (wert >= 0) this.wert = wert;
6     }
7     public K1(int wert) {setWert(wert);}
8 } // class K1
9
10 class K2 {
11     K1 anna = new K1(10);
12     K1 bert = new K1(20);
13     K1 carl = new K1(30);
14 } // class K2
15
16 class K3
17     static public void main(String[] args) {
18         K1 data = new K1(15);
19         K1 emil = new K1(25);
20         K2 fred = new K2();
21         ...
22     } // class main
23
24     static void main(Object ob) {System.out.println(ob);}
25 } // class K3
    
```

✓ 1.1. Analysieren Sie die Klassen und füllen Sie dann die folgende Tabelle aus. Tragen Sie jeweils nur die (einfachen) Namen der betreffenden Elemente in die Kästchen der Tabelle ein:

	class K1	class K2	class K3
Klassenattribute	—	—	—
Klassenmethoden	—	—	main, plus
Objektattribute	wert	anna, bert, carl	—
Objektmethoden	getWert, setWert	—	—

✓ 1.2. Stellen Sie sich vor, dass der Ausfühler die main-Methode von K3 bis einschließlich Zeile 19 ausgeführt hat. **Wieviele** Module existieren in diesem Moment und wie heißen diese Module? *g*

K3, data, emil, K1

✓ 1.3. Stellen Sie sich vor, dass der Ausfühler die main-Methode von K3 bis einschließlich Zeile 20 ausgeführt hat. **Wieviele** Module existieren in diesem Moment und wie heißen diese Module? Geben Sie die Namen an, mit denen man in der main-Methode auf die Module zugreifen kann (falls nötig also Namen der Form `mmmm.oooo`). *g*

K3, data, emil, fred, anna, fred.bert, fred.carl, K1, K2

✓ 1.4. Stellen Sie sich vor, dass der Ausfühler die main-Methode von K3 bis einschließlich Zeile 20 ausgeführt hat (wie bei der vorigen Teilaufgabe). **Wieviele** internen Variablen existieren in diesem Moment?

5

✓ Aufgabe 2 (15/12 Punkte) In der main-Methode der Klasse K3 der vorigen Aufgabe werden drei Variablen namens `data`, `emil` und `fred` vereinbart. Stellen Sie diese drei Variablen als Bojen dar. Benutzen Sie dabei die **ausführliche** Bojendarstellung (nicht die vereinfachte Darstellung). Wenn Sie Referenzen "erfinden müssen", dann nehmen Sie dafür die Zahlen 110, 120, 130, 140 ... in dieser Reihenfolge.

✓ Aufgabe 3 (20/20 Punkte): Schreiben Sie ein Unterprogramm entsprechend der folgenden Spezifikation:

```

on:
1     static String alsZer(char zahl) {
2         // Liefert einen String der Laenge 16, der die zahl als 2-er-Zahl
3         // (Binärzahl) enthaelt. Achtung: Die Komponenten des Ergebnis-
4         // strings sollen Zeichen {'0' bzw. '1'} sein, nicht die
5         // Zahlen 0 bzw. 1!
6         //
7         // Beispiele:
8         // alsZer('\u0000') ist gleich "0000000000000000"
9         // alsZer('\u0001') ist gleich "1111111111111111"
10        // alsZer('\u0001') ist gleich "0000000001000001"
11        // alsZer('A') ist gleich "0000000001000001"
12        ...
13    } // alsZer
    
```

✓ Aufgabe 4 (20/20 Punkte): Schreiben Sie ein Unterprogramm entspr. der folgenden Spezifikation:

```

1     static int ansAB(String[] arr) {
2         // Wie oft kommt die Zeichenfolge "AB" in den elementaren
3         // Komponenten der Reihung arr ("in den String-Komponenten
4         // von arr") vor? Diese Funktion liefert die Antwort.
5         //
6         // Beispiele: Seien folgende Reihungen vereinbart:
7         // String[] arr01 = {"ABCAB", "XABX", "AB"}, {"XX", "A", ""}, {}
8         // String[] arr02 = {"ABABAB", "ABABABA", "AABABABX", "AA"};
9         // String[] arr03 = {};
10        //
11        // Dann ist ansAB(arr01) gleich 4
12        // ansAB(arr02) gleich 9
13        //
14        //
15        // ansAB
    
```

Aufgabe 5 (15/13 Punkte): Betrachten Sie die folgenden for-Schleifen:

```

1 // Teilaufgabe 5.1.
2 for (int i=0; i<5; i++) {
3     for (int j=0; j<1; j++) p("XX");
4     for (int j=1; j<5; j++) p("OO");
5     println();
6 }
7
8 // Teilaufgabe 5.2.
9 for (int i=5; i>0; i--) {
10    for (int j=0; j<1; j++) p("XX");
11    for (int j=1; j<5; j++) p("OO");
12    println();
13 }
14
15 // Teilaufgabe 5.3.
16 for (int i=1; i<=5; i++) {
17    for (int j=5; j>1; j--) p("XX");
18    for (int j=1; j>0; j--) p("OO");
19    println();
20 }
    
```

Geben Sie für jede Teilaufgabe an, was durch sie zum Bildschirm ausgegeben wird. Die Prozedur p gibt (wie in zahlreichen Beispielprogrammen) ihren String-Parameter zum Bildschirm aus, ohne den Bildschimzeiger (engl. cursor) zum Anfang der folgenden Zeile vorzuschieben. Die parameterlose Prozedur println schiebt den Bildschimzeiger zum Anfang der nächsten Zeile vor.

**Aufgabe 6: (15/13 Punkte)**

- ✓6.1. Erläutern Sie kurz den Unterschied zwischen einem *Sammlungsobjekt* (engl. collection object) und einem *Behälterobjekt* (engl. container object).
- ✓6.2. Geben Sie eine *Art* und eine *Oberart* von Ereignissen an, die in einem Java-Programm im Zusammenhang mit einem *Grabo-Objekt* (oder: einem *Component-Objekt*) auftreten können.
- ✓6.3. Geben Sie mit dem Befehl `println` ein *anonymes Objekt* aus. Den Typ des Objekts dürfen Sie völlig frei wählen.
- ✓6.4. Vereinbaren Sie eine Variable namens `berthold`, die zu irgendeinem *anonymen Typ* gehört.

15  
14  
15  
19  
15  
14  
92

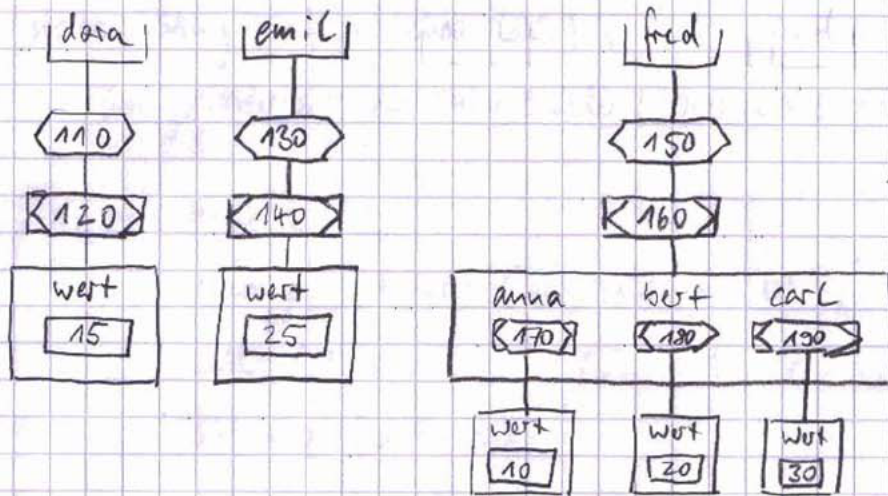
Note 1,3

07.02.06

*lfn.*

Zweitbeste Klausur!  
Glückwunsch!

Aufgabe 2



gut!

Nicht ausführlich -1

14

Aufgabe 3

```

static String alsZer(char zahl) {
    String Builder sb = new String Builder ("000000000000000");
    int s = 15;
    while (true) {
        char z = (char) (zahl % 2 + '0');
        zahl /= 2;
        sb.insert(s--, z);
        if (zahl == 0) break;
    } // while
    return sb + "";
} // alsZer
    
```

Cast (char)

evtl. falsche Länge -5

15

Aufgabe 4

```

static int anzAB(String[][] srr) {
    int anz = 0;
    for (String[] sr : srr) {
        for (String s : sr) {
            for (int i = 0; i < s.length - 1; i++) {
                if (s.substring(i, i+1).equals("AB")) {
                    anz++;
                    i++;
                }
            }
        }
    }
    return anz;
}

```



109

Aufgabe 5

5.1)

0 0 0 0 0 0 0 0 0  
x x 0 0 0 0 0 0 0  
x x x x 0 0 0 0 0  
x x x x x x 0 0 0  
x x x x x x x x 0 0



5.2)

x x x x x x x x x  
x x x x x x x 0 0  
x x x x x x 0 0 0  
x x x x 0 0 0 0 0  
x x 0 0 0 0 0 0 0



5.3)

x x x x x x x 0 0  
x x x x x x 0 0 0  
x x x x 0 0 0 0 0  
x x 0 0 0 0 0 0 0  
0 0 0 0 0 0 0 0 0



15

## Aufgabe 6

6.1) Es gibt Sammlungsobjekte verschiedener Typen, in die man bestimmte Werte tun kann, z.B. int-Werte oder Strings oder andere Sammlungen.

In Behälterobjekte kann man nur Grabo-Objekte packen, z.B. JButtonen oder JCheckBox oder andere Behälter.

Zweiter wichtiger Unterschied:

Ein Objekt kann in mehreren Sammlungen sein, ein Grabo-Objekt ist maximal in genau einem Behälterobjekt.

6.2) Es könnte ein Ereignis der Art mouseClicked auftreten. mouseClicked gehört zur Oberart MouseEreignis (die dazu gehörige Schnittstelle heißt MouseListener und die Adapterklasse MouseAdapter).

6.3) ~~public MouseAdapter~~  
 plus(new String("Hallo"));

6.4) ~~JFrame~~  
 JFrame beithold = new JFrame() {  
 public void add3() {  
 add(new JButton());  
 } // add3

3X

14