

Vorname

Nachname

Matrikel-Nr

Schreiben Sie jede **Lösung** möglichst auf die Vorderseite eines **neuen** Blattes (und lassen Sie die Rückseiten Ihrer Lösungsblätter **leer**).

Aufgabe 1 (15 Punkte): Schreiben Sie ein Unterprogramm entsprechend der folgenden Spezifikation:

```
1  static int[] ziffernHisto(int[] zahlR);
2      // Stellen Sie sich die Komponenten der Reihung zahlR als Dezimal-
3      // zahlen (und ohne ueberfluessige fuehrende Nullen) dargestellt vor.
4      //
5      // Wie oft kommt in all diesen Dezimalzahlen die Ziffer 0 vor?
6      // Wie oft kommt in all diesen Dezimalzahlen die Ziffer 1 vor?
7      // Wie oft kommt in all diesen Dezimalzahlen die Ziffer 2 vor?
8      // ...
9      // Wie oft kommt in all diesen Dezimalzahlen die Ziffer 9 vor?
10     //
11     // Die Funktion ziffernHisto liefert die Antworten auf diese
12     // 10 Fragen in Form einer Reihung mit 10 Komponenten.
13     //
14     // Beispiel: Sei die Reihung ir wie folgt vereinbart:
15     // int[] ir = {1230, 3210, 0, 9999};
16     // Dann muss ziffernHisto(ir) gleich folgender Reihung sein:
17     // {3, 2, 2, 2, 0, 0, 0, 0, 0, 4}
```

Aufgabe 2 (15 Punkte): Schreiben Sie ein Unterprogramm entsprechend der folgenden Spezifikation:

```
1  static String[] durchSchnitt(String[] sr1, String[] sr2) {
2      // Liefert als Ergebnis eine Reihung die alle Strings enthaelt,
3      // die sowohl in sr1 als auch in sr2 vorkommen.
4      //
5      // Beispiel: Sei
6      // String[] srA = {"ab", "cd", "ef"};
7      // String[] srB = {"ef", "ab", "xx"};
8      // Dann muss durchSchnitt(srA, srB) gleich folgender Reihung sein:
9      // {"ab", "ef"}
10     // Dabei ist es gleichgueltig, in welcher Reihenfolge die Komponenten
11     // in dieser Ergebnis-Reihung stehen ({"ef", "ab"} waere genau so
12     // korrekt).
```

Aufgabe 3 (15 Punkte): Betrachten Sie die folgenden Klassenvereinbarungen:

```

1  // -----
2  class KA {
3      static public int anzahl = 0;
4
5      public KA next;
6      public long data;
7
8      public KA(KA next, long data) {
9          this.next = next;
10         this.data = data;
11         anzahl++;
12     } // Konstruktor KA
13 } // class KA
14 // -----
15 class KB {
16     static public void main(String[] susi) {
17         KA anf, end;
18         end = new KA(null, 17);
19         anf = new KA(end, 25);
20         anf = new KA(anf, 38);
21         System.out.println("Anzahl: " + KA.anzahl);
22         ...
23     } // main
24 } // class KB
25 // -----

```

Führen Sie das Programm KB mit Papier und Bleistift aus. Wenn Sie dabei neue Referenzen erfinden müssen, dann verwenden Sie möglichst die Zahlen [1], [2], [3], ... etc. (in dieser Reihenfolge). Wie sehen die Variablen **anf** und **end** in dem Moment aus, in dem der Ausführer zur **Zeile 21** kommt? Zeichnen Sie die Variablen **anf** und **end** und "alles was daranhängt" als Bojen.

Tip: Zeichnen Sie (zumindest erstmal) mit **Bleistift** und möglichst "**großzügig**".

Aufgabe 4 (15 Punkte): Betrachten Sie noch einmal das Programm **KB** aus der vorigen Aufgabe und stellen Sie sich noch einmal den Moment vor, in dem der Ausführer die **Zeile 21** erreicht. Was gilt in diesem Moment? Beantworten Sie die folgenden Fragen entsprechend:

- 4.1. Wieviele **Klassen** hat der Ausführer bis dahin geladen und wie heißen diese Klassen?
- 4.2. Wieviele **Objekte** hat der Ausführer bis dahin erzeugt? Geben Sie für jedes dieser Objekte einen **Namen** an, wenn möglich einen **einfachen** Namen (wie **otto** oder **emil** etc.) und sonst einen **zusammengesetzten** Namen (wie **otto.anna** oder **berta.emil** etc.).
- 4.3. Wieviele **Module** gehören in diesem Moment zum Programm **K3**?
- 4.4. Geben Sie für jeden **Modul** an, **wieviele Konstruktoren**, **Attribute** und **Methoden** er enthält, und wie die Attribute und Methoden **heißen**, etwa so:
 "Der Modul **otto.emil** enthält **2** Konstruktoren, **3** Attribute namens **x**, **y** und **summe** und **2** Methoden namens **berechneSumme** und **halbiere**".
- 4.5. Was gibt der Ausführer zum Bildschirm aus wenn er den Befehl in Zeile 21 ausführt?

Aufgabe 5 (15 Punkte): Betrachten Sie das folgende Java-Programm:

```
1 class Aufgabe5 {
2     // -----
3     static class AusA extends Exception {};
4     static class AusAX extends AusA {};
5     static class AusAY extends AusA {};
6
7     static void pln(Object ob) {System.out.println(ob);}
8     // -----
9     static public void main(String[] susi) {
10         for (int n=1; n<=3; n++) {
11
12             try {
13                 upro(n);
14             } catch (AusAY aus) {
15                 pln("A n: " + n);
16             } catch (AusA aus) {
17                 pln("B n: " + n);
18             } catch (Throwable aus) {
19                 pln("C n: " + n);
20             } finally {
21                 pln("D n: " + n);
22             } // try/catch/finally
23
24             pln("E n: " + n);
25             pln("-----");
26
27         } // for
28     } // main2
29     // -----
30     static public void upro(int n) throws AusA {
31
32         try {
33             switch(n) {
34                 case 3: throw new AusAX();
35                 case 2: throw new AusAY();
36                 case 1: throw new Error();
37                 default: throw new AusA();
38             } // switch
39         } catch (AusA aus) {
40             pln("F n: " + n);
41             throw aus;
42         } // try/catch
43
44     } // upro
45     // -----
46 } // class Aufgabe5
```

Geben Sie an, was dieses Programm **zur Standardausgabe** (d.h. zum Bildschirm) **ausgibt**.

Aufgabe 6 (15 Punkte): Beantworten Sie die folgenden Fragen möglichst kurz. Nur die Frage 6.3. erfordert eine etwas längere Antwort.

- 6.1. Aus **wieviel** Teilen besteht jede **Variable** mindestens und wie **heissen** diese Teile?
- 6.2. Welche **weiteren** Teilen können zu einer **Variablen** gehören?
- 6.3. Was ist ein **Modul**?
- 6.4. Seien **OK** und **UK** zwei Klassen und sei UK eine **Unterklasse** von OK. Zu welcher der beiden Klassen gehören dann (mindestens so viele wie oder) **mehr Objekte**? Zu welcher der beiden Klassen gehören dann (mindestens so viele wie oder) **mehr Elemente**?
- 6.5. Welcher **Buchstabe** (A, B, C) gehört zu welcher **Ziffer** (1, 2, 3)?

- A. Alle **Klassen** mit der **erweitert**-Relation
- B. Alle **Schnittstellen** mit der **erweitert**-Relation
- C. Alle **Pakete** mit der **enthält**-Relation

1. Wald
2. Zyklentreier Graph
3. Baum

Lösung 1:

```

1  static int[] ziffernHisto(int[] zahlR) {
2
3      int[] histo = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
4
5      for (int i=0; i<zahlR.length; i++) {
6          int zahl = Math.abs(zahlR[i]);
7          int ziff;
8
9          while (true) {
10             ziff = zahl % 10;
11             zahl = zahl / 10;
12             histo[ziff]++;
13             if (zahl == 0) break;
14         } // while
15     } // for
16
17     return histo;
18 } // ziffernHisto

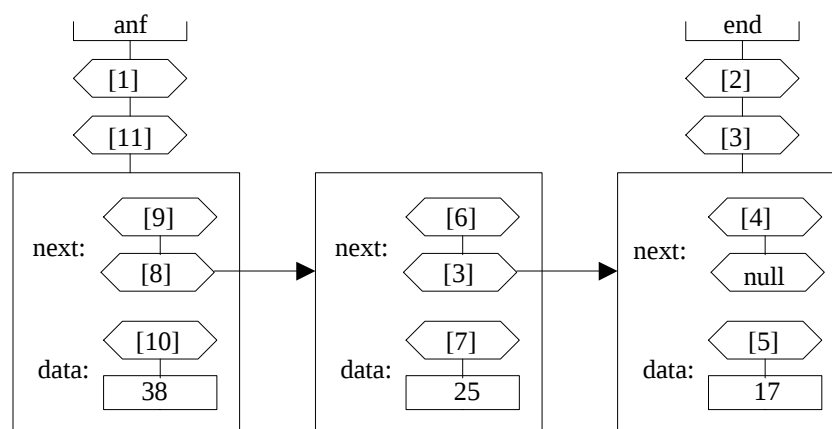
```

Lösung 2:

```

1  static String[] durchSchnitt(String[] sr1, String[] sr2) {
2
3      Vector ergV = new Vector();
4
5      schleife1: for (int i1=0; i1<sr1.length; i1++) {
6          schleife2: for (int i2=0; i2<sr2.length; i2++) {
7              if (sr1[i1].equals(sr2[i2])) {
8                  ergV.add(sr1[i1]);
9                  continue schleife1;
10             } // if
11         } // schleife2
12     } // schleife1
13
14     String[] ergS = new String[ergV.size()];
15
16     for (int i=0; i<ergS.length; i++) {
17         ergS[i] = (String) ergV.get(i);
18     }
19
20     return ergS;
21 } // durchSchnitt

```

Lösung 3:

Lösung 4:

4.1. Wieviele Klassen hat der Ausführer bis dahin geladen und wie heissen diese Klassen?

2 Klassen, KB und KA

4.2. Wieviele Objekte hat der Ausführer bis dahin erzeugt? Geben Sie für jedes dieser Objekte einen Namen an, wenn möglich einen einfachen Namen (wie otto oder emil etc.) und sonst einen zusammengesetzten Namen (wie otto.anna oder berta.emil etc.).

3 Objekte, anf, anf.next und end (oder: anf.next.next)

4.3. Wieviele Module gehören in diesem Moment zum Programm K3?

5 Module

4.4. Geben Sie für jeden Modul an, wieviele Konstruktoren, Attribute und Methoden er enthält, und wie die Attribute und Methoden heissen:

Der Modul KB enthält eine Methode namens main (und einen Standardkonstruktor)

Der Modul KA enthält ein Attribut namens anzahl und einen Konstruktor.

Der Modul anf enthält zwei Attribute namens next und data.

Der Modul anf.next enthält zwei Attribute namens next und data.

Der Modul end (oder: anf.next.next) enthält zwei Attribute namens next und data.

4.5. Was gibt der Ausführer zum Bildschirm aus wenn er den Befehl in Zeile 21 ausführt?

Anzahl: 3

Lösung 5:

```
1 C n: 1
2 D n: 1
3 E n: 1
4 -----
5 F n: 2
6 A n: 2
7 D n: 2
8 E n: 2
9 -----
10 F n: 3
11 B n: 3
12 D n: 3
13 E n: 3
14 -----
```

Lösung 6:

6.1. Aus wieviel Teilen besteht jede Variable mindestens und wie heissen diese Teile?

Aus zwei Teilen, einer Referenz und einem Wert.

6.2. Welche weiteren Teilen können zu einer Variablen gehören?

Ein Name und/oder eine Zielwert

6.3. Was ist ein Modul?

Ein Behälter für Variablen, Unterprogramme und andere Dinge (je nach Programmiersprache), der aus mindestens 2 Teilen besteht, einem öffentlichen (ungeschützten) und einem privaten (geschützten) Teil.

6.4. Seien OK und UK zwei Klassen und sei UK eine Unterklasse von OK. Zu welcher der beiden Klassen gehören dann (mindestens so viele wie oder) mehr Objekte? Zu welcher der beiden Klassen gehören dann (mindestens so viele wie oder) mehr Elemente?

Zur Oberklasse OK gehören mehr Objekte als zur Unterklasse UK.

Zur Unterklasse UK gehören mehr Elemente als zur Oberklasse OK.

6.5. Welcher Buchstabe (A, B, C) gehört zu welcher Ziffer (1, 2, 3)?

- A. Alle Klassen mit der erweitert-Relation
- B. Alle Schnittstellen mit der erweitert-Relation
- C. Alle Pakete mit der enthält-Relation

- 1. Wald
- 2. Zyklensfreier Graph
- 3. Baum

Lösung: A3, B2, C1