

Vorname

Nachname

Matrikel-Nr

Schreiben Sie jede Lösung möglichst auf die Vorderseite eines neuen Blattes (und lassen Sie die Rückseiten Ihrer Lösungsblätter leer). Insgesamt erreichbar sind 90 Punkte.

Aufgabe 1 (15 Punkte): Führen Sie die folgende Befehlsfolge mit Papier und Bleistift aus. Geben Sie als Lösung (nur) an, was zum Bildschirm ausgegeben wird:

```
1      int otto = 1;
2      for (int i=0; i<25; i+=3) {
3          otto = -(2*otto);
4          i    = i + otto;
5          System.out.print(i + " ");
6      }
7      System.out.println();
```

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

```
1      static long summeDerZiffern(String[] sr) {
2          // Liefert die Summe aller Ziffern, die in den Strings in der Reihung
3          // sr vorkommen.
4          // Beispiel: Angenommen, eine Reihung r enthaelt die 3 Strings
5          // "DM 25", "30 Euro 17 Cent" und "Hallo!". Dann hat der Ausdruck
6          // summeDerZiffern(r) den Wert 18 (weil 2 + 5 + 3 + 0 + 1 + 7 gleich
7          // 18 ist).
8          ...
9      } // summeDerZiffern
```

Hinweis: Achten Sie auf den Unterschied zwischen dem Zeichen '0' und der Zahl 0 (für die anderen Ziffern entsprechend). In dieser Aufgabe sollen Zahlen wie z.B. 0, 1, 2, ... summiert werden, nicht Zeichen wie '0', '1', '2'

Aufgabe 3 (15 Punkte): Schreiben Sie ein Unterprogramm entsprechend der folgenden Spezifikation:

```
1      static int nteZiffer(long zahl, char n) {
2          // Liefert die n-te Ziffer der zahl. Beispiele:
3          // nteZiffer(582, 0) ist gleich 2
4          // nteZiffer(582, 1) ist gleich 8
5          // nteZiffer(582, 2) ist gleich 5
6          // nteZiffer(582, 3) ist gleich 0
7          // nteZiffer(582, 7) ist gleich 0
8          ...
9      } // nteZiffer
```

Hinweis: Sie können sicher sein, dass der Parameter n keinen negativen Wert enthält, denn zum Ganzzahltyp char gehören keine negativen Zahlen.

Aufgabe 4 (30 Punkte): Betrachten Sie die folgenden Klassenvereinbarungen:

```

1 class Haupt4 {
2     static public void main(String[] susi) {
3         Neben41 anna = new Neben41();
4         Neben41 bert = new Neben41();
5         ...
6     }
7 } // class Haupt4
8 // -----
9 class Neben41 {
10     static float x = 1.5;
11     Neben42 carl = new Neben42();
12     Neben42 dora = carl;
13 } // class Neben41
14 // -----
15 class Neben42 {
16     static float y = 2.5;
17     long g01 = 17;
18 } // class Neben42

```

Stellen Sie sich den Moment vor, in dem der Ausführer die Zeile 4 fertig ausgeführt hat.

4.1. Wie sehen in diesem Moment die Variablen **anna** und **bert** aus? Stellen Sie die beiden Variablen als **Bojen** dar (wahlweise in ausführlicher oder in abgekürzter Darstellung).

4.2. Wieviele Module existieren in diesem Moment und wie heißen diese Module?

4.3. Geben Sie für jeden der Modul die **Anzahl** und die **Namen** seiner Elemente an!

Aufgabe 5 (15 Punkte): Was gibt das Programm Haupt5 zur Standardausgabe aus?

```

1 import de.tfh_berlin.einaus.AM01;
2 class Ex51 extends Exception {};
3 class Ex52 extends Exception {};
4 // -----
5 class Haupt5 {
6     static public void main(String[] susi) {
7         for (int i=1; i<=2; i++) {
8             try {
9                 met1(i);
10                met2(i);
11            } catch (Ex51 e) {
12                AM01.pln("main: Ausnahme Ex51 trat auf!");
13            } catch (Ex52 e) {
14                AM01.pln("main: Ausnahme Ex52 trat auf!");
15            } // try/catch
16        } // for
17    } // meinMain
18    // -----
19    static public void met1(int n) throws Ex51 {
20        AM01.pln("met1: Anfang!");
21        if (n == 1) throw new Ex51();
22        AM01.pln("met1: Ende!");
23    } // met1
24    // -----
25    static public void met2(int n) throws Ex52 {
26        AM01.pln("met2: Anfang!");
27        try {
28            if (n == 2) throw new Ex52();
29        } catch (Ex52 e) {
30            AM01.pln("met2: Ausnahme Ex52 trat auf!");
31            throw e;
32        } // try/catch
33        AM01.pln("met2: Ende!");
34    } // met2
35 } // class Haupt5

```

Lösung 1: Ausgabe der Befehlsfolge zur Standardausgabe:

```
1  -2 5 0 19 -10 57
```

Lösung 2:

```

1  static long summeDerZiffern(String[] sr) {
2      // Liefert die Summe aller Ziffern, die in den Strings in der Reihung
3      // sr vorkommen.
4      // Beispiel: Angenommen, eine Reihung r enthaelt die 3 Strings
5      // "DM 25", "30 Euro 17 Cent" und "Hallo!". Dann hat der Ausdruck
6      // summeDerZiffern(r) den Wert 18 (weil 2 + 5 + 3 + 0 + 1 + 7 gleich
7      // 18 ist).
8
9      long summe = 0;
10
11     for (int i=0; i<sr.length; i++) {
12         String s = sr[i];
13         for (int j=0; j<s.length(); j++) {
14             char c = s.charAt(j);
15             if ('1' <= c && c <= '9') {
16                 c -= '0';
17                 summe += c;
18             }
19         } // for j
20     } // for i
21     return summe;
22 } // summeDerZiffern

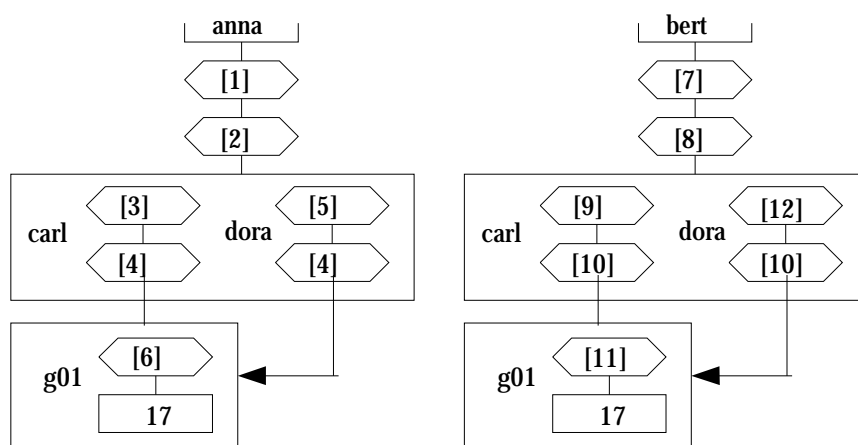
```

Lösung 3:

```

1  static int nteZiffer(long zahl, char n) {
2      // Liefert die n-te Ziffer der zahl. Beispiele:
3      // nteZiffer(582, 0) ist gleich 2
4      // nteZiffer(582, 1) ist gleich 8
5      // nteZiffer(582, 2) ist gleich 5
6      // nteZiffer(582, 3) ist gleich 0
7      // nteZiffer(582, 7) ist gleich 0
8
9      while (n-- > 0) zahl /= 10;
10     return (int) (zahl % 10);
11 } // nteZiffer

```

Lösung 4.1: (15 Punkte) Die Variablen anna und bert in Bojendarstellung:

Lösung 4.2.: (5 Punkte) Wieviele Module existieren in diesem Moment und wie heissen diese Module?

7 Module:

Haupt4, Neben41, Neben42, anna, bert, anna.carl (oder anna.dora), bert.carl (oder bert.dora)

Lösung 4.3.: (10 Punkte) Geben Sie für jeden der Module an, wieviele Elemente er enthält und wie diese Elemente heissen.

Modul Haupt4:	1 Element namens	Haupt4.main.
Modul Neben41:	1 Element namens	Neben41.x.
Modul Neben42:	1 Element namens	Neben42.y.
Modul anna:	2 Elemente namens	anna.carl und anna.dora.
Modul bert:	2 Elemente namens	bert.carl und bert.dora.
Modul anna.carl:	1 Element namens	anna.carl.g01.
Modul bert.carl:	1 Element namens	bert.carl.g01.

Lösung 5: Ausgabe des Programms Haupt5:

```
1 met1: Anfang!
2 main: Ausnahme Ex51 trat auf!
3 met1: Anfang!
4 met1: Ende!
5 met2: Anfang!
6 met2: Ausnahme Ex52 trat auf!
7 main: Ausnahme Ex52 trat auf!
```