

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 long maxDiff(long[] r) {
2     // Berechnet von jedem Paar benachbarter Komponenten der Reihung r
3     // den Betrag ihrer Differenz. Liefert den groessten Betrag als
4     // Ergebnis. Liefert 0 als Ergebnis, falls r weniger als 2 Kompo-
5     // nenten enthaelt.
6
7     // Beispiel 1:
8     // long[] sally = {3, 1, 4, 5};
9     // Die Betraege der Differenzen benachbarter Komponenten sind
10    // 2, 3 und 1. Also ist maxDiff(sally) gleich 3.
11
12    // Beispiel 2:
13    // long[] ludwig = {-2, 2, -3, 3, 2};
14    // Die Betraege der Differenzen benachbarter Komponenten sind
15    // 4, 5, 6 und 1. Also ist maxDiff(ludwig) gleich 6.
16
17    ...
18
19 }
```

Aufgabe 2 (20 Punkte): Schreiben Sie ein Unterprogramm entsprechend der folgenden Spezifikation:

```
1 static ArrayList toArrayList(long[][] rr) {
2     // Wandelt die zweistufige Reihung rr (Reihung von Reihungen von
3     // long-Variablen) in ein entsprechendes "zweistufiges Objekt
4     // der Klasse ArrayList" um und liefert es als Ergebnis.
5     //
6     // Das "zweistufige Ergebnis-Objekt" soll ein ArrayList-Objekt sein,
7     // welches ArrayList-Objekte enthaelt. Jedes dieser ArrayList-Objekte
8     // soll Long-Objekte enthalten.
9
10    ...
11
12 }
```

Aufgabe 3 (10 Punkte): Betrachten Sie die folgenden Klassenvereinbarungen:

```
1 // -----
2 class Ober {
3     void gibAus(String s) {
4         pln("A " + s);
5     }
6
7     void gibAus(int n) {
8         pln("B " + n);
9     }
10    static void pln(Object ob) {System.out.println(ob);}
11 } // class Ober
12 // -----
13 class Unter extends Ober {
14     void gibAus(int n) {
15         pln("C " + n);
16     }
17     static void pln(Object ob) {System.out.println(ob);}
18 } // class Unter
19 // -----
20 class OberUnterTst {
21     static public void main(String[] siegfried) {
22         Ober otto = new Ober();
23         Ober ulla = new Unter();
24
25         otto.gibAus(17);
26         otto.gibAus("Hallo!");
27         ulla.gibAus(17);
28         ulla.gibAus("Hallo!");
29         ulla = otto;
30         ulla.gibAus(17);
31         ulla.gibAus("Hallo!");
32     } // main
33
34     static void pln(Object ob) {System.out.println(ob);}
35 }
```

Was gibt das Programm **OberUnterTst** zur Standardausgabe (zum Bildschirm) aus?

Aufgabe 4 (15 Punkte): Betrachten Sie die folgenden Klassenvereinbarungen:

```

1 // -----
2 class KA {
3     static private int anzahl = 0;
4     static public int getAnzahl() { return anzahl;}
5
6     private String name;
7     private KB next = null;
8
9     public KA(String name) {
10         this.name = name;
11         anzahl++;
12     } // Konstruktor KA
13
14 } // class KA
15 // -----
16 class KB {
17     static private int anzahl = 0;
18     static public int getAnzahl() { return anzahl;}
19
20     private String name;
21     private KA next;
22
23     public KB(String name, KA next) {
24         this.name = name;
25         this.next = next;
26         anzahl++;
27     } // Konstruktor KB
28
29 } // class KB
30 // -----
31 class Test {
32     static public void main(String[] siegfried) {
33         KA ob1 = new KA("Sara");
34
35         KB ob2 = new KB("Bert", ob1);
36         KB ob3 = new KB("Anna", ob1);
37
38         System.out.println("KA.getAnzahl(): " + KA.getAnzahl());
39         System.out.println("KB.getAnzahl(): " + KB.getAnzahl());
40         ...
41     } // main
42 } // class Test

```

Führen Sie das Programm **Test** mit Papier und Bleistift aus. Wie sehen die Variablen **ob1**, **ob2** und **ob3** in dem Moment aus, in dem der Ausführer zur **Zeile 37** kommt? Zeichnen Sie die Variablen **ob1**, **ob2** und **ob3** als Bojen.

Aufgabe 5 (10 Punkte): Betrachten Sie noch einmal das Programm **Test** aus der vorigen Aufgabe und stellen Sie sich noch einmal den Moment vor, in dem der Ausführer die **Zeile 37** erreicht. Was gilt in diesem Moment? Beantworten Sie die folgenden Fragen entsprechend:

- 5.1. Wieviele **Klassen** hat der Ausführer bis dahin erzeugt und wie heissen diese Klassen?
- 5.2. Wieviele **Objekte** hat der Ausführer bis dahin erzeugt? Geben Sie für jedes dieser Objekte einen **Namen** an.
- 5.3. Wieviele **Module** gehören in diesem Moment zum Programm **Test**?
- 5.4. 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** enthält 2 Konstruktoren, 3 Attribute namens **x**, **y** und **summe** und 2 Methoden namens **berechneSumme** und **halbiere**".
- 5.5. Was gibt der Ausführer zum Bildschirm aus, wenn er die Befehle in Zeile 38 und 39 ausführt?

Aufgabe 6 (20 Punkte): Betrachten Sie das folgende Java-Programm:

```

1 class Ex extends Exception {};
2 class Ex01 extends Ex {};
3 class Ex02 extends Ex {};
4 class Ex03 extends Ex {};
5
6 class Ausnahmen {
7     static public void main(String[] siegfried) {
8         for (char c='X'; c<='Z'; c++) {
9             try {
10                 upro01(c);
11             } catch (Ex01 ex) {
12                 pln("A c: " + c);
13             } catch (Ex02 ex) {
14                 pln("B c: " + c);
15             } catch (Ex ex) {
16                 pln("C c: " + c);
17             } finally {
18                 pln("D ---");
19             } // try/catch/finally
20         } // for c
21     } // main
22
23     static void upro01(char c) throws Ex {
24         try {
25             switch(c) {
26                 case 'X': throw new Ex01();
27                 case 'Y': throw new Ex02();
28                 default: throw new Ex();
29             } // switch
30         } catch (Ex02 ex) {
31             pln("E c: " + c);
32         } catch (Ex ex) {
33             pln("F c: " + c);
34             throw ex;
35         } finally {
36             pln("G c: " + c);
37         } // try/switch/finally
38     } // upro01
39
40     static void pln(Object ob) {System.out.println(ob);}
41 } // class Ausnahmen

```

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

Lösung 1 (15 Punkte):

```

1  static long maxDiff(long[] r) {
2
3      long max = 0;
4      for (int i=0; i<r.length-1; i++) {
5          long diff = Math.abs(r[i+1] - r[i]);
6          if (max < diff) max = diff;
7      }
8      return max;
9  } // maxDiff

```

Lösung 2 (20 Punkte):

```

10 static ArrayList toArrayList(long[][] rr) {
11
12     ArrayList aa = new ArrayList();
13
14     for (int i=0; i<rr.length; i++) {
15         long[] r = rr[i];
16         ArrayList a = new ArrayList();
17
18         for (int j=0; j<r.length; j++) {
19             Long wob = new Long(r[j]);
20             a.add(wob);
21         }
22         aa.add(a);
23     }
24     return aa;
25 } // toArrayList

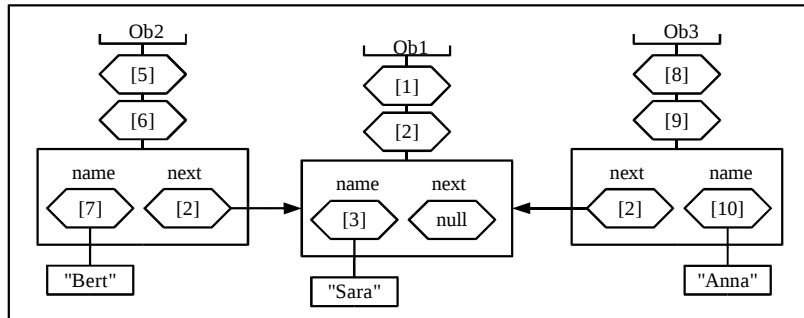
```

Lösung 3 (10 Punkte):

```

26 B 17
27 A Hallo!
28 C 17
29 A Hallo!
30 B 17
31 A Hallo!

```

Lösung 4 (15 Punkte):**Lösung 5 (10 Punkte):**

5.1. Wieviele Klassen hat der Ausführer bis dahin geladen und wie heißen diese Klassen?

Drei Klassen, KB, KA und Test.

5.2. Wieviele Objekte hat der Ausführer bis dahin erzeugt? Geben Sie für jedes dieser Objekte einen Namen an.

Drei Objekte: ob1, ob2 und ob3

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

Sechs Module

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

Der Modul KA enthält einen Konstruktor, eine Methode namens getAnzahl und ein Attribut namens anzahl.

Der Modul KB enthält einen Konstruktor, eine Methode namens getAnzahl und ein Attribut namens anzahl.

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

Der Modul ob1 enthält zwei Attribute namens name und next.

Der Modul ob2 enthält zwei Attribute namens name und next.

Der Modul ob3 enthält zwei Attribute namens name und next.

5.5. Was gibt der Ausführer zum Bildschirm aus wenn er die Befehle in Zeile 38 und 39 ausführt?

KA.getAnzahl(): 1

KB.getAnzahl(): 2

Lösung 6 (20 Punkte):

```

32 F C: X
33 G C: X
34 A C: X
35 D ---
36 E C: Y
37 G C: Y
38 D ---
39 F C: Z
40 G C: Z
41 C C: Z
42 D ---

```