

Vorname

Nachname

Matrikel-Nr

Tragen Sie Ihre Lösungen zu den **Aufgaben 1 bis 3** in das folgende **Formular** ein. Schreiben Sie Ihre Lösungen zu **Aufgabe 4** und **Aufgabe 5** auf extra Blätter und kennzeichnen Sie diese Blätter oben links mit Ihrem **Nachnamen**.

Formular für die **Lösungen** zu den Aufgaben 1 bis 3:

1.1. Anzahl und Namen der Instanzelemente der Klasse Perle ?	1.2. Anzahl und Namen der Klassenelemente der Klasse TstPerle ?
1.3. Anzahl und Namen der Module ?	
1.4. Die vollen Namen aller Elemente in allen Modulen?	
1.5. Ausgabe des Programms TstPerle zum Bildschirm?	
1.6. Zeichnen Sie die Bojen der Variablen p1 , p2 und p3 bitte auf ein extra-Blatt !	
2. Die Nummern der falschen Zeilen in Aufgabe2:	
3.1. Die Ausgabe der Prozedur a31:	3.2. Die Ausgabe der Prozedur a32:
3.3. Die Ausgabe der Prozedur a33:	3.4. Die Ausgabe der Prozedur a34:

Aufgabe 1 (30 Punkte) : Betrachten Sie das folgende Programm und beantworten Sie dann die darunterstehenden Fragen:

```

1 // -----
2 public class Perle {
3     String name;
4     Perle next;
5
6     Perle(String name) {
7         this.name = name;
8     }
9
10    void setNext(Perle next) {
11        this.next = next;
12        System.out.println(next.name + " folgt auf " + this.name);
13    } // setNext
14
15 } // class Perle
16 // -----
17 public class TstPerle {
18     static Perle p1 = new Perle("Anna");
19     static Perle p2 = new Perle("Bert");
20     static Perle p3 = new Perle("Carl");
21
22     static public void main(String[] susi) {
23         System.out.println("TstPerle: Jetzt geht es los!");
24         p1.setNext(p2);
25         p2.setNext(p3);
26         p3.setNext(p1);
27         System.out.println("TstPerle: Das war's erstmal!");
28     } // main
29 } // class TstPerle
30 // -----

```

Frage 1.1.: Wieviele **Instanzelemente** enthält die Klasse **Perle** und wie heißen sie?

Frage 1.2.: Wieviele **Klassenelemente** enthält die Klasse **TstPerle** und wie heißen sie?

Frage 1.3.: Wenn der Ausführer das Programm **TstPerle** bis Zeile 23 ausgeführt hat, **wieviele** Module existieren dann und **wie heißen** diese Module?

Frage 1.4.: Wie heißen die **Elemente** in all diesen Modulen mit **vollem Namen** (z.B. **otto.emil** für ein Element namens **emil** in einem Modul namens **otto**)?

Frage 1.5.: Was gibt das Programm **Betrag** zum Bildschirm aus?

Frage 1.6.: Wie sehen die Variablen **p1**, **p2** und **p3** als Bojen dargestellt aus? Beantworten Sie diese Frage bitte, indem Sie die Bojen **zeichnen** (auf ein extra-Blatt). Tragen Sie dabei in **jedes** sechseckige Kästchen eine **Referenz** ein (z.B. [123] oder [37] etc.).

Aufgabe 2 (10 Punkte): Die folgende Klasse enthält mehrere **falsche Zeilen**, die den Compiler zu einer Fehlermeldung der Art "**Can't make a static reference to ...**" veranlassen. Welche Zeilen sind das? Geben Sie als Lösung nur die **Zeilennummern** der falschen Zeilen an.

```

1 class Aufgabe2 {
2     int n1 = 17;
3     int n2 = 2 * n1;
4     int f1(int i) {return i + n1;}
5     int f2(int i) {return i + f3(i);}
6     static int n3 = n2;
7     static int n4 = f1(5);
8     static int n5 = 33;
9     static int n6 = 2 * n5;
10    static int f3(int i) {return i + n5;}
11    static int f4(int i) {return i + n1;}
12    static int f5(int i) {return i + f1(i);}
13 } // class Aufgabe2

```

Aufgabe 3 (20 Punkte): Jede der folgenden Prozeduren gibt **eine Zeile** zum Bildschirm aus. Geben Sie für jede Prozedur an, wie diese Zeile aussieht.

```
1  // -----
2  static void a31() {
3      int a1 = 3;
4      int a2 = a1++;
5      int a3 = --a1;
6      a1     = a2++;
7      a2     = ++a3;
8      a3     = a1--;
9      System.out.println("Summe: " + (a1 + a2 + a3));
10 } // a31
11 // -----
12 static void a32() {
13     int b1 = 17;
14     int erg = 0;
15     while (b1 > 0) {
16         erg++;
17         b1 /= 2;
18     }
19     System.out.println("erg: " + erg);
20 } // a32
21 // -----
22 static void a33() {
23     int[] ir = {2, 5, 3, 1, 8};
24     int first = 0;
25     int last = ir.length - 1;
26     for (int i=0; i<ir.length/2; i++) {
27         ir[first + i] += ir[last - i];
28         ir[last - i] += ir[first + i];
29     }
30     System.out.print("ir: ");
31     for (int i=0; i<ir.length; i++) {
32         System.out.print(ir[i] + ", ");
33     }
34     System.out.println();
35 } // a33
36 // -----
37 static void a34() {
38     int zahl = 1;
39     for (int i=0; i < 25; i++) {
40         if (i % 2 == 0) continue;
41         if (i % 5 == 0) break;
42         zahl *= 2;
43     }
44     System.out.println("zahl: " + zahl);
45 } // a34
46 // -----
```

Aufgabe 4 (20 Punkte): Schreiben Sie eine Funktion namens **wieOft**, die dem folgenden "Skelett" entspricht:

```
1 // -----
2 static int wieOft(String s) {
3     // Wie oft steht in s eine Ziffer ('0' bis '9') unmittelbar nach
4     // einem kleinen Buchstaben ('a' bis 'z')? Diese Funktion liefert
5     // die Antwort. Falls s eine Null-Referenz ist oder auf einen
6     // leeren String referiert, wird 0 als Ergebnis geliefert.
7     //
8     // Beispiele:
9     // wieOft("a4b5c6d7") liefert den Wert 4
10    // wieOft("abcd4567") liefert den Wert 1
11    // wieOft("4567abcd") liefert den Wert 0
12    // wieOft("") liefert den Wert 0
13    // wieOft(null) liefert den Wert 0
14    ...
15 } // wieOft
16 // -----
```

Aufgabe 5 (20 Punkte): Schreiben Sie eine Funktion namens **erfuelltDrei**, die dem folgenden "Skelett" entspricht und nur **Vereinbarungen** und **einfache Anweisungen** enthält (d.h. in ihrer Lösung darf keine **if**-, **switch**-, **for**-, **while**- oder **do-while**-Anweisung vorkommen. Blockanweisungen sind erlaubt aber nicht nötig):

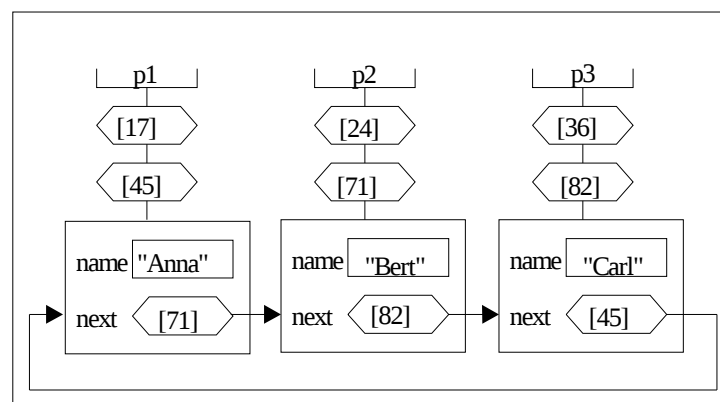
```
1 // -----
2 static boolean erfuehltDrei(int n) {
3     // Liefert true, wenn n die folgenden drei Bedingungen erfuehlt:
4     // 1. n ist groesser als 0
5     // 2. n ist ungerade
6     // 3. n ist durch 5 teilbar
7     // Sonst wird false als Ergebnis geliefert.
8     //
9     // Beispiele:
10    // erfuehltDrei( 15) liefert true (n erfuehlt alle 3 Bedingungen)
11    // erfuehltDrei(-15) liefert false (n ist nicht groesser als 0)
12    // erfuehltDrei( 10) liefert false (n ist nicht ungerade)
13    // erfuehltDrei( 11) liefert false (n ist nicht durch 5 teilbar)
14    ...
15    return istGroesserAls0 && istUngerade && istDurch5Teilbar;
16 } // erfuehltDrei
17 // -----
```

Hinweis: Am Anfang des Semesters haben Sie eine Aufgabe gelöst, in der es um **Dreiseite** und ihre Eigenschaften (**rechtwinklig**, **gleichseitig**, **gleichschenkelig** etc.) ging. Lösen Sie die vorliegende Aufgabe 5 "im gleichen Stil wie die Dreiseit-Hausaufgabe".

Formular mit den **Lösungen** zu den Aufgaben 1 bis 3:

1.1. Anzahl und Namen der Instanzelemente der Klasse Perle? 3 Instanzelemente: name, next, setNext() (4 Pkte)	1.2. Anzahl und Namen der Klassenelemente der Klasse TstPerle? 4 Klassenelemente: p1, p2, p3, main() (4 Pkte)
1.3. Anzahl und Namen der Module? 5 Module: TstPerle, Perle, p1, p2, p3 (4 Pkte)	
1.4. Die vollen Namen aller Elemente in allen Modulen? TstPerle.p1, p1.name, p1.next, p1.setNext(), TstPerle.p2, p2.name, p2.next, p2.setNext(), TstPerle.p3, p3.name, p3.next, p3.setNext(), TstPerle.main(). (6 Pkte)	
1.5. Ausgabe des Programms TstPerle zum Bildschirm? TstPerle: Jetzt geht es los! Bert folgt auf Anna Carl folgt auf Bert Anna folgt auf Carl TstPerle: Das war's erstmal! (6 Pkte)	
1.6. Zeichnen Sie die Bojen der Variablen p1, p2 und p3 bitte auf ein extra-Blatt! (6 Pkte)	
2. Die Nummern der falschen Zeilen in Aufgabe2: 6, 7, 11, 12.	
3.1. Die Ausgabe der Prozedur a31: Summe: 9	3.2. Die Ausgabe der Prozedur a32: erg: 5
3.3. Die Ausgabe der Prozedur a33: ir: 10, 6, 3, 7, 18,	3.4. Die Ausgabe der Prozedur a34: zahl: 4

Lösung 1.6.: Die Variablen p1, p2 und p3 als Bojen dargestellt:



Lösung 4:

```
1 // -----
2 static int wieOft(String s) {
3     // Wie oft steht in s eine Ziffer ('0' bis '9') unmittelbar nach
4     // einem kleinen Buchstaben ('a' bis 'z')? Diese Funktion liefert
5     // die Antwort. Falls s eine Null-Referenz ist oder auf einen
6     // leeren String referiert, wird 0 als Ergebnis geliefert.
7
8     if (s == null) return 0;
9
10    int anzahl = 0;
11    char c1, c2;
12
13    for (int i=0; i<s.length()-1; i++) {
14        c1 = s.charAt(i);
15        c2 = s.charAt(i+1);
16        if ('a' <= c1 && c1 <= 'z' &&
17            '0' <= c2 && c2 <= '9') {
18            anzahl++;
19        }
20    } // for
21    return anzahl;
22 } // wieOft
23 // -----
```

Lösung 5:

```
1 // -----
2 static boolean erfuelltDrei(int n) {
3     // Liefert true, wenn n die folgenden drei Bedingungen erfuehlt:
4     // 1. n ist groesser als 0
5     // 2. n ist ungerade
6     // 3. n ist durch 5 teilbar
7     // Sonst wird false als Ergebnis geliefert.
8
9     boolean istGroesserAls0 = n > 0;
10    boolean istUngerade = n % 2 != 0;
11    boolean istDurch5Teilbar = n % 5 == 0;
12
13    return istGroesserAls0 && istUngerade && istDurch5Teilbar;
14 } // erfuehltDrei
15 // -----
```