

**KLAUSUR SS 08**  
**Programmieren 1 - 90 mn**

Name:..... Vorname:.....

Nr:..... Punkte: **93** **93** /100 Note: **1,3** **1,3**

**Aufgabe 1 (10 Punkte).** Zeigen Sie die Ausgabe zur Konsole, wenn die folgende main Methode ausgeführt wird.

```
static public void main(String[] sonja) {  
    int n1 = 3;  
    int n2 = 3;  
    StringBuilder s1 = new StringBuilder("Maus");  
    StringBuilder s2 = new StringBuilder("Maus");  
    System.out.println(n1==n2);  
    System.out.println(s1==s2);  
    n2 += 3;  
    s2.setCharAt(2, 'i');  
    System.out.println(n1==n2);  
    System.out.println(s1==s2);  
} // main
```

10

Konsole

```
true  
false  
false  
false ✓
```

**Aufgabe 2 (15 Punkte).**

2.1 Zeigen Sie die Veränderungen im Speicher und die Ausgabe zur Konsole, wenn die folgenden Schleifen ausgeführt werden. Welchen Wert haben *i* und *j* jeweils nach vollständiger Ausführung der Anweisungen?

```
int i, j;
for (i=5; i>0; i--) {
    for(j=0; j<i; j++){
        System.out.print('*');
    }
    System.out.println(i);
}
```

Schritt	i	j
1	5	1
2	5	0
3	5	1
4	5	2
5	5	3
6	5	4
7	5	5
8	4	5
9	4	0
10	4	1
...		

Konsole:

x x x x x 5  
 x x x x 4  
 x x x 3  
 x x 2  
 x 1



Die letzten Werte die *i* und *j* haben sind:

*i* = 0    *j* = 1



2.2 Zeigen Sie die Ausgabe zur Konsole, wenn die folgenden Schleifen ausgeführt werden.

```
for (int i=2; i>0; i--) {
    System.out.print('X');
    for(int j=0; j<3; j++){
        System.out.print(j+"X");
    }
    System.out.println();
    System.out.print(i);
    for(int j=1; j<4; j++){
        System.out.print("Y"+i);
    }
    System.out.println();
}
```

Konsole:

XOX1X2X  
 2Y1Y2Y3  
 XOX1X2X  
 1Y1Y1Y1



**Aufgabe 3 (25 Punkte).** Betrachten Sie die folgende Klasse

```
class PunkteKonto {
    static protected long anzahlKontos = 0;
    public static long getAnzahlKontos() {
        return anzahlKontos;
    }
    private long punkte;
    public PunkteKonto() {
        punkte = 0;
        anzahlKontos++;
    }
    public PunkteKonto (long anfangsWert) {
        this();
        add(anfangsWert);
    }
    public void add(long n) {
        if (0 < n && n <= 10) punkte += n;
    }
    public String toString() {
        return "Punkte: " + punkte + "." ;
    }
}
```

**3.1** Schreiben Sie eine Klasse PunkteKontoA als Erweiterung der Klasse PunkteKonto.

1. Ein Objekt der Klasse PunkteKontoA soll ein zusätzliches Attribut namens kontoNummer (vom Typ String) enthalten.
2. Die Klasse PunkteKonto hat zwei Konstruktoren, einen ohne Parameter und einen mit einem long-Parameter. Ganz entsprechend soll die Klasse PunkteKontoA zwei Konstruktoren enthalten, einen mit einem String-Parameter und einen mit einem long-Parameter und einem String-Parameter.
3. Jedes PunkteKonto-Objekt enthält eine Methode namens toString, die die Anzahl der Punkte als String liefert (z.B. so: "Punkte: 8."). Jedes PunkteKonto-Objekt soll eine entsprechende Methode namens toString enthalten, die Strings wie z.B. "Punkte: 8, Kontonummer: 78795414." liefert.

✓ public class PunkteKontoA extends PunkteKonto {

✓ ~~protected~~ protected String kontoNummer;

✓ public PunkteKontoA (String nummer) {  
 super();  
 this.kontoNummer = nummer;

✓ }  
public PunkteKontoA (long anfangsWert, String nummer) {  
 this (nummer);  
 add (anfangsWert);  
}

```
public String toString() {
```

```
    StringBuilder s = new StringBuilder(super.toString());
```

```
    s.insert(s.length()-2, ", Kontonummer: " + this.KontoNummer);
```

```
    return s.toString();
```

```
}
```

### 3.2 Betrachten Sie das folgende Programmstück:

```
3. System.out.println(PunkteKontoA.getAnzahlKontos());
4. PunkteKonto pk1 = new PunkteKonto ();
5. PunkteKontoA pk2 = new PunkteKontoA ("455567");
6. PunkteKontoA pk3 = new PunkteKontoA (8, "466787");
7. PunkteKonto[] pkA = {pk1, pk2, pk3};
8. for (PunkteKonto pk : pkA ) {
9.     System.out.println(pk);
10. }
```

1 5

Angenommen, das Programmstück ist bis Zeile 4 (einschließlich) ausgeführt worden.

**Frage 1:** Wieviele Objekte existieren in diesem Moment?

1

**Frage 2:** Welchen Wert hat die Variable PunkteKontoA.anzahlKontos in diesem Moment?

1

**Frage 3:** Wieviele Variablen des Typs long existieren in diesem Moment? Wie heißen diese Variablen mit vollem Namen? (Bin mir nicht sicher ob das gemeint ist)

Es gibt nur ein Klassenattribut mit Namen: PunkteKontoA.anzahlKontos  
und ein <sup>privates</sup> Objektattribut mit Namen: pk1.punkte. ✓

Angenommen, das Programmstück ist bis Zeile 7 (einschließlich) ausgeführt worden.

**Frage 4:** Wieviele Objekte existieren in diesem Moment?

3 (+ 1 Array) ✓

**Frage 5:** Welchen Wert hat die Variable PunkteKontoA.anzahlKontos in diesem Moment?

3 ✓

**Frage 6:** Wieviele Variablen des Typs long existieren in diesem Moment? Wie heißen diese Variablen mit vollem Namen?

4 ✓ siehe Frage 3, nur diesmal auch für pk2 und pk3, aber auch wieder von außen nicht zugreifbar (privat).

Was gibt dieses Programmstück zum Bildschirm aus?

Punkte: 0. ✓

Punkte: 0, Kontonummer: 455567. ✓

Punkte: 8, Kontonummer: 466787. ✓

**Aufgabe 4 (15 Punkte)** Schreiben Sie den Rumpf der folgenden Methode, damit sie leistet, was im Kommentar beschrieben ist.

15

```
public static void diagonale(String s){  
    // Zeigt am Schirm die Zeichen der Zeichenkette s  
    // in einer Diagonale.  
    // Zum Beispiel, zeigt der Aufruf diagonale("abcde")  
    // am Schirm das Folgende:  
    // a  
    //  b  
    //   c  
    //    d  
    //     e
```

```
for for(int i=0; i < s.length(); i++){  
    for(int j=0; j < i; j++){  
        System.out.print(" ");  
    }  
    System.out.println(s.charAt(i));
```

}

}

✓

Aufgabe 5 (20 Punkte)

5.1 Erklären Sie mit eigenen Beispielen die 3 Hauptmerkmale einer Klassen-Methode.

Kann verschiedene Sichtbarkeiten haben: public, private, protected  
Enthält vor der Deklaration, bzw. in der Deklaration das Wort static. ✓  
~~Kann~~ Sollte nur statisch aufgerufen werden. ✓

```
class A {
    static void method();
}
```

```
A a = new A();
```

A.method(); // korrekt ✓  
B.method(); // sollte nicht gemacht werden

Auf welche Elemente kann sie greifen?

5.2 Erklären Sie mit eigenen Beispielen die 3 Hauptmerkmale einer Objekt-Methode.

Kann verschiedene Sichtbarkeiten haben: public, private, protected  
Ist auf keinen Fall statisch. ✓

```
class B {
    public void foo();
}
```

```
B a = new B();
```

a.foo(); // korrekt ✓  
B.foo(); // total falsch ✓

**Aufgabe 6 (15 Punkte)** Schreiben Sie den Rumpf der folgenden Methoden, damit sie leisten, was im Kommentar beschrieben ist.

**6.1**

```

public static boolean hatNurPositiveZahlen(int[] ir){
// Liefert true genau dann wenn ir nur positive Zahlen enthaelt.
//
// Zum Beispiel mit:
// int[] ir1 = {1, 2, 3}, int[] ir2 = {}, int[] ir3 = {-1, 2} gilt
// hatNurPositiveZahlen(ir1) ist gleich true.
// hatNurPositiveZahlen(ir2) ist gleich true.
// hatNurPositiveZahlen(ir3) ist gleich false.

```

```

for (int i=0; i < ir.length; i++)
    if (ir[i] < 0) return false;

```

```

return true;

```

}

**6.2**

```

public static int[] anzahlZiffer(int[] ir){
// Verlaesst sich darauf, dass ir nur Zahlen zwischen 0 und 9
// enthaelt.
// Liefert ein Array der Laenge 10 zurueck, die enthaelt wie oft
// jede Zahl in ir vorkommt.
//
// Zum Beispiel mit:
// int[] ir1 = {1, 2, 3}, int[] ir2 = {}, int[] ir3 = {1, 1} gilt
// anzahlZiffer(ir1) ist gleich [0, 1, 1, 1, 0, 0, 0, 0, 0, 0].
// anzahlZiffer(ir2) ist gleich [0, 0, 0, 0, 0, 0, 0, 0, 0, 0].
// anzahlZiffer(ir3) ist gleich [0, 2, 0, 0, 0, 0, 0, 0, 0, 0].

```

```

for (int i=0; i < ir.length; i++)

```

nicht Java!

```

int[] result = new int Array(0,0,0,0,0,0,0,0,0,0);

```

// Ich weiß nicht ob man in Java ein Array so initialisiert  
// Kann, auf jeden fall muss  
// überall eine 0 drin sein.

```

for (int i=0; i < ir.length; i++) {
    if (ir[i] > 9 || ir[i] < 0) continue;
    result[ir[i]]++;
}

```

```

return result;

```

}