

# Test 1 in Programmkonstruktion

59 / 100 Punkte

## 1. Multiple-Choice-Aufgaben

20 / 30 Punkte

Bitte wählen Sie *alle* zutreffenden Antwortmöglichkeiten aus. Es können beliebig viele Antwortmöglichkeiten zutreffen, auch alle oder keine.

### Aufgabe 1.1.

3 / 5 Punkte

Wählen Sie jene Variablendeklarationen mit Initialisierungen aus, die in Java hinsichtlich der Typen korrekt sind:

- ☐ `short n = 'a' - 7;`
- ☐ `double d = 'f' * '0';`
- ☐ `long l = 900 + 0.9;`
- ☒ `String xs = '0' + "000A";`
- ☐ `float f = 1.0 / 8.8;`

### Aufgabe 1.2.

4 / 5 Punkte

Angenommen, `a` und `b` sind initialisierte `boolean`-Variablen. Wählen Sie jene Anweisungen aus, die in Java dieselben Seiteneffekte haben wie `System.out.print(a &= b);`

- ☒ `System.out.print(a = a & b);`
- ☐ `a = a && b;`
- ☐ `System.out.print(a = a && b);`
- ☐ `System.out.print(a == a && b);`
- ☐ `a &= b;`

### Aufgabe 1.3.

1 / 5 Punkte

Angenommen, `x` und `y` sind initialisierte `int`-Variablen. Wählen Sie jene Ausdrücke aus, die in Java zu `++x * y` äquivalent sind:

- ☒ `x++ * y + x`
- ☐ `x++ * y + y`
- ☒ `(x + 1) * y`
- ☐ `(x++ + 1) * y`
- ☒ `y * (x += 1)`

### Aufgabe 1.4.

12 / 15 Punkte

Welche Methoden geben als Ergebnis die Summe aller *nicht* durch 3 teilbaren Zahlen



```
public static int sumDivDo(int x) {  
    int sum = 0;  
    do {  
        if (x % 3 != 0) {  
            sum += x;  
        }  
    } while (--x > 0);  
    return sum;  
}
```



```
public static int sumDivRet(int x) {  
    int sum = 0;  
    while (x > 0) {  
        if (x % 3 != 0) {  
            return sum + x;  
        }  
        x--;  
    }  
    return sum;  
}
```



```
public static int sumDivWhile(int x) {  
    int sum = 0;  
    while (x < 0) {  
        if (x % 3 != 0) {  
            sum += x;  
        }  
        x -= 1;  
    }  
    return sum;  
}
```



```
public static int sumDivForRev(int x) {  
    int sum = 0;  
    for (; x > 0; --x) {  
        sum += x % 3 != 0 ? x : 0;  
    }  
    return sum;  
}
```



```
public static int sumDivFor(int x) {  
    int sum = 0;  
    for (int i = 0; i++ < x; ) {  
        sum += i % 3 == 0 ? 0 : i;  
    }  
    return sum;  
}
```

## 2. Multiple-Choice-Aufgaben zur Ergänzung von Methoden

10 / 20 Punkte

Bitte wählen Sie *alle* `return`-Anweisungen aus, die dazu führen, dass die davor stehenden Methoden sich so verhalten wie in den Kommentaren beschrieben. Es können beliebig viele Antwortmöglichkeiten zutreffen, auch alle oder keine.

### Aufgabe 2.1.

4 / 5 Punkte

```
// returns the smallest number in the parameters x, y, and z;  
// x, y and z may be equal.  
public static int smallest(final int x, final int y, final int z) {  
    // TODO: Welche return-Anweisung kann hier stehen?  
}
```

☒ `return x >= y && y <= z ? y : (z >= x ? x : z);`

☐ `return x <= z || y <= z ? (x <= y ? x : y) : z;`

☒ `return x < y && x < z ? x : (y < z ? y : z);`

☐ `return x <= y || y <= z ? (y <= z ? y : z) : x;`

☒ `return x > y && y < z ? y : (x > z ? z : x);`

### Aufgabe 2.2.

3 / 5 Punkte

```
// returns true if (and only if) x is in the range between y and z  
// (including y and z); z >= y is assumed.  
public static boolean inRange(final int x, final int y, final int z) {  
    // TODO: Welche return-Anweisung kann hier stehen?  
}
```

☐ `return x - y >= 0 && z - x >= 0;`

☒ `return y <= x && x <= z;`

☐ `return y <= x <= z;`

☐ `return !(x < y || x > z);`

☐ `return z - (x - y) >= 0;`

### Aufgabe 2.3.

1 / 5 Punkte

```
// returns a string containing the characters in c and d
// ordered according to the corresponding int values;
// e.g., ord('a', 'b') gives "ab", and ord('b', 'a') gives "ab".
public static String ord(char c, char d) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

☒ `return (c == 'a' && d == 'b') || (c == 'b' && d == 'a') ? "ab" : "";`

☐ `return "" + (c > d ? d + c : c + d);`

☐ `return c > d ? "" + d + c : "" + c + d;`

☐ `return (c > d ? "" + d : "" + c) + (c > d ? "" + c : "" + d);`

☐ `return "" + (c > d ? d : c) + (c > d ? c : d);`

### Aufgabe 2.4.

2 / 5 Punkte

```
// returns a string containing a grammar (Grammatik) in EBNF form;
// the grammar accepts expressions that can be used as literals in Java;
// terminal symbols (Terminalsymbole) are written in single quotes;
// examples: "'true' | 'false'", "'true'", "'false'".
public static String ebnf() {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

☒ `return "('0' | '1') '+' ('2' | '3');`

☒ `return "('0' | '1' | '2')" + " [ 'L' ]";`

☐ `return "('a' | 'b') { 'a' | 'b' | '0' | '1' }";`

☐ `return "'\'' { 'a' | 'b' | 'c' } '\''";`

☐ `return "'0' '.' ('0' | '1') ['e' '2']";`

## 3. Auswahlaufgaben

14 / 30 Punkte

Jede dieser Aufgaben hat genau eine zutreffende Antwortmöglichkeit. Bitte wählen Sie diese aus.

### Aufgabe 3.1.

3 / 3 Punkte

Der Ausdruck `0x00F0 & '0'` liefert in Java ein Ergebnis vom Typ:

☒ `int` ☐ `long` ☐ `char` ☐ `boolean` ☐ `String`

### Aufgabe 3.2.

3 / 3 Punkte

Angenommen, `b` ist eine uninitialisierte lokale Variable vom Typ `boolean`. Wählen Sie die erlaubte Verwendung von `b`:

- ☐ `System.out.print(b && 0 == 0);`
- ☐ `System.out.print(b ? 8 : 0);`
- ☒ `b = false;`
- ☐ `while(b) { System.out.print('0'); b = !b; }`
- ☐ `b = 8 | 0;`

### Aufgabe 3.3.

3 / 3 Punkte

Die Auswertung von `(1 + 2) + "3"` liefert in Java folgendes Ergebnis:

- ☐ `"24"`
- ☐ `"231"`
- ☒ `"33"`
- ☐ `"123"`
- ☐ `"6"`

### Aufgabe 3.4.

0 / 3 Punkte

`for (int i = 0; i < 10; i++) { f(i); }` ist in Java äquivalent zu:

- ☐ `int i; for (i = 0; i++ < 10; ) { f(i); }`
- ☒ `for (int i = 1; i <= 10; i++) { f(i - 1); }`
- ☐ `int i = 0; while (i < 10) { i++; f(i); }`
- ☐ `for (int i = 1; i <= 10; ++i) { f(i); };`
- ☐ `int i; do { i = 0; f(i); } while (++i < 10);`

### Aufgabe 3.5.

0 / 5 Punkte

```
public static void whileLoop(int v) {  
    int sum = 0;  
    while (v-- > 2) {  
        sum += v;  
    }  
    System.out.println(sum);  
}
```

Welche Zahl wird durch einen Aufruf von `whileLoop(5)` ausgegeben?

- ☒ weniger als 9
- ☐ 9
- ☐ 10
- ☐ 11
- ☐ mehr als 11

### Aufgabe 3.6.

5 / 5 Punkte

```
public static void forLoop() {  
    for (int i = 4; i < 44; i += 4) {  
        System.out.println(i);  
    }  
}
```

Wie viele Zeilen werden bei einem Aufruf von `forLoop()` ausgegeben?

- ☐ weniger als 9    ☐ 9    ☒ 10    ☐ 11    ☐ mehr als 11

### Aufgabe 3.7.

0 / 3 Punkte

Die Auswertung von `(long) ('a' + 1.6)` liefert in Java folgendes Ergebnis (wobei 97 der ASCII-Wert von 'a' ist):

- ☒ 98    ☐ 'b'    ☒ 98.6    ☐ 'c'    ☐ 99

### Aufgabe 3.8.

0 / 5 Punkte

```
public static void seq(int x) {  
    int sum = 7 % x;  
    sum += 7 / x;  
    sum += 2 << x;  
    System.out.println(sum);  
}
```

Welche Zahl wird von `seq(2)` ausgegeben?

- ☐ weniger als 9    ☒ 9    ☐ 10    ☐ 11    ☐ mehr als 11

## 4. Auswahlaufgaben zur Ergänzung von Methoden

15 / 20 Punkte

In den Methoden sind die Buchstaben A, B, C, D und E jeweils durch Ausdrücke zu ersetzen. Bitte wählen Sie für jeden dieser Buchstaben genau eine zutreffende Antwortmöglichkeit. Die Methoden müssen sich so verhalten, wie in den Kommentaren angegeben. Punkte gibt es nur, wenn die gewählten Antwortmöglichkeiten zusammenpassen.

### Aufgabe 4.1.

5 / 5 Punkte

```
// returns "0" if x == 0, "..0" if x < 0, and "0.." if x > 0.
public static String asString(int x) {
    String result = A;
    if (B) {
        result = result + C;
    }
    if (D) {
        result = E + result;
    }
    return result;
}
```

A:

- ☐ "0.." 
 ☐ "..0" 
 ☒ "0" 
 ☐ "" 
 ☐ ".."

B:

- ☐ x <= 0 
 ☒ x > 0 
 ☐ x < 0 
 ☐ x >= 0 
 ☐ x != 0

C:

- ☐ "0.." 
 ☐ "..0" 
 ☐ "0" 
 ☐ "" 
 ☒ ".."

D:

- ☐ x <= 0 
 ☐ x > 0 
 ☒ x < 0 
 ☐ x >= 0 
 ☐ x != 0

E:

- ☐ "0.." 
 ☐ "..0" 
 ☐ "0" 
 ☐ "" 
 ☒ ".."

### Aufgabe 4.2.

0 / 5 Punkte

```
// returns the sum of all numbers not divisible by m (nicht durch m teilbar)
// in the range from 1 to n (including 1 and n);
// m >= 1 and n >= 1 is assumed.
public static long divSum(long m, long n) {
    long sum = 0;
    while (A) {
        sum += B ? C : D;
        n--;
    }
    return sum;
}
```

A:

- ☒ n >= 1 
 ☐ n <= 0 
 ☐ m <= 0 
 ☐ m >= 1 
 ☐ m != n

B:

- ☐ `n % 2 == 0`    ☐ `m != n`    ☐ `m % 2 == 0`    ☒ `n % m == 0`
- ☐ `m % n == 0`

C:

- ☒ `0`    ☐ `1`    ☐ `2`    ☐ `m`    ☒ `n`

D:

- ☐ `0`    ☐ `1`    ☐ `2`    ☒ `m`    ☒ `n`

#### Aufgabe 4.3.

5 / 5 Punkte

```
// returns true if the value of x contains the digit '7'
// where x is in decimal notation (Dezimaldarstellung),
// returns false otherwise; x >= 0 is assumed.
public static boolean seven(int x) {
    while (A) {
        if (B) {
            return true;
        }
        x = C;
    }
    return false;
}
```

A:

- ☐ `x < 0`    ☐ `x == 0`    ☐ `x <= 0`    ☒ `x > 0`    ☐ `x >= 0`

B:

- ☐ `x % 10 != 7`    ☐ `x % 7 == 10`    ☒ `x % 10 == 7`    ☐ `x == 0`
- ☐ `x % 7 != 10`

C:

- ☐ `x / 7`    ☐ `x * 7`    ☐ `x * x`    ☐ `x * 10`    ☒ `x / 10`

#### Aufgabe 4.4.

5 / 5 Punkte

```
// prints the squares (Quadratzahlen) of all numbers
// in the range from -x to x (including -x and x).
public static void squares(int x) {
    for (int i = A; B; C) {
        System.out.println(D);
    }
}
```



A:

- ☐ 0    ☐ -1    ☐ x    ☒ -x    ☐ 1

B:

- ☐ i == x    ☐ x < i    ☐ i < x    ☒ i <= x    ☐ x <= i

C:

- ☐ x \*= 2    ☐ i--    ☐ x--    ☐ x++    ☒ i++

D:

- ☐ x \* 2    ☐ i \* x    ☐ x \* x    ☒ i \* i    ☐ i \* 2