

1. Multiple-Choice-Aufgaben

23 / 30 Punkte

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

Aufgabe 1.1.

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);`

☒ `a ^= b;`☐ `System.out.print(a ^ b);`☒ `System.out.print(a = (a && !b) || (!a && b));`☒ `System.out.print(a = a ^ b);`☐ `a ^ b;`

Aufgabe 1.2.

3 / 5 Punkte

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

☒ `String xs = "D" + 0;`☒ `float f = 8.0f * 8.9f;`☐ `long l = 'D';`☐ `float f = '0' + 0;`☒ `char c = (char)('0' + 9.0);`

Aufgabe 1.3.

4 / 5 Punkte

Angenommen, `x` und `y` sind initialisierte `long`-Variablen. Wählen Sie jene Ausdrücke aus, die in Java zu `x *= y + 1` äquivalent sind:

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

Aufgabe 1.4.

12 / 15 Punkte

Welche Methoden geben als Ergebnis das Produkt

$(1/2) * (2/3) * \dots * (x/(x+1))$ zurück, wobei der Parameter x eine Zahl größer 0 ist?



```
public static double prodDivFor(int x) {
    double prod = 0.5;
    for (int i = 2; i <= x; i++) {
        prod *= ((double)i) / (i + 1);
    }
    return prod;
}
```



```
public static double prodDivDo(int x) {
    double prod = 1.0;
    do {
        prod *= x / (double)(x + 1);
    } while (--x > 0);
    return prod;
}
```



```
public static double prodDivRet(int x) {
    while (x > 0) {
        return ((double)x) / (double)(x + 1);
        x--;
    }
    return 0.5;
}
```



```
public static double prodDivWhile(int x) {
    double prod = 1.0;
    while (x > 0) {
        prod *= ((double)x) / (double)(x + 1);
        x--;
    }
    return prod;
}
```



```
public static double prodDivForRev(int x) {
    double prod = 1.0;
    for (; x > 0; x--) {
        prod = prod * x / (x + 1);
    }
    return prod;
}
```

2. Multiple-Choice-Aufgaben zur Ergänzung von Methoden

14 / 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 true if (and only if) both x and y (sowohl x als auch y)
// are in the range between -z and z (including -z and z).
public static boolean inRange(final int x, final int y, final int z) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

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

☒ `return x * x <= z * z && y * y <= z * z;`

☐ `return x * y <= z * z;`

☒ `return -z <= x && x <= z && -z <= y && y <= z;`

☐ `return -z <= x <= z && -z <= y <= z;`

Aufgabe 2.2.

5 / 5 Punkte

```
// result contains the characters in c and d if d is a digit (from '0' to '9');
// otherwise the result contains just the character in c;
// e.g., initial('a', '3') gives "a3", and initial('b', 'c') gives "b".
public static String initial(char c, char d) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

☐ `return (d < '0' || d > '9' ? "" + d : "") + c;`

☐ `return "" + c + (d < '0' || d > '9' ? '' : d);`

☒ `return d >= '0' && d <= '9' ? "" + c + d : "" + c;`

☐ `return c + (d >= '0' && d <= '9' ? d : "");`

☒ `return c + (d < '0' || d > '9' ? "" : "" + d);`

Aufgabe 2.3.

2 / 5 Punkte

```
// returns true if and only if all parameters have the same value.  
public static boolean allEqual(boolean a, boolean b, boolean c) {  
    // TODO: Welche return-Anweisung kann hier stehen?  
}
```

- ☒ `return !((a ^ b) || (b ^ c));`
- ☐ `return (a && b && c) || (!a && !b && !c);`
- ☐ `return (a ? b && c : !b && !c);`
- ☐ `return !(a || b || c) || (a && b && c);`
- ☒ `return (a & b & c) | (!a & !b & !c);`

Aufgabe 2.4.

3 / 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 "'\'' ('a' | '3') '\''";`
- ☐ `return " ('0' | '3') ('*' | '+') ('0' | '3')";`
- ☒ `return "'0' ['D' | 'F' | 'L']";`
- ☒ `return "'3' ['3'] ['.' '3' { '3' }]";`
- ☐ `return "'0' 'x' ('0' | 'a') { '0' | 'a' }";`

3. Auswahlaufgaben

24 / 30 Punkte

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

Aufgabe 3.1.

0 / 3 Punkte

`for (int i = n; i != 0; i /= 2) { f(i); }` ist in Java äquivalent zu:

- ☒ `{ int i = n; while (i != 0) { f(i); i = i / 2; } }`
- ☐ `{ int i; do { i = 1; f(i); } while ((i /= 2) != 0); }`
- ☒ `for (int i = 1; i <= n; i *= 2) { f(i); }`
- ☐ `{ int i; for (i = n; (i /= 2) != 0;) { f(i); } }`
- ☐ `for (int i = n; i > 0;) { f(i /= 2); }`

Aufgabe 3.2.

5 / 5 Punkte

```
public static void forLoop() {  
    for (int i = 66; i > 3; i -= 6) {  
        System.out.println(i);  
    }  
}
```

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

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

Aufgabe 3.3.

0 / 3 Punkte

Der Ausdruck `0xbb0E + 0000` liefert in Java ein Ergebnis vom Typ:

- ☒ `int`
- ☐ `boolean`
- ☐ `long`
- ☒ `char`
- ☐ `short`

Aufgabe 3.4.

5 / 5 Punkte

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

Welche Zahl wird durch einen Aufruf von `whileLoop(12)` ausgegeben?

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

Aufgabe 3.5.

3 / 3 Punkte

Die Auswertung von `(byte) ('a' + 1.9)` liefert in Java folgendes Ergebnis (wobei 97 der ASCII-Wert von 'a' ist):

- ☐ 98.9
- ☐ 'c'
- ☒ 98
- ☐ 'b'
- ☐ 99

Aufgabe 3.6.

5 / 5 Punkte

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

Welche Zahl wird von `seq(5)` ausgegeben?

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

Aufgabe 3.7.

3 / 3 Punkte

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

- ☐ "33" ☒ "123" ☐ "321" ☐ "15" ☐ "6"

Aufgabe 3.8.

3 / 3 Punkte

Angenommen, `b` ist ein als `final` deklarierter formaler Parameter vom Typ `boolean`.

Wählen Sie die erlaubte Verwendung von `b`:

- ☐ `b = 0 & 8;`
- ☐ `System.out.print(b = 0);`
- ☒ `System.out.print(b ? 3 : 6);`
- ☐ `b |= false;`
- ☐ `if(b) { System.out.print('0'); b = !b; }`

4. Auswahlaufgaben zur Ergänzung von Methoden

15 / 20 Punkte

In den Methoden sind die Buchstaben A, B, C und D 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 "x < 0" if x < 0,
//          "0"      if x == 0,
//          "0 < x"  if x > 0.
public static String asString(int x) {
    String result = "0";
    if (A) {
        result += B;
    }
    if (C) {
        result = D + result;
    }
    return result;
}
```

A:

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

B:

- ☐ "x < 0"
 ☐ "0"
 ☐ "0 < x"
 ☒ "< x"
 ☐ "x < "

C:

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

D:

- ☐ "x < 0"
 ☐ "0"
 ☐ "0 < x"
 ☐ "< x"
 ☒ "x < "

Aufgabe 4.2.

5 / 5 Punkte

```
// counts the integers divisible by m (Anzahl der durch m teilbaren ganzen Zahl
// in the range from 1 to n (including 1 and n);
// m >= 1 and n >= 1 is assumed.
public static int divCount(long m, long n) {
    int count = 0;
    while (A) {
        count += B ? C : D;
        n--;
    }
    return count;
}
```

A:

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

B:

- ☐ `m != n` ☐ `m % n == 1` ☐ `n % m == 1` ☐ `m % n == 0`
- ☒ `n % m == 0`

C:

- ☐ `2` ☐ `m` ☐ `n` ☐ `0` ☒ `1`

D:

- ☐ `2` ☐ `m` ☐ `n` ☒ `0` ☐ `1`

Aufgabe 4.3.

5 / 5 Punkte

```
// returns true if the value of x in decimal notation (Dezimaldarstellung)
// does NOT contain the digit '7', returns false otherwise;
// x >= 0 is assumed.
public static boolean noSeven(int x) {
    while (x > 0) {
        if (A) {
            return B;
        }
        x = C;
    }
    return D;
}
```

A:

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

B:

- ☒ `false` ☐ `0` ☐ `-1` ☐ `true` ☐ `1`

C:

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

D:

- ☐ `false` ☐ `0` ☐ `-1` ☒ `true` ☐ `1`

Aufgabe 4.4.

0 / 5 Punkte

```
// prints the numbers i / (i + 1) using floating point division (double)
// for all integers i in the range from 1 to x (including 1 and x).
public static void divD(int x) {
    for (int i = A; B; C) {
        System.out.println(D);
    }
}
```

A:

- ☐ 0 ☒ 1 ☐ x + 1 ☐ x - 1 ☐ x

B:

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

C:

- ☐ x-- ☐ x++ ☐ x *= 2 ☒ i++ ☐ i--

D:

- ☐ (double)(x / (x + 1)) ☐ (double)(i / x)
- ☐ i / (double)(i + 1) ☒ (double)(i / (i + 1))
- ☐ i / (double)(x + 1)