PK-Einstufungstest

76 / 100 Punkte

1. Allgemeine Multiple-Choice-Aufgaben

10 / 15 Punkte

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

Aufgabe 1.1.

4 / 5 Punkte

Wählen Sie jene Ausdrücke aus, die als Ergebnis den Typ `double` haben:

- [ ] `9 / 1`
- [x] `80000808080d`
- [x] `0.0D`
- [x] `7.9`
- [ ] `0.009e-9D`

Aufgabe 1.2.

4 / 5 Punkte

Angenommen, `a` und `b` sind initialisierte `boolean`-Variablen. Wählen Sie jene Java-Anweisungen aus, durch die mindestens einmal Ausgabe erzeugt wird:

- [x] `do { b = !b; System.out.print('+'); } while (b);`
- [ ] `for (; a; a = !a) { System.out.print('-'); }`
- [ ] `if (!a && b) { System.out.print('!'); } else if (a || !b) { System.out.print('?'); }`
- [x] `if (a || b) { System.out.print('!'); } else { System.out.print('?'); }`
- [x] `while (b || !b) { System.out.print('-'); }`

Aufgabe 1.3.

2 / 5 Punkte

Wählen Sie jene Ausdrücke aus, die in Java äquivalent sind zu `6L`:

- [ ] `'g' - 'a'`
- [ ] `3L << 1`
- [ ] `14L & 22L`
- [ ] `12L >> 2`
- [ ] `2 | 6L`

2. Allgemeine Auswahlaufgaben

9 / 15 Punkte

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

Aufgabe 2.1.

0 / 3 Punkte

Die Auswertung von `"" + 4 + 2` liefert in Java folgendes Ergebnis:

- [x] `"6"`
- [ ] `"42"`
- [ ] `6`
- [ ] `42`
- [ ] keines davon

Aufgabe 2.2.

3 / 3 Punkte

Der Ausdruck `'A' + 'B'` liefert in Java ein Ergebnis vom Typ:

- [ ] `long`
- [ ] `String`
- [x] `int`
- [ ] `char`
- [ ] keinem davon

Aufgabe 2.3.

3 / 3 Punkte

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

- [ ] `i = i++;`
- [ ] `i += 2;`
- [ ] `System.out.println(i * 6);`
- [ ] `while (i < 0) i++;`
- [x] `int j = (i = 9);`

Aufgabe 2.4.

3 / 3 Punkte

Die Auswertung von `'A' + 1f` liefert in Java folgendes Ergebnis:

- [ ] `66`
- [x] `66.0`
- [ ] `'f'`
- [ ] `"A1f"`
- [ ] keines davon

Aufgabe 2.5.

0 / 3 Punkte

`if (x == y) { x = y - 1; }` ist in Java äquivalent zu:

- [ ] `while ((x = y - 1) == (y - 1));`
- [ ] `for (; x == y; y--);`
- [ ] `if (x == y) x = --y;`
- [x] `switch (x) { case y: x = y - 1; }`
- [ ] `while (x == y) x--;`

3. Multiple-Choice-Aufgaben zu Ausdrücken und Bedingungen

15 / 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 3.1.

4 / 5 Punkte

```
// returns the initials constructed from first name f and last name l;
// e.g., initials("Grace", "Hopper") gives "G.H.";
// f and l contain at least one character (not null)
public static String initials(String f, String l) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return f[0] + "." + l[0] + ".";`
- [ ] `return f.charAt(0) + "." + l.charAt(0) + ".";`
- [ ] `return unzip((f, l).charAt(0) + ".");`
- [ ] `return (f + l)[0][0] + "." + (f + l)[0][1] + ".";`
- [ ] `return f[0..1] + "." + l[0..1] + ".";`

Aufgabe 3.2.

4 / 5 Punkte

```
// returns true if (and only if) x is in the range between -y and y (including -y and y
// y > 0
public static boolean fromIntsToBool(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return ((x + y) >= 0) && ((x - y) <= 0);`
- [ ] `return -y <= x <= y;`
- [ ] `return ((y - x) + (x - y)) > 0;`
- [x] `return (-y <= x) && (x <= y);`
- [x] `return !((y < x) || (x < -y));`

Aufgabe 3.3.

4 / 5 Punkte

```
// returns the non-negative difference between x and y
public static int absolute(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return (x - y);`
- [x] `return Math.abs(x - y);`
- [x] `return (x < y) ? (y - x) : (x - y);`
- [ ] `return Math.max(x - y, y - x);`
- [x] `return (x - y) * (((x - y) > 0) ? 1 : -1);`

Aufgabe 3.4.

3 / 5 Punkte

```
// returns 0 if a and b are false,
// returns 1 if a is true and b is false,
// returns 2 if a is false and b is true,
// returns 3 if a and b are true
public static int fromBoolsToInt(boolean a, boolean b) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return (a ? 1 : 0) | (b ? 2 : 0);`
- [x] `return (b ? 3 : 1) - (a ? 0 : 1);`
- [x] `return (a ? 1 : 0) + (b ? 2 : 0);`
- [ ] `return (a ? 3 : 2) & (b ? 3 : 1);`
- [ ] `return (a ? 1 : 2) ^ (b ? 2 : 1);`

4. Auswahlaufgaben zu Programmverzweigungen

20 / 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 the largest index i where i < high and a[i] equals c;
// returns -1 if there is no such index;
// high <= a.length always holds
public static int index(final int high, final char c, final char[] a) {
    if (A) {
        return -1;
    } else if (B) {
        return C;
    }
    return index(D, c, a);
}
```

A:

- ○ `high >= 0`
- ○ `high > 0`
- ● `high <= 0`
- ○ `high > 1`
- ○ `high < 0`

B:

- ○ `c == a[high]`
- ○ `c == high[a - 1]`
- ○ `c == a`
- ● `c == a[high - 1]`
- ○ `c == high[a]`

C:

- ○ `1`
- ○ `high + 1`
- ● `high - 1`
- ○ `high`
- ○ `-1`

D:

- ○ `1`
- ○ `high + 1`
- ● `high - 1`
- ○ `high`
- ○ `-1`

Aufgabe 4.2.

5 / 5 Punkte

```
// returns "pos" if n is greater than 0, returns "neg" otherwise
public static String posNeg(int n) {
    switch (A) {
        case 1:
            return B;
    }
    return C;
}
```

A:

- ○ `(n >= 0) ? n : 1`
- ○ `n <= 0`
- ○ `n`
- ● `(n < 0) ? 1 : 0`
- ○ `n < 0`

B:

- ● `"neg"`
- ○ `"pos"`

C:

- ○ `"neg"`
- ● `"pos"`

Aufgabe 4.3.

5 / 5 Punkte

```
// returns "a_b" if left equals "a" and right equals "b",
// returns "a_" if left equals "a" and right differs from "b",
// returns "_b" if left differs from "a" and right equals "b",
// returns "_" if left differs from "a" and right differs from "b"
public static String combString(String left, String right) {
    String result = "_";
    if (A) {
        result += B;
    }
    if (C) {
        result = D + result;
    }
    return result;
}
```

A:

- [x] `right.equals("b")`
- [ ] `left.equals("a") && right.equals("b")`
- [ ] `!right.equals("b")`
- [ ] `left.equals("a")`
- [ ] `left.equals("a") || right.equals("b")`
- [ ] `!left.equals("a")`

B:

- [x] `"b"`
- [ ] `"_"`
- [ ] `"a_b"`
- [ ] `"_b"`
- [ ] `"a_"`
- [ ] `"a"`

C:

- [ ] `right.equals("b")`
- [ ] `left.equals("a") && right.equals("b")`
- [ ] `!right.equals("b")`
- [x] `left.equals("a")`
- [ ] `left.equals("a") || right.equals("b")`
- [ ] `!left.equals("a")`

D:

- [ ] `"b"`
- [ ] `"_"`
- [ ] `"a_b"`
- [ ] `"_b"`
- [ ] `"a_"`
- [x] `"a"`

Aufgabe 4.4.

5 / 5 Punkte

```
// returns the product of all integers from 1 to n (this is 1 * 2 * ... * n) if n > 0;
// returns 1 otherwise
public static long product(final long n) {
    if (A) {
        return B;
    }
    return product(C) * n;
}
```

A:

- ○ `n != 1`
- ○ `n == 1`
- ○ `n >= 1`
- ● `n <= 1`
- ○ `n > 1`

B:

- ○ `n`
- ● `1L`
- ○ `n + n`
- ○ `n - 1`
- ○ `n + 1`

C:

- ○ `n`
- ○ `1L`
- ○ `n + n`
- ● `n - 1`
- ○ `n + 1`

5. Auswahlaufgaben zu Schleifen

10 / 15 Punkte

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

Aufgabe 5.1.

5 / 5 Punkte

```
public static void forLoop() {
    for (int i = 0; i <= 500; i += 10) {
        System.out.println(i);
    }
}
```

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

- ○ weniger als 49
- ○ 49
- ○ 50
- ● 51
- ○ mehr als 51

Aufgabe 5.2.

0 / 5 Punkte

```
public static void whileLoop(int v) {
    int sum = 0;
    while (v > 1) {
        sum += v--;
    }
    System.out.println(sum);
}
```

Welche Zahl wird durch einen Aufruf von `whileLoop(4)` ausgegeben?

- ○ 7
- ○ 9
- ● 10
- ○ 16
- ○ mehr als 16

Aufgabe 5.3.

5 / 5 Punkte

```
public static void forEachLoop() {
    int min = 1000;
    int[] is = { 9 + 8, 30 / 2, 847 % 13, 4 * 6 };
    for (int i : is) {
        if (i < min) {
            min = i;
        }
    }
    System.out.println(min);
}
```

Welche Zahl wird von `forEachLoop()` ausgegeben?

- ● weniger als 13
- ○ 13
- ○ 14
- ○ 15
- ○ mehr als 15

6. Multiple-Choice-Aufgabe zu Schleifen

12 / 15 Punkte

Bitte wählen Sie ALLE Java-Methoden aus, die das beschriebene Verhalten haben. Beliebig viele Methoden können dieses Verhalten haben, auch alle oder keine.

Aufgabe 6.1.

12 / 15 Punkte

Welche Methoden geben als Ergebnis die Summe aller quadrierten Zahlen von 2 bis 9 (also `2 * 2 + 3 * 3 + ... + 9 * 9`) zurück?

- ☑

```
public static long quadSumArray() {
    long[] quadSums = new long[8];
    quadSums[7] = 4;
    for (int i = 6, j = 3; i >= 0; i--, j++) {
        quadSums[i] = quadSums[i + 1] + j * j;
    }
    return quadSums[0];
}
```

- [x]
```
public static long quadSumFor() {
    long quadSum = 0L;
    for (int i = 2; i < 10; i++) {
        quadSum += i * i;
    }
    return quadSum;
}
```

- [ ]
```
public static long quadSumWhile() {
    long quadSum = 0, i = 10;
    while (--i >= 2) {
        quadSum += i * i;
    }
    return quadSum;
}
```

- [x]
```
public static long quadSumForEach() {
    long[] quads = new long[8];
    long quadSum = 0;
    for (int i = 0; i < quads.length; i++) {
        quads[i] = (i + 2) * (i + 2);
    }
    for (long quad : quads) {
        quadSum += quad;
    }
    return quadSum;
}
```

- [ ]
```
public static long quadSumDo() {
    long quadSum = 0, i = 9;
    do {
        quadSum *= i--;
    } while (i > 0);
    return quadSum;
}
```