PK-Einstufungstest

64 / 100 Punkte

1. Allgemeine Multiple-Choice-Aufgaben

11 / 15 Punkte

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

Aufgabe 1.1.

2 / 5 Punkte

Wählen Sie jene Ausdrücke aus, die in Java den `int`-Wert `7` ergeben:

- [ ] `2 | 7`
- [x] `012 - 5`
- [ ] `15 & 64`
- [ ] `256 % 8`
- [x] `32 >> 2`

Aufgabe 1.2.

4 / 5 Punkte

Wählen Sie jene Ausdrücke aus, die in Java Literale des Typs `String` sind:

- [ ] `0X0B`
- [ ] `"\""`
- [x] `"C\043F"`
- [x] `"0\nB\tf\r0"`
- [x] `"000L"`

Aufgabe 1.3.

5 / 5 Punkte

Angenommen, `a` und `b` sind initialisierte `boolean`-Variablen. Wählen Sie jene Java-Anweisungen aus, durch die maximal ein Zeichen ausgegeben wird:

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

2. Allgemeine Auswahlaufgaben

12 / 15 Punkte

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

Aufgabe 2.1.

0 / 3 Punkte

Die Auswertung von `2 + (8 + "")` liefert in Java folgendes Ergebnis:

- ○ `10`
- ○ `28`
- ● `"28"`
- ○ `"10"`
- ● keines davon

Aufgabe 2.2.

3 / 3 Punkte

Der Ausdruck `0 + 'F'` liefert in Java ein Ergebnis vom Typ:

- ○ `char`
- ○ `String`
- ● `int`
- ○ `float`
- ○ keinem davon

Aufgabe 2.3.

3 / 3 Punkte

`do { f(x); } while (x-- > 0);` ist in Java äquivalent zu:

- ○ `f(x); if (x-- > 0) { f(x); }`
- ○ `f(x); switch (x) { case 0: break; default: f(x); }`
- ○ `f(x); while (--x > 0) { f(x); }`
- ● `f(x); for (; x-- > 0; f(x));`
- ○ `if (!(--x > 0)) { f(x); }; f(x);`

Aufgabe 2.4.

3 / 3 Punkte

Die Auswertung von `3F + 2L` liefert in Java folgendes Ergebnis:

- ○ `5FL`
- ● `5F`
- ○ `"FFFLL"`
- ○ `5L`
- ○ keines davon

Aufgabe 2.5.

3 / 3 Punkte

Angenommen, `s` ist ein als `final` deklarierter formaler Parameter vom Typ `String`.
Wählen Sie die erlaubte Verwendung von `s`:

- ○ `while (s < 0) { s++; }`
- ○ `s += "C";`
- ○ `int i = (s = "0");`
- ● `System.out.println(s + 9);`
- ○ `s = "0DC" + 'C';`

3. Multiple-Choice-Aufgaben zu Ausdrücken und Bedingungen

17 / 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.

3 / 5 Punkte

```
// returns a string containing the length of s;
// e.g., withLength("xx") gives "2"; if s is null, then the result is "0"
public static String withLength(String s) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [x] `return (s.charAt(0) == null) ? "0" : s.length() + "";`
- [x] `return ((s == null) ? "" : s).length() + "";`
- [x] `return (s != null) ? "" + s.length() : "0";`
- [x] `return "" + ((s == null) ? 0 : s.length());`
- [x] `return ("xx".equals(s)) ? "2" : "" + s.length();`

Aufgabe 3.2.

5 / 5 Punkte

```
// returns x % y if x >= y, returns y % x if x <= y;
// x != 0 and y != 0 is assumed
public static int fromIntsToInt(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return (x == y) ? 1 : (Math.abs(x) % Math.abs(y));`
- [ ] `return (Math.min(x % y, 1) < 1) ? (x % y) : (y % x);`
- [x] `return (x > y) ? (x % y) : (y % x);`
- [ ] `return (x % y) ^ (y % x);`
- [x] `return (x < y) ? (y % x) : (x % y);`

Aufgabe 3.3.

5 / 5 Punkte

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

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

Aufgabe 3.4.

4 / 5 Punkte

```
// returns true if (and only if) neither x nor y is in the range between 0 and 100
public static boolean fromIntsToBool(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return ((x & y) < 0) && ((x & y) > 100);`
- [ ] `return !(0 <= (x | y) <= 100);`
- [x] `return !((x >= 0) && (x <= 100)) && !((y >= 0) && (y <= 100));`
- [ ] `return ((x & y) < 0) || ((x & y) > 100);`
- [ ] `return ((0 > x) || (100 < x)) && ((0 > y) || (100 < y));`

4. Auswahlaufgaben zu Programmverzweigungen

5 / 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 product (1.0/2) * (1.0/3) * ... * (1.0/n) if n > 1;
// returns 1.0 if n <= 1
public static double prodRec(final int n) {
    if (A) {
        return B;
    }
    return prodRec(C) * (1.0 / n);
}
```

A:

- (●) `n <= 1`
- ( ) `n > 1`
- ( ) `n >= 1`
- ( ) `n == 1`
- ( ) `n != 1`

B:

- ( ) `n + 1`
- ( ) `n - 1`
- ( ) `'1.0'`
- (●) `1D`
- ( ) `"1.0"`

C:

- ( ) `n + 1`
- (●) `n - 1`
- ( ) `'1.0'`
- ( ) `1D`
- ( ) `"1.0"`

Aufgabe 4.2.

0 / 5 Punkte

```
// returns "!" if left equals "left" and right equals "right",
// returns "?!" if either left equals "left" or right equals "right" (but not b
// returns "?" if left differs from "left" and right differs from "right"
public static String combString(String left, String right) {
    String result = "";
    if (A) {
        result = B;
    }
    if (C) {
        result += D;
    }
    return result;
}
```

A:

- ( ) `"left".equals(left)`
- ( ) `!("left".equals(left) && "right".equals(right))`
- (●) `"left".equals(left) && "right".equals(right)`
- ( ) `"right".equals(right)`
- ( ) `"left".equals(left) || "right".equals(right)`

B:

- [ ] `""`
- [x] `"!"`
- [ ] `"?"`
- [ ] `"!?"`
- [ ] `"?!"`

C:

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

D:

- [ ] `""`
- [ ] `"!"`
- [ ] `"?"`
- [ ] `"!?"`
- [ ] `"?!"`

Aufgabe 4.3.

0 / 5 Punkte

```
// returns the smallest index i where i >= low and a[i] is odd (ungerade);
// returns -1 if there is no such index;
// a != null and low >= 0 always hold
public static int index(final int low, final int[] a) {
    if (A) {
        return -1;
    } else if (B) {
        return index(C, a);
    }
    return D;
}
```

A:

- [ ] `low > a.length`
- [ ] `low == 0`
- [ ] `low >= a.length`
- [ ] `low <= a.length`
- [ ] `low < a.length`

B:

- [ ] `(a[low] % 2) == 1`
- [ ] `(low[a] % 2) == 0`
- [ ] `(low[a] % 2) == 1`
- [ ] `(a[low] % 2) == 0`
- [ ] `(low[a] % 2) < 0`

C:

- [ ] `low + 1`
- [ ] `1`
- [ ] `low`
- [ ] `-1`
- [ ] `low - 1`

D:

- ○ `low + 1`
- ○ `1`
- ● `low`
- ○ `-1`
- ○ `low - 1`

Aufgabe 4.4.

0 / 5 Punkte

```
// returns "divisible by 3" if i is divisible by 3, returns "not divisible by 3
public static String divisible(int i) {
    switch (A) {
        case 1:
            return B;
        default:
            return C;
    }
}
```

A:

- ○ `i`
- ● `(i % 3) == 0`
- ○ `(i % 3) + 1`
- ○ `(i % 3) - 1`
- ○ `i % 3`

B:

- ● `"divisible by 3"`
- ○ `"not divisible by 3"`

C:

- ○ `"divisible by 3"`
- ● `"not divisible by 3"`

5. Auswahlaufgaben zu Schleifen

10 / 15 Punkte

Bitte beantworten Sie jede dieser Fragen durch Auswahl der einen zutreffenden Antwortmöglichkeit.

Aufgabe 5.1.

5 / 5 Punkte

```
public static void forLoop() {
    for (int i = 0; i <= 200; i += 4) {
        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.

5 / 5 Punkte

```
public static void forEachLoop() {
    int sum = 0;
    int[] is = { 1 + 2, 7 / 2, 2 << 2, 'c' - 'a' };
    for (int i : is) {
        sum += i;
    }
    System.out.println(sum);
}
```

Welche Zahl wird von `forEachLoop` ausgegeben?

- ○ weniger als 16
- ● 16
- ○ 17
- ○ 18
- ○ mehr als 18

Aufgabe 5.3.

0 / 5 Punkte

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

Welche Zahl wird durch einen Aufruf von `whileLoop(3)` ausgegeben?

- ○ weniger als 4
- ○ 4
- ○ 5
- ● 6
- ○ mehr als 6

6. Multiple-Choice-Aufgabe zu Schleifen

9 / 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. 9 / 15 Punkte

Welche Methoden geben als Ergebnis die Summe `1.0/2.0 + 1.0/4.0 + ... + 1.0/48.0 + 1.0/50.0` (also jeweils durch die geraden Zahlen zwischen 2 und 50 dividiert) zurück?

- [x]
```java
public static double fracSumForEach() {
    double[] fracs = new double[25];
    double fracSum = 0.0;
    for (int i = 1; i <= fracs.length; i++) {
        fracs[i - 1] = 1.0 / (i * 2);
    }
    for (double frac : fracs) {
        fracSum += frac;
    }
    return fracSum;
}
```

- [x]
```java
public static double fracSumFor() {
    double fracSum = 0.0;
    for (double d = 2.0; d < 51.0; d += 2.0) {
        fracSum += 1.0 / d;
    }
    return fracSum;
}
```

- [x]
```java
public static double fracSumDo() {
    double fracSum = 0.0, d = 0.0;
    do {
        fracSum += 1.0 / (d += 2.0);
    } while (d < 49.0);
    return fracSum;
}
```

- [ ]
```java
public static double fracSumArray() {
    double[] fracSums = new double[25];
    fracSums[0] = 0.5;
    for (int i = 1; i < 25; i++) {
        fracSums[i] = fracSums[i - 1] + 1.0 / (i * 2 + 2);
    }
    return fracSums[24];
}
```

- [ ]
```java
public static double fracSumWhile() {
    double fracSum = 0.0, d = 52.0;
    while (d > 3.0) {
        fracSum += 1.0 / (d -= 2.0);
    }
    return fracSum;
}
```