PK-Einstufungstest

100 Punkte

1. Allgemeine Multiple-Choice-Aufgaben

15 Punkte

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

Aufgabe 1.1. 5 Punkte

Wählen Sie jene Ausdrücke aus, die in Java den `int`-Wert `33` ergeben:

- [ ] `132 >> 2`
- [ ] `33.5 - 0.5`
- [x] `37 & 35`
- [ ] `30 | 3`
- [ ] `3 << 264`

Aufgabe 1.2. 5 Punkte

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

- [ ] `9`
- [ ] `"8"`
- [ ] `'\ucA0c'`
- [ ] `'\004'`
- [x] `'8'`

Aufgabe 1.3. 5 Punkte

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

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

2. Allgemeine Auswahlaufgaben

15 Punkte

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

Aufgabe 2.1. 3 Punkte

`for (; a = b; f(a, b = !b));` ist in Java äquivalent zu:

- ( ) `if (a = (b = !b)) { f(a, b); }`
- (•) `while (a == b) { f(a, b = !b); }`
- ( ) `for (; b = !b; f(a, a = b));`
- ( ) `switch (b) { case a: f(a, b = !b); }`
- ( ) `for (; a = b; b = !b, f(a, b));`

Aufgabe 2.2. 3 Punkte

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

( ) `FFFDDD` ( ) `6J` (•) `6.0` ( ) `'3F3D'` ( ) keines davon

Aufgabe 2.3. 3 Punkte

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

- ( ) `c = "10";`
- (•) `System.out.println("c = " + c);`

- [ ] `c += 2;`
- [ ] `int i = (c = '0');`
- [ ] `while (c > 'a') c--;`

Aufgabe 2.4. 3 Punkte

Der Ausdruck `'X' + "Y"` liefert in Java ein Ergebnis vom Typ:

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

Aufgabe 2.5. 3 Punkte

Die Auswertung von `"" + (9 + 9)` liefert in Java folgendes Ergebnis:

- [ ] `"99"`
- [ ] `99`
- [ ] `"18"`
- [x] `18`
- [ ] keines davon

3. Multiple-Choice-Aufgaben zu Ausdrücken und Bedingungen

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. 5 Punkte

```
// returns the string constructed from the non-empty strings r and s by appending the longer string at the end of the shorter o
// e.g., constr("xx", "yyy") gives "xxyyy" and constr("xxx", "yy") gives "yyxxx";
// constr("xx", "yy") gives either "xxyy" or "yyxx";
// s and r are never null
public static String constr(String r, String s) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return (r.charAt(0) > s.charAt(0)) ? (s + r) : (r + s);`
- [ ] `return (r.equals("xx") || s.equals("yy")) ? (r + s) : (s + r);`
- [ ] `return (r.charAt(r.length()) > s.charAt(s.length())) ? (s + r) : (r + s);`
- [ ] `return Math.min(r, s) + Math.max(r, s);`
- [x] `return (r.length() > s.length()) ? (s + r) : (r + s);`

Aufgabe 3.2. 5 Punkte

```
// returns true if (and only if) x as well as y is in the range between 0 and 100 (including 0 and 100)
public static boolean fromIntsToBool(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

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

Aufgabe 3.3. 5 Punkte

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

- [x] `return (a && !b) ? 1 : 0;`
- [ ] `return ((a != true) && (b != false)) ? 0 : 1;`
- [ ] `return (a ^ (b && a)) ? 0 : 1;`
- [ ] `return (a ? 1 : 0) * (b ? 1 : 0);`
- [x] `return a ? (b ? 0 : 1) : 0;`

Aufgabe 3.4. 5 Punkte

```
// returns -1 if x < y, returns 0 if x == y, returns 1 if x > y
public static int fromIntsToInt(int x, int y) {
    // TODO: Welche return-Anweisung kann hier stehen?
}
```

- [ ] `return (x == y) ? 0 : (x - y) / Math.abs(x - y);`
- [x] `return (x < y) ? -1 : ((x == y) ? 0 : 1);`
- [ ] `return (x - y) - (y - x);`
- [ ] `return (Math.abs(x - y) <= 1) ? (x - y) : (y - x);`
- [x] `return ((x < y) ? -1 : 0) + ((x > y) ? 1 : 0);`

4. Auswahlaufgaben zu Programmverzweigungen

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 Punkte

```
// returns "!!!" if left equals "left" and right equals "right",
// returns "!!" if either left equals "left" or right equals "right" (but not both),
// 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;
    }
    else if (C) {
        result += D;
    }
    return result;
}
```

A:

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

B:

- ○ `""`
- ○ `"!"`
- ● `"!!"`
- ○ `"!!!"`
- ○ `"!!!!"`

C:

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

D:

- ○ `""`
- ● `"!"`
- ○ `"!!"`
- ○ `"!!!"`
- ○ `"!!!!"`

Aufgabe 4.2. 5 Punkte

```
// returns the sum 1.0/1 + 1.0/2 + ... + 1.0/n if n >= 1;
// returns 0.0 if n <= 0
public static double sumRec(final long n) {
    if (A) {
        return B;
    }
    return sumRec(C) + (1.0 / n);
}
```

A:

- ● `n <= 0`
- ○ `n == 0`
- ○ `n > 0`
- ○ `n >= 0`
- ○ `n != 0`

B:

- ○ `n + 1`
- ○ `n - 1`
- ○ `'0.0'`
- ● `0D`
- ○ `"0.0"`

C:

- ○ `n + 1`
- ● `n - 1`
- ○ `'0.0'`
- ○ `0D`
- ○ `"0.0"`

Aufgabe 4.3. 5 Punkte

```
// returns the largest index i where i < high and a[i] differs from 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 index(C, c, a);
    }
    return D;
}
```

A:

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

B:

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

C:

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

D:

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

Aufgabe 4.4. 5 Punkte

```
// returns "large" if a contains more than 100 elements, returns "small" otherwise
public static String largeArray(int[] a) {
    switch (A) {
        case 100:
            return B;
        default:
            return C;
    }
}
```

A:

- ○ `a.length`
- ○ `a.length <= 100`
- ● `Math.max(a.length, 100)`
- ○ `a.length >= 100`
- ○ `Math.min(a.length, 100)`

B:

- ○ `"small"`
- ● `"large"`

C:

- ● `"small"`
- ○ `"large"`

5. Auswahlaufgaben zu Schleifen

15 Punkte

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

Aufgabe 5.1. 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(3)` ausgegeben?

- ● weniger als 5
- ○ 5
- ○ 6
- ○ 7
- ○ mehr als 7

Aufgabe 5.2. 5 Punkte

```java
public static void forLoop() {
    for (int i = 1; i < 100; i += 2) {
        System.out.println(i);
    }
}
```

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

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

Aufgabe 5.3. 5 Punkte

```java
public static void forEachLoop() {
    int min = 1000;
    int[] is = { 10 + 10, 69 / 2, 24 << 2, 4 * 7 };
    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

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. 15 Punkte

Welche Methoden geben als Ergebnis die Summe `1.0/1.0 + 1.0/2.0 + ... + 1.0/9.0 + 1.0/10.0` zurück?

- ☑
```java
public static double fracSumArray() {
    double[] fracSums = new double[10];
    fracSums[0] = 1.0;
    for (int i = 1; i < 10; i++) {
        fracSums[i] = fracSums[i - 1] + 1.0 / (i + 1);
    }
    return fracSums[9];
}
```
- ☐
```java
public static double fracSumForEach() {
    double[] fracs = new double[10];
    double fracSum = 0.0;
    for (int i = 0; i < fracs.length; i++) {
        fracs[i] = 1.0 / i;
    }
    for (double frac : fracs) {
        fracSum += frac;
    }
    return fracSum;
}
```
- ☐
```java
public static double fracSumFor() {
    double fracSum = 0.0;
    for (double d = 0.0; d < 10.5; d += 1.0) {
        fracSum += 1.0 / d;
    }
    return fracSum;
}
```
- ☑
```java
public static double fracSumDo() {
    double fracSum = 0.0, d = 0.0;
    do {
        fracSum += 1.0 / (d += 1.0);
    } while (d < 9.5);
    return fracSum;
}
```

```java
public static double fracSumWhile() {
    double fracSum = 0.0, d = 11.0;
    while ((d -= 1.0) > 0.5) {
        fracSum += 1.0 / d;
    }
    return fracSum;
}
```