

Test 1 in Programmkonstruktion

84 / 100 Punkte

Alle Aufgaben beziehen sich auf Java.

1. Multiple-Choice-Aufgaben

39 / 50 Punkte

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

Aufgabe 1.1.

3 / 4 Punkte

Welche der folgenden Symbolfolgen bzw. Ausdrücke liefern dasselbe Ergebnis (in Wert und Typ) wie das Literal `97` (wobei 97 der ASCII-Wert von `'a'` ist)?

`'a' + 0`

`96.0 + 1.0`

`98 - 1`

`'a'`

Aufgabe 1.2.

2 / 4 Punkte

Welche der folgenden Ausdrücke liefern dasselbe Ergebnis (in Wert und Typ) wie das Literal `"1234"`?

`1 + '2' + 3 + "4"`

`1 + 2 + 3 + "4"`

`1 + "2" + '3' + 4`

`'1' + '2' + '3' + "4"`

Aufgabe 1.3.

1 / 4 Punkte

Welche der folgenden Anweisungen bzw. Anweisungsfolgen sind gültig, d.h. werden vom Java-Compiler unter bestimmten Umständen akzeptiert?

`return (a > 0 ? -0 : -a + 'a');`

`while (!a) { return !!a; }`

`if (--a < b--) { c = b > 0; }`

`if (a) { a = 'a'; }`

Aufgabe 1.4.

2 / 4 Punkte

Welche der folgenden Anweisungen bzw. Anweisungsfolgen sind gültig, d.h. werden vom Java-Compiler akzeptiert?

`do { int i = 0; i++; } while (i < 10);`

`int i; do { i = 0; } while (i < 10);`

`for (int i = 0; i < 10; i++) { int i = 0; }`

`int i = 0; do (i++) while { i < 10 };`

Aufgabe 1.5.

4 / 4 Punkte

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

`y + x--`

`y + (x-1)`

`(x -= 1) + y`

`x + y - 1`

Aufgabe 1.6.

3 / 4 Punkte

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

`x *= ++y`

`x * (y+1)`

`x = x * y + 1`

`x *= y++`

Aufgabe 1.7.

4 / 4 Punkte

Angenommen, `a` und `b` seien initialisierte `boolean`-Variablen. Welche der folgenden Ausdrücke liefern immer `true` zurück?

`(!a && b) || (a && !b)`

`(!a && b) != ((a == b) && b)`

`(a && !a) == false`

`!(a == b) || !(a || b)`

Aufgabe 1.8.

8 / 8 Punkte

Welche der folgenden Methoden liefern als Ergebnis die Summe von `a` und `b`? Es gilt: `a >= 0` und `b >= 0`.

```
public static int sum(int a, int b) {  
    while (--a > 0) {  
        ++b;  
    }  
    return b;  
}
```



```
public static int sum(int a, int b) {  
    for (; b > 0; b--) { a++; }  
    return a;  
}
```



```
public static int sum(int a, int b) {  
    if (b == 0) return a;  
    return sum(++a,--b);  
}
```



```
public static int sum(int a, int b) {  
    for (int i = 0; i <= b; i++) {  
        a++;  
    }  
    return a;  
}
```

Aufgabe 1.9.

6 / 8 Punkte

```
public static void f(boolean a) {  
    if (g(a)) {  
        System.out.println(" World");  
    } else {  
        f(!g(a));  
    }  
}
```

Welche der folgenden möglichen Implementierungen von `g` sorgen dafür, dass sowohl `f(true)` als auch `f(false)` den String "Hello World" ausgeben?

```
public static boolean g(boolean a) {  
    if (a) {  
        return false;  
    } else {  
        System.out.print("Hello");  
        return true;  
    }  
}
```

```
public static boolean g(boolean a) {  
    System.out.print("Hello");  
    if (a) {  
        return a;  
    } else {  
        return false;  
    }  
}
```

```
public static boolean g(boolean a) {  
    if (a) {  
        System.out.print("Hello");  
        return false;  
    } else {  
        return true;  
    }  
}
```

```
public static boolean g(boolean a) {  
    if (a) {  
        System.out.print("Hello");  
    }  
    return a;  
}
```

Aufgabe 1.10.

6 / 6 Punkte

Welche der folgenden Methoden berechnen die Summe aller ungeraden Zahlen von 1 bis einschließlich n (für $n > 0$)?

```
public static int f(int n) {  
    int y = 1;  
    do {  
        n--;  
        if (n % 2 != 0) {  
            y++;  
        }  
    } while (n > 0);  
    return y;  
}
```

public static int x(int n) {
 int a = 0;
 int b = 0;
 for (b = 1; b <= n; b++) {
 if (b % 2 == 1) {
 a += b;
 }
 }
 return a;
}

```
public static int a(int n) {  
    int x = 1;  
    while (n > 0) {  
        n--;  
        if (n % 2 == 0) {  
            x += n;  
        }  
    }  
    return x;  
}
```

2. Single-Choice-Aufgaben

30 / 30 Punkte

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

Aufgabe 2.1.

5 / 5 Punkte

```
public static int loop(int n) {  
    int x = 0;  
    for (int i = 0; i <= 100; i += n) {  
        if (i % 2 == 1) {  
            x++;  
        }  
    }  
    return x;  
}
```

Welche Zahl wird durch einen Aufruf von `loop(5)` zurückgegeben?

- 10 11 20 21 50

Aufgabe 2.2.

5 / 5 Punkte

```
public static int fab(int n) {  
    if (n < 3) {  
        return 1;  
    }  
    return fab(n-2) + fab(n-1);  
}
```

Welche Zahl wird durch einen Aufruf von `fab(4)` zurückgegeben?

- 2 3 4 5 6

Aufgabe 2.3.

5 / 5 Punkte

```
public static void go(int a, int b) {  
    int i = 0;  
    while (a < b) {  
        i++;  
        if (b % a == i) {  
            System.out.println(7);  
        }  
        a += i;  
        i--;  
    }  
}
```

Wieviele Zeilen werden durch einen Aufruf von `go(5, 10)` ausgegeben?

- 0 1 2 5 10

Aufgabe 2.4.

5 / 5 Punkte

```
public static int g() {  
    int a = 3;  
    a += f(a);  
    return a;  
}  
  
public static int f(int a) {  
    a--;  
    int x = a;  
    a--;  
    x++;  
    return x;  
}
```

Welchen Wert liefert `g()` zurück?

- 2 3 4 5 6

Aufgabe 2.5.

5 / 5 Punkte

```
public static int q() {  
    int x = 128;  
    p(x);  
    p(x);  
    return x;  
}  
  
public static void p(int x) {  
    if (x > 8) {  
        p(x/2);  
    } else {  
        x = 64;  
    }  
}
```

Welchen Wert liefert `q()` zurück?

- 0 8 16 32 64 128

Aufgabe 2.6.

5 / 5 Punkte

`if (a || b) { f(c); }` ist in Java äquivalent zu:

- `if (a && b) { } else { f(c); }`
- `while (a || b) { f(c); break; }`
- `f(c);`
- `(a || b) ? f(c);`
- `if (a) { if (b) { f(c); } }`

3. Auswahlaufgaben zu Programmverzweigungen

15 / 20 Punkte

In den Methoden sind die Buchstaben A, B, C bis höchstens D jeweils durch einen der vorgeschlagenen Programmteile 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 3.1.

5 / 5 Punkte

```
// liefert 1 wenn a kleiner als b ist, 0 wenn a gleich b ist und -1 sonst
public static int compare(int a, int b) {
    switch (A) {
        case 0:
            return B;
    }
    return C;
}
```

A:

- `a == b ? -1 : 0`
- `a`
- `a < b ? 0 : a`
- `-1`
- `a < b ? 1 : -1`
- `b`
- `0`

B:

- `-1`
- `a < b ? 1 : -1`
- `1`
- `b`
- `0`
- `a < b ? 0 : -1`
- `a`

C:

a == b ? -1 : 0 a a < b ? 0 : a -1

a < b ? 1 : -1 b 0

Aufgabe 3.2.

5 / 5 Punkte

```
// liefert die Namen der Parameter, deren Wert mit dem Wert
// vom Parameter a übereinstimmen,
// genauer gesagt:
// "b" wenn a == b und a != c,
// "c" wenn a != b und a == c,
// "b,c" wenn a == b und a == c,
// "!" wenn a != b und a != c.
// Hinweis: Alle Parameter sind vom Typ boolean,
// d.h. können nur true oder false sein.
public static String compare(boolean a, boolean b, boolean c) {
    String result = A;

    if (a == b) {
        if (B) {
            result += ",c";
        }
    } else if (C) {
        result = "!";
    } else {
        result = D;
    }

    return result;
}
```

A:

"b,c" "c" "b" "

B:

a == b a != b b != c b == c

C:

a == b a != b b != c b == c

D:

- "b, c"
 "c"
 "b"
 "

Aufgabe 3.3.

5 / 5 Punkte

```
// liefert true, wenn alle Ziffern bis zum höchsten Stellenwert
// der Binärdarstellung von d Einser sind.
// Z.B.:
// onlyOnesBinary(3) liefert true,
// onlyOnesBinary(7) liefert true,
// onlyOnesBinary(8) liefert false,
// onlyOnesBinary(1) liefert true,
// onlyOnesBinary(0) liefert false.
// Annahme: d >= 0
public static boolean onlyOnesBinary(int d) {
    do {
        if (A) {
            return B;
        }
        d/=2;
    } while (C);
    return D;
}
```

A:

- false
 $d \% 2 == 0$
 $d > 0$
 $d \% 10 == 0$

 $d > 1$
 true

B:

- false
 $d \% 2 == 0$
 $d > 0$
 $d \% 10 == 0$

 $d > 1$
 true

C:

- false
 $d \% 2 == 0$
 $d > 0$
 $d \% 10 == 0$

 $d > 1$
 true

D:

- d != 0 d > 0 true d == 1 d > 1
- false

Aufgabe 3.4.

0 / 5 Punkte

```
// liefert die Summe der Zahlen im Intervall von x bis y (inklusive x und y).  
// Annahme: x <= y  
public static int sum(int x, int y) {  
    if (A >= B) {  
        return x + C; //Rekursionsanfang (sorgt für Fundiertheit)  
    }  
    return y + D; //Rekursionsschritt  
}
```

A:

- y sum(x, y-1) sum(x+1, y) x 0
- sum(x-1, y)

B:

- y sum(x, y-1) sum(x+1, y) x 0
- sum(x-1, y)

C:

- y sum(x, y-1) sum(x+1, y) x 0
- sum(x-1, y)

D:

y

sum(x, y-1)

sum(x+1, y)

x

0

sum(x-1, y)