

Probeklausur
„Grundlagen der Programmierung“ im SoSe 2022
31.05.2022

Vorname:	
Nachname:	
Matrikelnummer:	

Hiermit bestätige ich, von den folgenden Punkten Kenntnis genommen zu haben:

1. Schreiben Sie auf JEDES Blatt Ihren Namen und Ihre Matrikelnummer.
2. Es sind keine Hilfsmittel zugelassen. Auf Ihrem Platz dürfen sich nur Ihr Schreibmaterial, evtl. bereitgestelltes Konzeptpapier und dieses geheftete Exemplar befinden. Eigenes Papier ist nicht erlaubt.
3. Der Test besteht aus 7 Aufgaben für die es insgesamt 24 Punkte gibt.
4. Alle Aufgaben des Tests sind zu bearbeiten. Es gibt keine optionalen Teile.
5. Bitte prüfen Sie die Vollständigkeit Ihres Exemplars anhand der abgedruckten Seitennummern (10 Seiten, inklusive Deckblatt)!
6. Verwenden Sie ausschließlich den gehefteten Test für die Beantwortung der Fragen.
7. Bei der Korrektur des Tests werden nur Lösungen auf den dafür vorgesehenen Lösungsblättern gewertet. Lösungen auf Konzeptpapier werden nicht gewertet.
8. Alle Arten elektronischer Geräte wie Handys, Wearables, Bluetooth Sets, Mini-Kameras etc. müssen ausgeschaltet und in der Jacke/Tasche am Rand des Raumes verstaut werden! - Das Vorhandensein eines Geräts am Platz gilt als Täuschungsversuch!
9. Ein Täuschungsversuch jeglicher Art in dieser Teilprüfungsleistung führt dazu, dass das gesamte Modul mit „nicht ausreichend“ (5,0) bewertet wird.

Unterschrift

Aufgabe	1	2	3	4	5	6	7		Summe
Punkte	/3	/3	/2	/4	/3	/6	/3		/24

Aufgabe 1: Datentypen und Ausdrücke

(3 Punkte)

Zu welchem Datentyp und Wert werten die folgenden Ausdrücke aus?

Ausdruck	Datentyp	Wert
<code>89 % 9</code>		
<code>0 % 2</code>		
<code>8 / 5.0</code>		
<code>10.0 / 2.0</code>		
<code>5 < (true && false)</code>		
<code>(7 > 9)</code>		

Aufgabe 2: Programmcode

(3 Punkte)

Die Methode *textAusgabe* soll den String im Eingabeparameter *text* 5 mal auf der Konsole ausgeben. Die Methode hat keinen Rückgabewert.

```
1    public static void textAusgabe(String text) {  
2  
3        int i = 0;  
4  
5        while (i < 5) {  
6  
7            System.out.println(text);  
8  
9        }  
10 }
```

1. Die Methode setzt die gewünschte Funktionalität nicht um. Warum?
2. Wie müsste die Methode erweitert werden, so dass die gewünschte Funktionalität umgesetzt wird? Geben Sie den Programmcode und die Zeile(n) an, in welcher/welchen dieser eingefügt werden müsste.

Aufgabe 3: Programmcode

(2 Punkte)

Die Methode *sum* soll drei ganze Zahlen als Eingabeparameter nehmen, sie aufsummieren und das Ergebnis zurückgeben.

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

Das Programm kompiliert nicht. Beschreiben Sie, um welchen Fehler es sich handelt.

Aufgabe 4: Boolesche Werte

(4 Punkte)

Betrachten Sie die Methode *computeBs*, welche die drei booleschen Werte *x*, *y* und *z* als Eingabeparameter hat und einen booleschen Wert berechnet und diesen zurückgibt.

```
public static boolean computeBs(boolean x, boolean y, boolean z){  
    if (x && y) {  
        return true;  
    }  
    if (!x && !y) {  
        return true;  
    }  
    return false;  
}
```

Vervollständigen Sie die nachfolgende Tabelle gemäß der Methode *computeBs*.

x	y	z	Ergebnis
true	true	true	
true	true	false	
true	false	true	
true	false	false	
false	true	true	
false	true	false	
false	false	true	
false	false	false	

Aufgabe 5: Programmcode

(3 Punkte)

Welche Aussage(n) trifft/treffen auf die folgende Methode zu?

```
public static int compute(int k) {  
    int erg = 1;  
    for (int i = 0; i < k; i++) {  
        erg = erg * 2;  
    }  
    return erg;  
}
```

Für $k=0$ liefert die Methode den Wert 1.☐Für $k \geq 0$ berechnet die Methode 2^k (2 hoch k).☐Für $k = 3$ liefert die Methode den Wert 6.☐Für $k < 0$ liefert die Methode den Wert 0.☐Für $k < 0$ kann die Methode nicht ausgeführt werden.☐

Aufgabe 8: Programmieren

(6 Punkte)

Schreiben Sie eine Methode *printConsole* welche eine ganze Zahl n als Eingabe nimmt und die folgende Konsolenausgabe generiert:

```
0
2
4
6
8
10
12
14
16
18
...
2*n
```

(n ist dabei natürlich der zur Laufzeit eingesetzte Wert)

Aufgabe 9: Programmieren

(3 Punkte)

Schreiben Sie eine Methode *computeSum*, welche eine natürliche Zahl n als Eingabe nimmt, die Summe $\sum i$ (von 0 bis n) berechnet und den berechneten Wert zurückgibt.

Die Summe $\sum i$ ist wie folgt definiert:

Für 0 : 0

Für n : $n + (n-1) + (n-2) + \dots + 1$

Beispiele:

Für $n = 6$ ist die Summe: $6+5+4+3+2+1 = 21$

Für $n = 10$ ist die Summe: $10+9+8+7+6+5+4+3+2+1 = 55$

Name

Matrikelnummer:

Notizen

Name

Matrikelnummer:

Notizen