

**Probeklausur**  
**„Grundlagen der Programmierung“ im WiSe 2023-2024**  
**19.12.2023**

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 darf sich nur Ihr Schreibmaterial befinden. Eigenes Papier ist nicht erlaubt.
3. Jeglicher Code ist in der Programmiersprache Java zu schreiben.
4. Der Test besteht aus 8 Aufgaben für die es insgesamt 24 Punkte gibt.
5. Alle Aufgaben des Tests sind zu bearbeiten. Es gibt keine optionalen Teile.
6. Bitte prüfen Sie die Vollständigkeit Ihres Exemplars anhand der abgedruckten Seitennummern (12 Seiten, inklusive Deckblatt)!
7. Verwenden Sie ausschließlich die gehefteten Blätter für die Beantwortung der Fragen. Bei der Korrektur der Aufgaben werden nur Lösungen auf den dafür vorgesehenen Lösungsblättern 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	8	Summe
Punkte	/3	/4	/4	/4	/3	/6	/6	/10	/40

**Aufgabe 1: Datentypen und Ausdrücke**

(3 Punkte)

- a) Deklarieren Sie eine Variable mit dem Variablennamen *zaehler* vom Datentyp *int* und initialisieren Sie die Variable mit dem Wert 5.
- b) Deklarieren Sie eine Variable mit dem Variablennamen *vorname* vom Datentyp *String*.
- c) Zu welchem Datentyp und Wert werten die folgenden Ausdrücke aus?

Ausdruck	Datentyp	Wert
$3 \% 5$		
$8.2 * 1.5$		
"Ana" + "nas"		
$8 / 5$		

**Aufgabe 2: Programmcode**

(4 Punkte)

Die Methode *textAusgabe* soll den String im Eingabeparameter *text* so oft auf der Konsole ausgeben, wie der Wert des Eingabeparameters *num*.

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

1. Die Methode setzt die gewünschte Funktionalität nicht um. Warum?
2. Wie müsste die Methode angepasst 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**

(4 Punkte)

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

```
public static int sum(int a, int b, int c) {  
    {    int erg = 1;  
        erg = erg * a * b * c  
  
        return erg;  
    }
```

Das Programm kompiliert nicht. Beschreiben Sie, um welche(n) Fehler es sich handelt und markieren Sie diesen im Code.

**Aufgabe 4: Boolesche Werte**

(4 Punkte)

Gegeben sei der folgende Ausdruck:

```
boolean erg = (!x && !y) || (z ^ x);
```

Vervollständigen Sie die nachfolgende Tabelle für den gegebenen Ausdruck:

x	y	z	erg
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 * 3;  
    }  
    return erg;  
}
```

Für  $k=0$  liefert die Methode den Wert 1.☐Für  $k = 3$  liefert die Methode den Wert 6.☐Für  $k < 0$  liefert die Methode den Wert 1.☐Für  $k < 0$  kann die Methode nicht ausgeführt werden.☐

**Aufgabe 6: Programmieren**

(6 Punkte)

- a) Schreiben Sie eine Methode *countTrueBooleans*, die für ein als Parameter übergebenes Array die Anzahl an booleschen Werten mit dem Wert `true` zurückgibt.

Beispiel: für ein Array

```
{ true, false, false, true, true }
```

liefert die Methode *countTrueBooleans* den Wert 3.

- b) Versetzen Sie Ihre Methode mit einem entsprechenden Dokumentationskommentar (Javadoc). (Tipp: Lassen Sie oberhalb der Methodendefinition Platz für Javadoc).

**Aufgabe 7: Programmieren**

(6 Punkte)

- a) Schreiben Sie eine Methode *calculateSquares*, die für eine gegebene Liste von nicht-negativen ganzen Zahlen die Quadrate dieser Zahlen berechnet und sie auf der Konsole ausgibt.

Beispiel:

Eingabe: [2, 3, 5, 7]

Ausgabe auf der Konsole:

4

9

25

49

- b) versehen Sie Ihre Methode mit einem entsprechenden Dokumentationskommentar (Javadoc). (Tipp: Lassen Sie oberhalb der Methodendefinition Platz für Javadoc).



**Aufgabe 8: Objektorientierung**

(10 Punkte)

Erstellen Sie eine Klasse *Spacecraft* (Raumfahrzeug). Ein Raumfahrzeug soll folgende Eigenschaften (Objektvariablen) besitzen:

Name des Raumfahrzeugs (*name*)

Typ des Raumfahrzeugs (*type*, z.B. Raumschiff, Satellit)

Baujahr des Raumfahrzeugs (*year*)

Aktuelle Geschwindigkeit in km/s (*velocity*)

Die Klasse soll einen Konstruktor haben, der allen Objektvariablen beim Erstellen eines neuen Objekts Werte zuweist. Die aktuelle Geschwindigkeit wird beim Erstellen auf 0 gesetzt.

Die Klasse soll die folgenden Methoden haben:

*accelerate(double acceleration)*: erhöht die Geschwindigkeit des Raumfahrzeugs um den angegebenen Beschleunigungswert *acceleration* in km/s.

*decelerate(double deceleration)*: verringert die Geschwindigkeit des Raumfahrzeugs um den angegebenen Verzögerungswert *deceleration* in km/s. Stellen Sie sicher, dass die Geschwindigkeit nicht negativ wird.

*printSpacecraftDetails()*: gibt alle Informationen über das Raumfahrzeug auf der Konsole aus, einschließlich des Namens, des Typs, des Baujahrs und der aktuellen Geschwindigkeit.

*launch()*: gibt eine Meldung aus, dass das Raumfahrzeug gestartet wurde.

Schreiben Sie darüber hinaus einen Getter für den Namen und die aktuelle Geschwindigkeit und einen Setter für die aktuelle Geschwindigkeit.

Name

Matrikelnummer:

---

Name

Matrikelnummer:

---

## **Notizen**

Name

Matrikelnummer:

---

## **Notizen**