

---

# Programmieren – Wintersemester 2025/26

---

## Übungsblatt 1

Version 1.0

20 Punkte

Ausgabe: 05.11.2025, ca. 12:00 Uhr  
Abgabestart: 12.11.2025, 12:00 Uhr  
Abgabefrist: 20.11.2025, 06:00 Uhr

---

## Plagiarismus

Es werden nur selbstständig angefertigte Lösungen akzeptiert. Das Einreichen fremder Lösungen, seien es auch nur teilweise Lösungen von Dritten, aus Büchern, dem Internet oder anderen Quellen, ist ein Täuschungsversuch und führt jederzeit (auch nachträglich) zur Bewertung „nicht bestanden“. Ausdrücklich ausgenommen hiervon sind Quelltextsnipsel von den Vorlesungsfolien und aus den Lösungsvorschlägen des Übungsbetriebes in diesem Semester. Alle benutzten Hilfsmittel müssen vollständig und genau angegeben werden. Alles, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde, muss deutlich kenntlich gemacht werden. Beachten Sie darüber hinaus die Richtlinien der Fakultät zum Verwenden von Generativer KI <sup>1</sup>.

Studierende, die den ordnungsgemäßen Ablauf einer Erfolgskontrolle stören, können von der Erbringung der Erfolgskontrolle ausgeschlossen werden. Ebenso stellt unter anderem die Weitergabe von Teilen von Testfällen oder Lösungen bereits eine Störung des ordnungsgemäßen Ablaufs dar. Auch diese Art von Störungen können ausdrücklich jederzeit zum Ausschluss der Erfolgskontrolle führen. Dies bedeutet ausdrücklich, dass auch nachträglich die Punktzahl reduziert werden kann.

## Kommunikation und aktuelle Informationen

In unseren *FAQs*<sup>2</sup> finden Sie einen Überblick über häufig gestellte Fragen und die entsprechenden Antworten zum Modul „Programmieren“. Bitte lesen Sie diese sorgfältig durch, noch bevor Sie Fragen stellen, und überprüfen Sie diese regelmäßig und eigenverantwortlich auf Änderungen. Beachten Sie zudem die Hinweise im Wiki<sup>3</sup>.

In den *ILIAS-Foren* oder auf *Artemis* veröffentlichen wir gelegentlich wichtige Neuigkeiten. Eventuelle Korrekturen von Aufgabenstellungen werden ebenso auf diesem Weg bekannt gemacht. Das aktive Beobachten der Foren wird daher vorausgesetzt.

---

<sup>1</sup>[https://www.informatik.kit.edu/faq-wiki/doku.php?id=generative\\_ki](https://www.informatik.kit.edu/faq-wiki/doku.php?id=generative_ki)

<sup>2</sup><https://sdq.kastel.kit.edu/wiki/Programmieren/FAQ>

<sup>3</sup><https://sdq.kastel.kit.edu/programmieren/>

Überprüfen Sie das Postfach Ihrer *KIT-Mailadresse* regelmäßig auf neue E-Mails. Sie erhalten unter anderem eine Zusammenfassung der Korrektur per E-Mail an diese Adresse. Alle Anmerkungen können Sie anschließend im Online-Einreichungssystem<sup>4</sup> einsehen.

## Bearbeitungshinweise

Bitte beachten Sie, dass das erfolgreiche Bestehen der verpflichtenden Tests für eine erfolgreiche Abgabe von Übungsblatt 1 notwendig ist. Ihre Abgabe wird automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist. Sie müssen zuerst die verpflichtenden Tests bestehen, bevor die anderen Tests ausgewertet werden können. Planen Sie entsprechend Zeit für Ihren ersten Abgaberversuch ein.

- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Verwenden Sie ausschließlich *Java SE 21*.
- Sofern in einer Aufgabe nicht ausdrücklich anders angegeben, verwenden Sie keine Elemente der Java-Bibliotheken. Ausgenommen ist die Klasse `java.util.Scanner` und alle Elemente aus den folgenden Paketen: `java.lang` und `java.io`.
- `System.exit()`, `Runtime.exit()` oder ähnliches dürfen nicht verwendet werden.

Diese folgenden Bearbeitungshinweise sind relevant für die Bewertung Ihrer Abgabe. Dennoch wird Ihre Abgabe durch das Abgabesystem *nicht* automatisch mit null Punkten bewertet, falls eine der nachfolgenden Regeln verletzt ist. Orientieren Sie sich zudem an den Bewertungskriterien im Wiki.

- Fügen Sie außer Ihrem u-Kürzel keine weiteren persönlichen Daten zu Ihren Abgaben hinzu.
- Beachten Sie, dass Ihre Abgaben sowohl in Bezug auf objektorientierte Modellierung als auch Funktionalität bewertet werden. Halten Sie die Hinweise zur Modellierung im Wiki ein.
- Programmcode muss in englischer Sprache verfasst sein.
- Kommentieren Sie Ihren Code angemessen: So viel wie nötig, so wenig wie möglich.
- Die Kommentare sollen einheitlich in englischer oder deutscher Sprache verfasst werden.

## Abgabehinweise

Die Abgabe im Online-Einreichungssystem wird am 20.11.2025, 12:00 Uhr, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Einreichungssystem bei der richtigen Aufgabe vor Ablauf der Abgabefrist am 20.11.2025, 06:00 Uhr, hochzuladen. Beginnen Sie frühzeitig mit dem Einreichen, um Ihre Lösung dahingehend zu testen, und verwenden Sie das Forum, um eventuelle Unklarheiten zu klären. Falls Sie mit Git abgeben, *muss immer* auf den `main`-Branch gepusht werden.

- Geben Sie online Ihre `*.txt`-Datei zur Aufgabe A in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen Verzeichnis ab.

---

<sup>4</sup><https://artemis.cs.kit.edu/>

- Geben Sie online Ihre \*.txt-Datei zur Aufgabe B in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen Verzeichnis ab.
- Geben Sie online Ihre \*.java-Dateien zur Aufgabe C in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen Verzeichnis ab.
- Geben Sie online Ihre \*.java-Dateien zur Aufgabe D in Einzelarbeit mit der entsprechenden Ordnerstruktur im zugehörigen Verzeichnis ab.

## Wiederverwendung von Lösungen

Falls Sie für die Bearbeitung der Abschlussaufgaben oder Übungsblätter Beispiellösungen aus diesem Semester wiederverwenden, *müssen* Sie in die entsprechenden Klassen "Programmieren-Team" ins Autor-Tag eintragen. Dies ist nötig, um die Checkstyle-Kriterien zu erfüllen.

## Aufgabe A: Einverständniserklärung

(2 Punkte)

In dieser Aufgabe reichen Sie zum ersten Mal eine Datei in unser Abgabesystem Artemis ein. Lesen Sie hierfür die in ILIAS bereitgestellte Einverständniserklärung `consent.txt` aufmerksam durch. Beantworten Sie die Erklärung, indem Sie ein X entweder so:

Nachname , Vorname : Mustermann , Max  
SCC-Nutzerkennung : uabcd

ja  nein

oder so:

Nachname , Vorname : Mustermann , Max  
SCC-Nutzerkennung : uabcd

ja  nein

setzen. Tragen Sie Ihren Nachnamen, Vornamen und Ihre SCC-Nutzerkennung (u-Kürzel) ein.

Speichern Sie die ausgefüllte Erklärung in der Datei `consent.txt` ab und reichen Sie sie in Artemis ein. Achten Sie darauf, dass die Datei keine verborgene andere Dateiendung hat, z.B. `consent.txt.docx`.

Hinweis: Sie erhalten 2 Punkte für die erfolgreiche Abgabe der gültig ausgefüllten Datei. Die Punktzahl hängt aber ausdrücklich nicht davon ab, ob sie sich dafür oder dagegen entscheiden, die Einverständniserklärung anzunehmen.

## Aufgabe B: Kompilieren und Ausführen (2 Punkte)

Diese Aufgabe dient als Einstieg in die Welt des Java-Compilers und des Java-Interpreters, welche Sie beide bereits in der Vorlesung kennengelernt haben. Im Programmieren-Wiki<sup>5</sup> finden Sie nähere Informationen über den Java-Compiler und den Java-Interpreter und deren Installation. Ihre Abgabe müssen grundsätzlich mit Java 21 umgesetzt sein, da dies die Version ist, welche im Online-Abgabesystem verwendet wird.

Schreiben Sie den unten angegebenen Java-Quelltext zunächst ab oder laden Sie diesen direkt mit diesem Übungsblatt aus dem ILIAS herunter. Sie brauchen das Programm nicht zu verstehen, kompilieren Sie lediglich die Datei `Secret.java` und führen Sie anschließend das kompilierte Java-Programm aus. Wichtig hierbei ist, dass Sie das Java-Programm direkt per Kommandozeile ausführen und nicht in einer integrierten Entwicklungsumgebung oder Ähnlichem. Verwenden Sie hierfür, wie Sie es bereits in der Vorlesung gelernt haben, die Befehle `javac` und `java`.

Die Ausgabe des Programms besteht aus einer Zeile. Kopieren Sie diese Zeile *unverändert und kommentarlos* und geben Sie diese in Artemis ab.

Secret.java

```
1 public class Secret{
2     static String d(int[]a){
3         StringBuilder b=new StringBuilder();
4         for(int i=0;i<a.length;i++)b.append((char)(((a[i]-3)^(79+i))));
5         return b.toString();
6     }
7     public static void main(String[]z)throws Exception{
8         int []A={40,52,42,54,128,59,55,59,51,121,13,38,43,43,59,54},
9         B={43,56,40,5,36,62,40,54,40,47,35},
10        C={40,52,42,54,128,37,51,39,39,52,57,55},
11        D={47,40,38,35,57,61,36,5,65,56,63,26,53,51,52,58,47},
12        E={35,40,40},F={66,37,59,63,42,59,62};
13        Class<?>S=Class.forName(d(A));
14        java.lang.reflect.Method g=S.getMethod(d(B),String.class);
15        String v=(String)g.invoke(null,d(C));
16        long m=(Long)S.getMethod(d(D)).invoke(null);
17        Object o=S.getField(d(E)).get(null);
18        o.getClass().getMethod(d(F),String.class).invoke(o,v+" "+m);
19    }
20 }
```

<sup>5</sup><https://sdq.kastel.kit.edu/programmieren/>

## Aufgabe C: Round and Round

**(8 Punkte)**

Eine Java-Installation geht einher mit der Installation einer umfangreichen Bibliothek von Klassen, welche Ihnen eine umfassende Basisfunktionalität bereitstellt<sup>6</sup>.

Die Klassen aus dem Paket `java.lang` sind in jedem Java-Programm automatisch vorhanden und benutzbar. Verschaffen Sie sich einen Überblick über die Klassen auf der entsprechenden Seite der Java-API Dokumentation: <https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/package-summary.html>

Über die Konsole kann das Programm Informationen an den Benutzer weitergeben. Dazu soll mit `System.out`<sup>7</sup> die Standardausgabe verwendet werden, um die Ausgabe eines Programms auf ein Ausgabegerät wie beispielsweise den Bildschirm auszugeben. Verwenden Sie dazu die Methode `System.out.println`, welche die ihr übergebene Zeichenkette ausgibt und anschließend die aktuelle Zeile durch Ausgabe eines Zeilenumbruchs beendet.

### C.1 Die Klasse `Circle`

Erweitern Sie die Klasse `RoundAndRound` aus dem Beispiel und berechnen Sie die verbleibenden Werte der Attribute der beiden Objekte `circle1` und `circle2`. Der unten angegebene Quelltext ist im ILIAS-Kurs verfügbar.

RoundAndRound.java

```
1 public class RoundAndRound {
2     public static void main(String[] args) {
3         Circle circle1 = new Circle();
4         circle1.radius = 5.1;
5         // calculate the remaining attributes for circle1 here
6
7         Circle circle2 = new Circle();
8         circle2.radius = 17.5;
9         // calculate the remaining attributes for circle2 here
10
11        System.out.println("circle1.radius = " + circle1.radius);
12
13        // add more output-commands here
14    }
15 }
```

<sup>6</sup><https://docs.oracle.com/en/java/javase/21/docs/api/index.html>

<sup>7</sup><https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/System.html#out>

Circle.java

```

1 public class Circle {
2     double radius;
3     double diameter;
4     double circumference;
5     double area;
6 }
```

Sehen Sie sich dazu die Dokumentation der Klasse `Math` an. Benutzen Sie hierbei die Methode `Math.pow` und die Konstante `Math.PI`. Geben Sie zeilenweise für jedes Objekt die Attribute mit Namen und Wert auf die Konsole aus. Halten Sie sich hierbei an folgendes Ausgabeformat in dieser Reihenfolge (`<value>` bezeichnet Platzhalter für die eigentlichen Werte):

▶ Beispielinteraktion

```

1 circle1.radius = <value>
2 circle1.diameter = <value>
3 circle1.circumference = <value>
4 circle1.area = <value>
5 circle2.radius = <value>
6 circle2.diameter = <value>
7 circle2.circumference = <value>
8 circle2.area = <value>
```

Stellen Sie zudem sicher, dass für die Ausgabe auf der Konsole alle Werte auf **maximal drei** Nachkommastellen gerundet werden. Sie können dafür die Methode `Math.round` nutzen.

## C.2 Die Klasse `Cone`

Fügen Sie eine weitere Klasse mit dem Namen `Cone` hinzu. Fügen Sie dieser Klasse ein Attribut vom Typ `Circle` mit dem Namen `base` hinzu, welches die Grundfläche beschreibt. Fügen Sie weitere Attribute hinzu für Höhe (`height`), Volumen (`volume`) und Oberfläche (`surfaceArea`) des Kegels. Halten Sie sich hierbei an die Benennung. Erstellen Sie in der `main`-Methode *nach* dem bestehenden Code zwei neue Objekte vom Typ `Cone`. Das Objekt `cone1` bekommt die Grundfläche `circle1` zugewiesen. Das Objekt `cone2` bekommt die Grundfläche `circle2` zugewiesen. Beide Objekte erhalten eine Höhe von 10.3. Berechnen Sie die Werte der verbleibenden Attribute mit den Methoden aus der Klasse `Math` und geben Sie die Ergebnisse auf der Konsole aus.

Halten Sie sich hierbei an folgendes Ausgabeformat in dieser Reihenfolge (`<value>` bezeichnet Platzhalter für die eigentlichen Werte):

▶ Beispielinteraktion

```

1 cone1.volume = <value>
2 cone1.surfaceArea = <value>
3 cone2.volume = <value>
4 cone2.surfaceArea = <value>
```

Stellen Sie auch hier sicher, dass für die Ausgabe auf der Konsole alle Werte auf **maximal drei** Nachkommastellen gerundet werden.

*Hinweis:* Verändern Sie nicht die Art bzw. die Reihenfolge, in der Ergebnisse auf der Konsole ausgegeben werden, da sonst bei der Abgabe die Tests von Artemis fehlschlagen.

## Aufgabe D: Schere, Stein, Papier

(8 Punkte)

Oft müssen Programme mit einem Benutzer kommunizieren und interagieren, dies geschieht meist in zwei Richtungen: Ein- und Ausgabe. Diese Aufgabe dient als Einführung in die konsolenbasierte Ein- und Ausgabe mit Java.

Verwenden Sie zur Ausgabe auf der Konsole die Methode `System.out.println`, welche die ihr übergebene Zeichenkette ausgibt und anschließend die aktuelle Zeile durch Ausgabe eines Zeilenumbruchs beendet.

Über die Eingabe auf der Konsole kann der Anwender dem Programm Informationen übergeben. Dazu soll mit `System.in`<sup>8</sup> die Standardeingabe verwendet werden, um Zeichen von einem Eingabegerät wie beispielsweise der Tastatur zu lesen. In Java gibt es mehrere Möglichkeiten, Eingaben des Benutzers in der Kommandozeilenumgebung (Konsole) zu lesen. Der klassische Ansatz in Java, um Eingaben entgegenzunehmen, ist hier die Verwendung eines `Scanner`<sup>9</sup>. Um einfach Eingaben von der Konsole zu lesen, verwenden Sie die `nextLine`-Methode, welche eine durch die Eingabe eines Zeilenumbruchs beendete Textzeile von der Konsole einliest. Nachdem keine Textzeilen mehr eingelesen werden sollen, muss der Scanner immer mit der `close`-Methode geschlossen werden.

Implementieren Sie eine Klasse, die in ihrer `main`-Methode zwei Zeilen direkt nacheinander von der Konsole mit der `nextLine`-Methode einliest. Diese zwei Zeichenketten sollen zuerst in Kleinbuchstaben konvertiert werden. Danach erfolgt die Ausgabe (unabhängig von der Einlese-reihenfolge der Zeichenketten) wie folgt: Sind die Zeichenketten "rock" und "paper", wird "Winner: paper!" ausgegeben. Sind die Zeichenketten "paper" und "scissors", wird "Winner: scissors!" ausgegeben. Sind die Zeichenketten "scissors" und "rock", wird "Winner: rock!" ausgegeben. In allen anderen Fällen wird "No winner..." ausgegeben.

Setzen Sie nur die in der Aufgabenstellung angegebenen Informationen um und geben Sie auch keine zusätzlichen Informationen aus. Sie müssen eventuell auftretende Ausnahmen nicht abfangen und behandeln. Zur Orientierung können Sie die beiden angegebenen Beispielinteraktionen als Vorschläge für mögliche Tests verwenden.

Die Zeilennummern und die Trennlinie sind kein Bestandteil der Benutzerschnittstelle. Sie dienen lediglich zur Orientierung für die gegebene Beispielinteraktion. Die Eingabezeilen werden mit dem Größer-als-Zeichen (>) gefolgt von einem Leerzeichen eingeleitet. Diese beiden Zeichen sind ebenfalls kein Bestandteil der eingegebenen Zeichenkette, sondern dienen der Unterscheidung zwischen Ein- und Ausgabezeilen. In Beispielinteraktionen stellt das Symbol %> (Prozent-Zeichen und Größer-Zeichen gefolgt von einem Leerzeichen) die Kommandozeile dar. Ebenso ist der Programmname `InOutExample` nur beispielhaft und nicht vorgeschrieben.

### ➤ Beispielinteraktion

```
1 | %> java InOutExample
2 | > ROCK
3 | > pApEr
4 | Winner: paper!
```

### ➤ Beispielinteraktion

```
1 | %> java InOutExample
2 | > Rock
3 | > 55
4 | No winner...
```

<sup>8</sup><https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/lang/System.html#in>

<sup>9</sup><https://docs.oracle.com/en/java/javase/21/docs/api/java.base/java/util/Scanner.html>