
Programmieren – Wintersemester 2024/25

Übungsblatt 1

Version 1.0

20 Punkte

Ausgabe: 30.10.2024, ca. 12:00 Uhr

Abgabestart: 06.11.2024, 12:00 Uhr

Abgabefrist: 14.11.2024, 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 ¹.

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*² 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 Wiki³.

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.

¹https://www.informatik.kit.edu/faq-wiki/doku.php?id=generative_ki

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

³<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-Einreichungssystem⁴ 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 Abgabeversuch ein.

- Achten Sie auf fehlerfrei kompilierenden Programmcode.
- Verwenden Sie ausschließlich *Java SE 17*.
- 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 ILIAS-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 ILIAS-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 06.11.2024, 12:00 Uhr, freigeschaltet. Achten Sie unbedingt darauf, Ihre Dateien im Einreichungssystem bei der richtigen Aufgabe vor Ablauf der Abgabefrist am 14.11.2024, 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.

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

- Geben Sie online Ihre *.java-Dateien 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.

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: Kompilieren und Ausführen (4 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-Wiki⁵ finden Sie nähere Informationen über den Java-Compiler und den Java-Interpreter und deren Installation. Ihre Abgabe müssen grundsätzlich mit Java 17 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 import java.util.function.Function;
2
3 public class Secret {
4     public static void main(final String[] args) {
5         System.console().format("%s%n", String.valueOf(new Object() {
6             public String toString() {
7                 var q = java.util.List.of("6f732e6e616d65",
8                     "66696c652e736570617261746f72", "6f732e61726368");
9                 Function<String, byte[]> one =
10                    java.util.HexFormat.of()::parseHex;
11                 Function<byte[], String> two = String::new;
12                 Function<String, String> three = System::getProperty;
13                 String[] a = q.stream().map(one.andThen(two.andThen(three)))
14                    .toArray(String[]::new);
15                 return String.format("%s\u0020%s\u0020%s", a[0], a[1], a[2]);
16             }
17         })));
18     }
19 }
```

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

Aufgabe B: Ein- und Ausgabe

(8 Punkte)

Oft müssen Programme mit Menschen 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`⁶ 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`⁷. 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 drei Zeilen direkt nacheinander von der Konsole mit der `nextLine`-Methode einliest. Diese drei Zeichenketten sollen zuerst in Kleinbuchstaben konvertiert werden und dann jeweils mit einem zusätzlichen Trennzeichen konkateniert werden. Anschließend soll das Ergebnis auf einer einzigen neuen Zeile auf der Konsole ausgegeben werden. Das Trennzeichen wird als Kommandozeilenargument beim Programmstart mit übergeben. Sie können in Ihrer Implementierung die Annahme treffen, dass beim Programmstart immer ein (einzelnes) Trennzeichen als Kommandozeilenargument übergeben wird. Auf das Trennzeichen können Sie über das Parameter `args` der `main`-Methode wie folgt zugreifen:

```
String separator = args[0];
```

Beachten Sie, dass das Trennzeichen nicht in Kleinbuchstaben konvertiert werden soll.

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. Ebenso ist der Programmname `InOutExample` nur beispielhaft.

▶ Beispielinteraktion

```
1 | > java InOutExample x
2 | > ABC
3 | > def
4 | > 1 2 3
5 | abcdefx1 2 3
```

▶ Beispielinteraktion

```
1 | > java InOutExample -
2 | > Ein
3 | > weiteres
4 | > Beispiel.
5 | ein-weiteres-beispiel.
```

⁶<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/System.html#in>

⁷<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html>

Aufgabe C: Modellierung einer Tagung

(8 Punkte)

In dieser Aufgabe sollen die grundlegenden Klassen für eine (vereinfachte) Tagung modelliert werden. Das primäre Ziel ist es, sich mit dem Entwurf und Erstellen von Klassen in Java vertraut zu machen. Darüber hinaus sollen Sie das Hinzufügen von Attributen zu Klassen üben und die Auswahl geeigneter Typen für diese Attribute erlernen. Es geht hierbei um die reine objektorientierte Modellierung der Klassen und ihrer Attribute. Das heißt, Ihre Klassen beinhalten noch keine Konstruktoren oder Methoden. Achten Sie auf den Einsatz von sinnvollen Datentypen und wählen Sie aussagekräftige englischsprachige Bezeichner. Fehlerfrei kompilierender Programmcode bildet die Mindestanforderung für eine gültige Abgabe.

Eine *Tagung* wird durch einen Namen und eine Adresse beschrieben. Eine *Adresse* besteht aus einem Straßennamen und einer Stadt, beides Zeichenketten, sowie einer Postleitzahl, die als Ganzzahl dargestellt wird. Die Tagung hat zwei feste *Teilnehmende*: eine *Leiterin* und einen *Protokollant*. Jeder *Teilnehmende* hat einen Namen, eine eindeutige ID und eine akademische Rolle. Die akademische Rolle gibt an, ob die Person ein *Doktorand*, *Post-Doc* oder *Professorin* ist (fester Wertebereich). Es gibt genau zwei Veranstaltungen in der Tagung: einen *Vortrag* und eine *Breakout-Session*. Beide Arten von Veranstaltungen haben einen Titel, eine festgelegte Dauer in Minuten und einen Veranstaltungsort (Adresse). Ein *Vortrag* wird von einem *Teilnehmenden* gehalten und hat ein *Thema* (Zeichenkette). Eine *Breakout-Session* wird von einem *Teilnehmenden* geleitet und hat eine maximale Anzahl von *Teilnehmenden*.