

## Allgemeine Hinweise:

- Die **Deadline** zur **Abgabe** der Hausaufgaben ist am **Dienstag, den 29.10.2024, um 12 Uhr**.
- Der **Workflow** sieht wie folgt aus. Die Abgabe der Hausaufgaben erfolgt **im Moodle-Lernraum** und kann nur in **Zweiergruppen** stattfinden. Dabei müssen die Abgabepartner\*innen **dasselbe Tutorium** besuchen. Nutzen Sie ggf. das entsprechende **Forum** im Moodle-Lernraum, um eine\*n Abgabepartner\*in zu finden. Es darf **nur ein\*e** Abgabepartner\*in die Abgabe hochladen. Diese\*r muss sowohl die **Lösung** als auch den **Quellcode** der Programmieraufgaben hochladen. Die Bepunktung wird dann von uns für **beide** Abgabepartner\*innen **separat** im Lernraum eingetragen. Die Feedbackdatei ist jedoch nur dort sichtbar, wo die Abgabe hochgeladen wurde und muss innerhalb des Abgabepaars **weitergeleitet** werden.
- Die **Lösung** muss als PDF-Datei hochgeladen werden. Damit die Punkte beiden Abgabepartner\*innen zugeordnet werden können, müssen **oben** auf der **ersten Seite** Ihrer Lösung die **Namen**, die **Matrikelnummern** sowie die **Nummer des Tutoriums** von **beiden** Abgabepartner\*innen angegeben sein.
- Der **Quellcode** der Programmieraufgaben muss als **.zip**-Datei hochgeladen werden und **zusätzlich** in der PDF-Datei mit Ihrer Lösung enthalten sein, sodass unsere Hiwis ihn mit Feedback versehen können. Auf diesem Blatt muss Ihre Codeabgabe Ihren vollständigen **Java-Code** in Form von **.java**-Dateien enthalten. Aus dem Lernraum heruntergeladene Klassen, etwa die Datei **SimpleIO.java**, dürfen nicht mit abgegeben werden.  
Stellen Sie sicher, dass Ihr Programm von **javac** **akzeptiert** wird, wenn die entsprechenden Klassen aus dem Lernraum hinzugefügt werden. Ansonsten werden keine Punkte vergeben.
- Einige Hausaufgaben müssen im Spiel **Codescape** gelöst werden. Klicken Sie dazu im Lernraum rechts im Block **“Codescape”** auf den angegebenen Link. Diese Aufgaben werden getrennt von den anderen Hausaufgaben gewertet.

### Aufgabe 3 (Programmierung):

(20 Punkte)

Implementieren Sie ein Programm, welches zu einem gegebenen Startdatum und einer gegebenen Anzahl  $t$  an Tagen ein Enddatum berechnet, sodass das Enddatum genau  $t$  Tage nach dem Startdatum liegt. Beispielsweise liegt der 30.10.2024 genau einen Tag nach dem 29.10.2024. Die Berechnung soll mithilfe *einer einzigen geeigneten Schleife* durchgeführt werden. Das Programm soll zudem weder `break` noch `continue` benutzen. Verwenden Sie erneut die Klasse `SimpleIO` zum Einlesen und Ausgeben von Werten.

Das Programm fragt zunächst nach dem Startdatum. Hierzu wird der\*die Benutzer\*in nacheinander aufgefordert, den entsprechenden Tag des Monats, die Nummer des Monats und das Jahr einzugeben. Anschließend fragt das Programm nach der Anzahl  $t$  an Tagen. Alle Werte sollen als `int` eingelesen werden. In allen Berechnungen kann die Existenz von Schaltjahren vernachlässigt werden. Falls das eingegebene Startdatum nicht existiert oder  $t$  keine positive Zahl ist, so darf sich das Programm beliebig verhalten.

Ein Beispiellauf des Programms könnte also so aussehen:

```
Bitte geben Sie die Tageskomponente des Startdatums ein.
27
Bitte geben Sie die Monatskomponente des Startdatums ein.
2
Bitte geben Sie die Jahreskomponente des Startdatums ein.
2024
Bitte geben Sie die Anzahl an Tagen ein:
365
Das Datum 27.2.2025 befindet sich 365 Tage nach dem Startdatum.
```

#### Hinweise:

- Für nicht-negative ganze Zahlen berechnet der Java-Operator `%` die modulo-Funktion. D.h. für natürliche Zahlen  $n$  und  $d$  ist  $n\%d$  gerade der Rest der Division von  $n$  und  $d$ .

## Aufgabe 5 (Verifikation):

(14 Punkte)

Gegeben sei folgendes Java-Programm über den Integer-Variablen `a`, `i`, `n` und `res`:

```

(0 ≤ n)                                (Vorbedingung)
    res = a;
    i = n;
    while (i > 0) {
        i = i - 1;
        res = res * res;
    }
⟨res = a2n⟩                            (Nachbedingung)
  
```

Vervollständigen Sie die folgende Verifikation der partiellen Korrektheit des Algorithmus im Hoare-Kalkül, indem Sie die unterstrichenen Teile ergänzen. Hierbei dürfen zwei Zusicherungen nur dann direkt untereinander stehen, wenn die untere aus der oberen folgt. Hinter einer Programmanweisung darf nur eine Zusicherung stehen, wenn dies aus einer Regel des Hoare-Kalküls folgt.

### Hinweise:

- Der Ausdruck  $a^{2^n}$  steht für  $a^{(2^n)}$ .
- Sie dürfen beliebig viele Zusicherungs-Zeilen ergänzen oder streichen. In der Musterlösung werden allerdings genau die angegebenen Zusicherungen benutzt.
- Bedenken Sie, dass die Regeln des Kalküls syntaktisch sind, weshalb Sie semantische Änderungen (beispielsweise von  $x + 1 = y + 1$  zu  $x = y$ ) nur unter Zuhilfenahme der Konsequenzregeln vornehmen dürfen.
- Es empfiehlt sich oft, bei der Erstellung der Zusicherungen in der Schleife von unten (d.h. von der Nachbedingung aus) vorzugehen.
- Geben Sie jeweils eine kurze Begründung an, warum die Konsequenzregeln korrekt angewandt wurden. D.h. beweisen Sie, dass aus der oberen Zusicherung die untere folgt, wenn diese direkt untereinander stehen.

	$\langle 0 \leq n \rangle$
<code>res = a;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>i = n;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>while (i &gt; 0) {</code>	$\langle \underline{\hspace{15cm}} \rangle$
	$\langle \underline{\hspace{15cm}} \rangle$
<code>    i = i - 1;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>    res = res * res;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>}</code>	$\langle \underline{\hspace{15cm}} \rangle$
	$\langle \underline{\hspace{15cm}} \rangle$
	$\langle \text{res} = a^{2^n} \rangle$

## Aufgabe 7 (Verifikation):

(16 Punkte)

Gegeben sei folgendes Java-Programm  $P$  über den Integer-Variablen  $x$ ,  $y$ ,  $i$  und  $j$ , welches den größten gemeinsamen Teiler (ggT) von  $x$  und  $y$  berechnet. Im folgenden sei  $\text{ggT}(a, b)$  der größte gemeinsame Teiler der beiden natürlichen Zahlen  $a, b \geq 1$ . Zum Beispiel gilt  $\text{ggT}(12, 9) = 3$  und  $\text{ggT}(3, 3) = 3$ .

```

( $x \geq 1 \wedge y \geq 1$ )          (Vorbedingung)
  i = x;
  j = y;
  while (i != j) {
    if (i > j) {
      i = i - j;
    } else {
      j = j - i;
    }
  }
( $i = \text{ggT}(x, y) = j$ )      (Nachbedingung)

```

Vervollständigen Sie die folgende Verifikation des Algorithmus im Hoare-Kalkül, indem Sie die unterstrichenen Teile ergänzen. Hierbei dürfen zwei Zusicherungen nur dann direkt untereinander stehen, wenn die untere aus der oberen folgt. Hinter einer Programmanweisung darf nur eine Zusicherung stehen, wenn dies aus einer Regel des Hoare-Kalküls folgt.

### Hinweise:

- Sie dürfen beliebig viele Zusicherungs-Zeilen ergänzen oder streichen. In der Musterlösung werden allerdings genau die angegebenen Zusicherungen benutzt.
- Bedenken Sie, dass die Regeln des Kalküls syntaktisch sind, weshalb Sie semantische Änderungen (beispielsweise von  $x + 1 = y + 1$  zu  $x = y$ ) nur unter Zuhilfenahme der Konsequenzregeln vornehmen dürfen.
- Sie können verwenden, dass für alle Zahlen  $a > b \geq 1$  gilt:  $\text{ggT}(a, b) = \text{ggT}(a - b, b)$  und  $\text{ggT}(b, a) = \text{ggT}(b, a - b)$ .
- Geben Sie jeweils eine kurze Begründung an, warum die Konsequenzregeln korrekt angewandt wurden. D.h. beweisen Sie, dass aus der oberen Zusicherung die untere folgt, wenn diese direkt untereinander stehen.

	$\langle x \geq 1 \wedge y \geq 1 \rangle$
<code>i = x;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>j = y;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>while (i != j) {</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>if (i &gt; j) {</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>i = i - j;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>} else {</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>j = j - i;</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>}</code>	$\langle \underline{\hspace{15cm}} \rangle$
<code>}</code>	$\langle \underline{\hspace{15cm}} \rangle$
	$\langle \underline{\hspace{15cm}} \rangle$
	$\langle i = \text{ggT}(x, y) = j \rangle$

### Aufgabe 8 (Deck 2):

(Codescape)

Lösen Sie die Missionen von Deck 2 des Codescape Spiels. Ihre Lösung für die Codescape Missionen wird nur dann für die Zulassung gezählt, wenn sie Ihre Lösung vor der einheitlichen Codescape Deadline am Freitag, den 24.01.2025, um 23:59 Uhr abschicken.