Prof. Dr. J. Giesl
J.-C. Kassing, M. Leven Rosarius, N. Lommen, E. Meyer

Allgemeine Hinweise:

- Die **Deadline** zur **Abgabe** der Hausaufgaben ist am **Donnerstag, den 06.11.2025, um 14 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 dürfen nicht mit abgegeben werden.

  Stellen Sie sicher, dass Ihr Programm von `javac` **in der Version 25 akzeptiert** wird. Generell sollten alle Programme für alle Eingaben terminieren, solange in der Spezifikation (bzw. der Aufgabenstellung) nicht explizit etwas anderes verlangt wird!
- 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.

Hausaufgabe 3 (Verifikation): (12 + 4 Punkte)

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

$\langle 0 \leq \texttt{n} \rangle$ (Vorbedingung)

```
   res = 1;
   i = n;
   while (i > 0) {
      i = i - 1;
      res = 42 * res;
   }
```

$\langle \texttt{res} = 42^{\texttt{n}} \rangle$ (Nachbedingung)

a) 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:

- 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.

$\langle 0 \leq \texttt{n} \rangle$

$\langle \_\_\_\_\_ \rangle$

```
res = 1;
```

$\langle \_\_\_\_\_ \rangle$

```
i = n;
```

$\langle \_\_\_\_\_ \rangle$

```
while (i > 0) {
```

$\langle \_\_\_\_\_ \rangle$

$\langle \_\_\_\_\_ \rangle$

```
    i = i - 1;
```

$\langle \_\_\_\_\_ \rangle$

```
    res = 42 * res;
```

$\langle \_\_\_\_\_ \rangle$

```
}
```

$\langle \_\_\_\_\_ \rangle$

$\langle \texttt{res} = 42^{\texttt{n}} \rangle$

b) Untersuchen Sie den Algorithmus P auf seine Terminierung. Für einen Beweis der Terminierung muss eine Variante angegeben werden und mit Hilfe des Hoare-Kalküls die Terminierung bewiesen werden.

Hausaufgabe 5 (Verifikation): (17 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 $\mathrm{ggT}(a, b)$ der größte gemeinsame Teiler der beiden natürlichen Zahlen $a, b \geq 1$. Zum Beispiel gilt $\mathrm{ggT}(12, 9) = 3$ und $\mathrm{ggT}(3, 3) = 3$.

$\langle \texttt{x} \geq 1 \wedge \texttt{y} \geq 1 \rangle$ (Vorbedingung)

```
i = x;
j = y;
while (i != j) {
   if (i > j) {
      i = i - j;
   } else {
      j = j - i;
   }
}
```

$\langle \texttt{i} = \mathrm{ggT}(\texttt{x}, \texttt{y}) = \texttt{j} \rangle$ (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 können verwenden, dass für alle Zahlen $a > b \geq 1$ gilt: $\mathrm{ggT}(a, b) = \mathrm{ggT}(a - b, b)$ und $\mathrm{ggT}(b, a) = \mathrm{ggT}(b, a - b)$.
- 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.
- 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.

$\langle \mathtt{x} \geq 1 \wedge \mathtt{y} \geq 1 \rangle$

$\langle \_\_\_\_\_ \rangle$

```
i = x;
```

$\langle \_\_\_\_\_ \rangle$

```
j = y;
```

$\langle \_\_\_\_\_ \rangle$

```
while (i != j) {
```

$\langle \_\_\_\_\_ \rangle$

```
    if (i > j) {
```

$\langle \_\_\_\_\_ \rangle$

$\langle \_\_\_\_\_ \rangle$

```
        i = i - j;
```

$\langle \_\_\_\_\_ \rangle$

```
    } else {
```

$\langle \_\_\_\_\_ \rangle$

$\langle \_\_\_\_\_ \rangle$

```
        j = j - i;
```

$\langle \_\_\_\_\_ \rangle$

```
    }
```

$\langle \_\_\_\_\_ \rangle$

```
}
```

$\langle \_\_\_\_\_ \rangle$

$\langle \mathtt{i} = \mathrm{ggT}(\mathtt{x}, \mathtt{y}) = \mathtt{j} \rangle$

Hausaufgabe 7 (Programmierung): (17 Punkte)

Bubblesort ist ein Algorithmus zum Sortieren von Arrays, der wie folgt vorgeht, um ein Array `a` zu sortieren: Das Array wird wiederholt von links nach rechts durchlaufen. Am Ende des n-ten Durchlauf gilt, dass die letzten n Array-Elemente an ihrer endgültigen Position stehen. Folglich müssen im $(n+1)$-ten Durchlauf nur noch die ersten `a.length`$-n$ Elemente betrachtet werden. In jedem Durchlauf wird in jedem Schritt das aktuelle Element mit seinem rechten Nachbarn verglichen. Falls das aktuelle Element größer ist als sein rechter Nachbar, werden sie getauscht.
Als Beispiel betrachten wir das Array `{3,2,1}`. Im ersten Durchlauf wird erst `3` mit `2` getauscht (dies ergibt `{2,3,1}`) und dann `3` mit `1`, was `{2,1,3}` ergibt. Im zweiten Durchlauf wird `2` mit `1` getauscht, was zu `{1,2,3}` führt.
Hierzu soll die `main`-Methode Ihrer Implementierung ein Array `String[] args` von `String`s übergeben bekommen. Verwenden Sie `Integer.parseInt`, um einen `String` in einen `int` zu konvertieren. Geben Sie außerdem das sortierte Array am Ende mithilfe der aus der Vorlesung bekannten Klasse `IO` aus. Ein Ablauf des Programms in der Datei `Bubblesort.java` könnte z.B. so aussehen:

```
> java Bubblesort 5 -3 1 0 5
-3 0 1 5 5
```

Hausaufgabe 8 (Deck 3): (Codescape)

Lösen Sie die Missionen von Deck 3 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 30.01.2026, um 23:59 Uhr abschicken.