

Präsenzübung – Hinweise und Beispielaufgaben

Allgemeine Hinweise

Am **17.01.2019** findet die Präsenzübung statt. Informationen zur Hörsaaleinteilung und zum Zeitplan werden rechtzeitig im ILIAS-Kurs *Programmieren WS 2018/2019*¹ und in der Vorlesung bekanntgegeben. Im Folgenden finden Sie wichtige Hinweise und Aufgaben, die beispielhaft für die Aufgaben in der Präsenzübung sein können.

Hinweis: Die Aufgaben werden in der Präsenzübung nur auf Deutsch gestellt. Falls Sie ein Wörterbuch benötigen, geben Sie dieses **bis spätestens 10.01.2019** im Sekretariat von Prof. Reussner (Informatik-Hauptgebäude Geb. 50.34, Raum 328, Abgabemöglichkeit: Mo-Fr 09:00-15:00 Uhr) zur Überprüfung ab. **Nicht durch uns geprüfte Wörterbücher können nicht verwendet werden.** Sie erhalten das Wörterbuch in der Präsenzübung von uns zurück. Falls Sie nicht an der Präsenzübung teilnehmen, können Sie es wieder im Sekretariat abholen.

All questions in the written test („Präsenzübung“) will be in German. If you need a dictionary, hand it in for checking at the secretary's office of Professor Reussner (building 50.34, room 328. Opening hours: Monday to Friday, 9:00 to 15:00) **at the latest on 10.01.2019. If we have not checked the dictionary, you will not be allowed to use it during the test.** We will return the dictionary to you immediately before the written test. If you don't participate in the written test you can pick the dictionary up at the secretary's office.

Die Beispielaufgaben werden nicht abgegeben oder korrigiert.

Hinweis: Die Größe der Lücken steht nicht unbedingt in Relation zur Länge des Textes, der für eine korrekte Lösung eingefüllt werden muss. Der von Ihnen eingefüllte Text muss kompilierbar sein. Achten Sie auch auf Groß- und Kleinschreibung. Setzen Sie nur die in den Aufgabenstellungen angegebenen Informationen um.

Hinweis: Sie müssen bei der Beantwortung der Präsenzübungsaufgaben keine Vorgaben zum Code-Style einhalten wie sonst bei der Bearbeitung der Übungsblätter. Insbesondere können Sie nichtsprechende und kurze Methoden- und Variablennamen wählen.

Aufbau der Präsenzübung

Für die Präsenzübung werden Sie **20 Minuten Bearbeitungszeit** haben. Sie können **maximal 14 Punkte** erreichen. Die Präsenzübung prüft nur Wissen ab, dass Sie bei der eigenständigen Bearbeitung der Übungsblätter erworben haben. Die Präsenzübung ist auf den Umgang mit der Sprache Java und Implementierungswissen fokussiert. Das bedeutet, Sie müssen wissen, wie Konstrukte wie Klassen, Methoden, Kontrollstrukturen, Schnittstellen, Vererbungsbeziehungen, Variablendeklarationen, Array-Zugriffe oder Attribute in Java syntaktisch ausgedrückt werden und was die Bedeutung dieser Konstrukte ist. Um dieses Wissen nachzuweisen, müssen Sie vor allem kleine Codeausschnitte schreiben oder ergänzen.

¹https://ilias.studium.kit.edu/goto.php?target=crs_851237&client_id=produktiv

Das bedeutet insbesondere, dass kein in der Vorlesung behandeltes Hintergrundwissen, wie beispielsweise Details zur IEEE-Gleitkommazahlendarstellung abgeprüft wird. Weiterhin müssen Sie keine Schnittstellen und Methoden der Java-API oder Namen von Methoden auswendig lernen. Falls nötig, werden diese in den Aufgaben angegeben.

Beispielaufgaben

Aufgabe 1

Ergänzen Sie die folgende Klasse so, dass gilt: **theText** ist ein öffentliches Attribut, das eine Zeichenfolge enthält, **numbers** ist ein privates Attribut, das ein Feld (*Array*) von Ganzzahlen enthält.

```
1 class ValueHolder {  
2     theText;  
3  
4     numbers;  
5 }
```

Aufgabe 2

Ergänzen Sie die folgende Schnittstelle **Worker** um eine Methode **doSomething()**, die ein Feld (*Array*) von Ganzzahlen entgegennimmt und keine Rückgabe hat.

```
1 interface Worker {  
2  
3  
4 }
```

Aufgabe 3

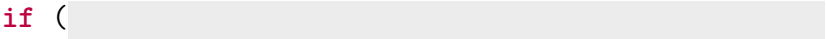
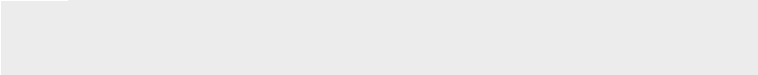
Ergänzen Sie den folgenden Code so, dass beim Aufruf von **doNTimes(int n)** die Methode **doSomething()** n -mal aufgerufen wird. Es gilt immer $n \geq 0$.

```
1 public static void doSomething() { /* ... */ }  
2  
3 public static void doNTimes(int n) {  
4  
5  
6  
7 }
```

Aufgabe 4

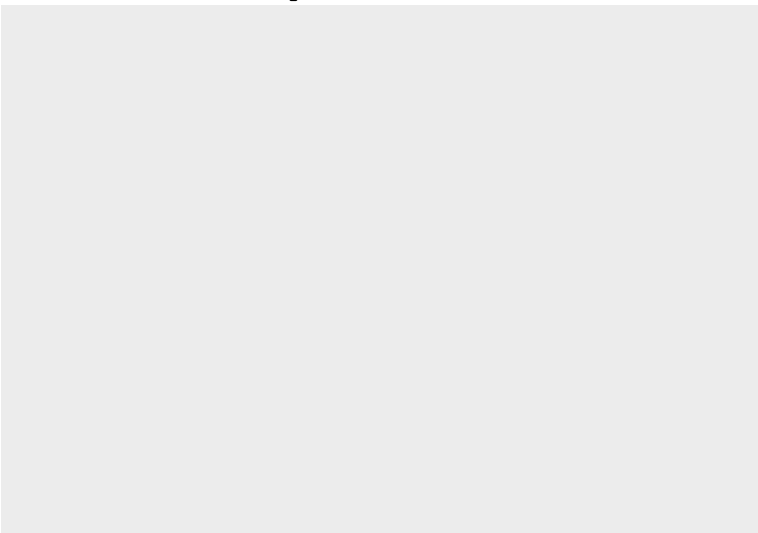
Ergänzen Sie den folgenden Code der Methode `doSomething(int i)` so, dass:

- Wenn `i` ungerade ist, erhöhe `i` um 1,
- wenn dies nicht galt und wenn `i` gerade und größer als oder gleich 3 ist, gebe den Text „Fall 2“ aus.

```
1 public static int doSomething(int i) {  
2  
3     if (  ) {  
4           
5  
6         System.out.println("Fall 2");  
7     }  
8  
9     return i;  
10 }
```

Aufgabe 5

Ergänzen Sie die folgende Methode `printGreaterThan()` so, dass alle Werte des Feldes `values` ausgegeben werden, die echt größer als `threshold` sind. Verwenden Sie zur Ausgabe die Methode `System.out.println`.

```
1 public static void printGreaterThan(int[] values, int threshold) {  
2       
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16 }
```

Lösungen

Lösung zu Aufgabe 1

Ergänzen Sie die folgende Klasse so, dass gilt: `theText` ist ein öffentliches Attribut, das eine Zeichenfolge enthält, `numbers` ist ein privates Attribut, das ein Feld (*Array*) von Ganzzahlen enthält.

```
1 class ValueHolder {
2     public String theText; // oder CharSequence
3
4     private int[] numbers; // oder long[], Integer[], ...
5 }
```

Lösung zu Aufgabe 2

Ergänzen Sie die folgende Schnittstelle `Worker` um eine Methode `doSomething()`, die ein Feld (*Array*) von Ganzzahlen entgegennimmt und keine Rückgabe hat.

```
1 interface Worker {
2     void doSomething(int[] values);
3     // kein Fehler, falls long[], Integer[], ...
4 }
```

Lösung zu Aufgabe 3

Ergänzen Sie den folgenden Code so, dass beim Aufruf von `doNTimes(int n)` die Methode `doSomething()` n -mal aufgerufen wird. Es gilt immer $n \geq 0$.

```
1 public static void doSomething() { /* ... */ }
2
3 public static void doNTimes(int n) {
4     for (int i = 0; i < n; i++) {
5         doSomething();
6     }
7 }
```

Lösung zu Aufgabe 4

Ergänzen Sie den folgenden Code der Methode `doSomething(int i)` so, dass:

- Wenn `i` ungerade ist, erhöhe `i` um 1,
- wenn dies nicht galt und wenn `i` gerade und größer als oder gleich 3 ist, gebe den Text „Fall 2“ aus.

```
1 public static int doSomething(int i) {
2
3     if ((i % 2) == 1) {
4         i++;
5     } else if (i >= 3) {
6         System.out.println("Fall 2");
7     }
8
9     return i;
10 }
```

Lösung zu Aufgabe 5

Ergänzen Sie die folgende Methode `printGreaterThan()` so, dass alle Werte des Feldes `values` ausgegeben werden, die echt größer als `threshold` sind. Verwenden Sie zur Ausgabe die Methode `System.out.println`.

```
1 public static void printGreaterThan(int[] values, int threshold) {
2
3     // Antwortmöglichkeit A
4     for (int i = 0; i < values.length; i++) {
5         if (values[i] > threshold) {
6             System.out.println(values[i]);
7         }
8     }
9
10    // Antwortmöglichkeit B
11    for (int value : values) {
12        if (value > threshold) {
13            System.out.println(value);
14        }
15    }
16 }
```