

# Übungsblatt 1 (15.10.2019)

## Teamorganisation, Java, Anforderungen, Vorbereitung Software-Projekt

### In dieser Übung:

- ✓ Organisieren Sie Ihr Team für die Aufgabenbearbeitung im Verlauf des Semesters.
- ✓ Unternehmen Sie erste Schritte in der Programmiersprache Java.
- ✓ Üben Sie Anforderungsbeschreibung.
- ✓ Bereiten Sie Begriffe zur Organisation eines Projektes vor.

### Hinweise zu den Übungen

Im Wiki zur Vorlesung unter: <https://confluence-se.ifi.uni-heidelberg.de/x/ZgCnBQ> finden Sie wichtige Hinweise zu den Übungsblättern. Bitte lesen Sie diese vollständig durch, bevor Sie mit der Bearbeitung des Übungsblattes beginnen.

### Abgabe Ihrer Lösung in Moodle

Damit Ihre Lösungen möglichst schnell korrigiert werden können, bitten wir Sie, für alle Übungen folgende Konventionen für Ihre Abgabe einzuhalten:

Benennen Sie Ihre Lösung stets mit nachname-aufgabennummer, beispielsweise **mueller-a12**, wenn Sie Müller heißen und Aufgabe 1.2 abgeben.

Diese Namenskonvention gilt für die Abgabe von Android Studio-Projekten und für die Abgabe von PDFs oder anderen Dateien. Sollten Sie in einer Aufgabe sowohl ein Android Studio-Projekt als auch ein PDF abgeben müssen, dann speichern Sie bitte beides (Android Studio-Projekt und PDF) in einem zip-Archiv, das ebenfalls dieser Namenskonvention folgt.

### Hinweise zu den Programmieraufgaben

Sie erhalten in der ersten Technologievorlesung heute Nachmittag (15.10.2019) eine Einführung in Java. Auf diesem und den nächsten Übungsblättern werden Grundlagen zu Java geübt. Ab dem 3. Blatt müssen Sie komplexeren Java-Code (Android App) lesen und ergänzen können.

Diese Woche haben Sie in den Übungsgruppen die Gelegenheit, Ihre Java-Kenntnisse zu vertiefen. Unabhängig von Ihrer Wahl der Gruppe können Sie dazu zu einem der folgenden Termine gehen:

- Mittwoch, 16.10. von 9:15 bis 10:45 Uhr,
- Donnerstag, 17.10. von 14:15 bis 15:45 Uhr oder
- Donnerstag, 17.10. von 16:15 bis 17:45 Uhr

## Aufgabe 1.1: Einschreibung in die Übungsgruppe

<b>Präsenz:</b> Nein	<b>Punkte:</b> keine	<b>Team:</b> Ja	
----------------------	----------------------	-----------------	--

Wie auf der Wiki-Seite mit Hinweisen zu den Übungen unter der **Teameinteilung** beschrieben, müssen Sie sich als 2er-Team für eine der drei Übungsgruppen entscheiden.

- 1) Entscheiden Sie sich zunächst für eine der drei Übungsgruppen. Tragen Sie sich dazu über den folgenden Link in eine Übungsgruppe im MÜSLI-Kurs ein.

<https://www.mathi.uni-heidelberg.de/muesli/lecture/view/1097>

- 2) Bilden Sie anschließend ein 2er-Team. Füllen Sie dafür über den folgenden Link die Umfrage zur Teambildung im Moodle-Kurs aus. Sollten Sie keineN TeampartnerIn für ein 2er-Team finden können, wählen Sie lediglich eine Übungsgruppe und kommen dann zum ersten Übungsgruppentermin am Mittwoch, den 16.10. bzw. Donnerstag, den 17.10.2019. Spätestens in der zweiten Woche wird dann am Mittwoch, den 23.10. bzw. Donnerstag, den 24.10. für alle verbleibenden Gruppenmitglieder die Zuteilung zu einem Team durchgeführt.

<https://elearning2.uni-heidelberg.de/course/view.php?id=22829>

### Ergebnis:

Tragen Sie sich in MÜSLI in eine der Übungsgruppen ein und füllen Sie das Formular zur Teambildung in Moodle bis **Montag 21.10.2019 um 10.00 Uhr** in Moodle aus.

### Installation der Android Studio Entwicklungsumgebung

Vor dem Bearbeiten der Java Programmieraufgaben empfiehlt es sich zunächst die in der Übung verwendete Entwicklungsumgebung Android Studio zu installieren. Verwenden Sie dafür die Anleitung unter: <https://confluence-se.ifi.uni-heidelberg.de/x/XwCnBQ>

Für dieses Übungsblatt genügt es allerdings auch, wenn Sie die Java-Klassen in einem Editor schreiben und nur die .java-Dateien abgeben. Ab dem nächsten Blatt sollten Sie jedoch Android Studio verwenden und bei Programmieraufgaben ein exportiertes Android Studio-Projekt abgeben.

## Aufgabe 1.2: Java (1) – Klassen und Objekte

<b>Präsenz:</b> Nein	<b>Punkte:</b> 9	<b>Team:</b> Nein	<b>Testat</b>
----------------------	------------------	-------------------	---------------

Erstellen Sie die Klasse `Pokemon` sowie den Aufzählungstyp `Type` mit den angegebenen Attributen:

<code>public class Pokemon</code>	<code>public enum Type</code>
<code>private String name;</code>	<code>FIRE, WATER, POISON;</code>
<code>private Type type;</code>	

Beantworten Sie die nachfolgenden Fragen, indem Sie die Klasse `Pokemon` programmieren.

- 1) **Lernziel Konstruktor:** Wie würde ein geeigneter Konstruktor für die Klasse `Pokemon` aussehen, der die Attribute `name` und `type` bei Objekterzeugung initialisiert? Programmieren Sie diesen Konstruktor.
- 2) **Lernziel main, Objektdeklaration und Objekterzeugung:** Erzeugen Sie ein Objekt der Klasse `Pokemon` in der main-Operation der Klasse. Deklarieren Sie das Objekt in einer Zeile und erzeugen Sie das Objekt in einer zweiten Zeile, um sich den Unterschied bewusst zu machen.

- 3) **Lernziel Getter/Setter Operationen:** Erzeugen Sie für die beiden Attribute `name` und `type` jeweils eine `get` und eine `set`-Operation (z.B. `setName`). Sie können dafür die von Android Studio bereitgestellte Hilfsfunktion „Generate...“ -> „Getter and Setter“, die Sie unter „Code“ oder im Rechtsklickmenü finden, benutzen.
- 4) **Lernziel this:** Erläutern Sie die Bedeutung der `this`-Referenz anhand der Operation `setName`, die den Namen zuweist. Schreiben Sie Ihre Erklärung als Kommentar an die Operation.
- 5) **Lernziel statische Attribute:** Erweitern Sie die Klasse `Pokemon` um ein Attribut `number`, welches es ermöglicht, allen Pokemons eine eindeutige, ganze fortlaufende Nummer zuzuordnen. Erweitern Sie die Klasse `Pokemon` um ein statisches Attribut `nextNumber`, welches die jeweils nächste zu vergebende Folgenummer enthält. Setzen Sie die nächste Nummer, sobald ein `Pokemon`-Objekt erzeugt wird.
- 6) **Lernziel Instanz-Operation:** Erweitern Sie die Klasse `Pokemon` um eine Operation `toString`, die die Werte aller Attribute des Objekts in einen String schreibt.

**Ergebnis:**

Speichern Sie bitte Ihre Ergebnisse als .zip-Datei bis **Montag 21.10.2019 um 10.00 Uhr** in Moodle bestehend aus den folgenden Java-Dateien:

- `Pokemon.java` und
- `Type.java`

**Aufgabe 1.3: Java (2) – Testszenario**

<b>Präsenz:</b> Nein	<b>Punkte:</b> 5	<b>Team:</b> Nein	<b>Testat</b>
----------------------	------------------	-------------------	---------------

In dieser Aufgabe erstellen Sie eine Testanwendung „**PokemonTest**“, welche die Klasse `Pokemon` und ihre Operationen testet.

- 1) **Lernziel Test:** Überlegen Sie sich ein Testszenario, in dem alle Operationen der Klasse `Pokemon` benötigt werden. Realisieren Sie dieses in der `main`-Operation einer neuen Klasse `PokemonTest`. Kennzeichnen Sie im Code durch Kommentare, welche Operation Sie testen und ggf. welche wichtigen Fälle Sie dabei unterscheiden. Aus den Ausgaben soll ebenfalls erkennbar sein, was gerade getestet wird, wie der Systemzustand aussieht und was das Ergebnis bei dem jeweiligen Test ist. Orientieren Sie sich dazu an dem nachfolgenden Beispiel:

```
// Test of getter of name
System.out.println();
System.out.println("Testen von getName");
System.out.println("Name von Pokemon p1:");
System.out.println(p1.getName());
```

**Ergebnis:**

Speichern Sie bitte Ihr Ergebnis als .zip-Datei bis **Montag 21.10.2019 um 10.00 Uhr** in Moodle bestehend aus den von Ihnen programmierten, kommentierten, kompilier- und lauffähigen Klassen

- `Pokemon.java`,
- `Type.java` und
- `PokemonTest.java`.

**Pokemon Manager**

In dieser und in späteren Aufgaben erstellen Sie die App PokemonManager, in welcher die Pokemon von Trainern verwaltet und getauscht werden können, sowie gegeneinander in einen Wettkampf treten können

Dabei üben Sie Programmierung mit Java und Android, die Erhebung von Anforderungen und leichtgewichtige Qualitätssicherung in Form von Paarprogrammierung, Coderichtlinien und unsystematischen Funktionstests.

**Aufgabe 1.4: Anforderungen PokemonManager (1)**

<b>Präsenz:</b> Nein	<b>Punkte:</b> 4	<b>Team:</b> Nein	
----------------------	------------------	-------------------	--

Betrachten Sie die folgende Beschreibung einer Systemfunktion des PokemonManagers. Precondition und Postcondition beschreiben den Systemzustand (die gespeicherten Daten), Input und Output beschreiben die Bildschirm- und -ausgabe (ohne konkrete Angaben von GUI-Details wie Layout). Erstellen Sie eine Beschreibung für die Funktionen Delete Pokemon und Edit Pokemon. Orientieren Sie sich dabei an den vorhandenen. Als Ausgabe soll immer die Liste der vorhandenen Pokemons mit den Attributen gezeigt werden.

<b>Name:</b>	Create Pokemon
<b>Preconditions:</b>	none
<b>Input:</b>	none
<b>Postconditions:</b>	New Pokemon exists with ID = generated, type = FIRE, name = ""
<b>Output:</b>	List including new pokemon

**Ergebnis:**

Speichern Sie bitte Ihre Beschreibungen als PDF-Dokument bis **Montag 21.10.2019 um 10.00 Uhr** in Moodle.

**Vorbereitungsaufgaben**

Im Laufe des Semesters wird es mehrere Vorbereitungsaufgaben wie Aufgabe 1.5 geben. Die Vorbereitungsaufgaben sind keine Testaufgaben. Die Themen der Vorbereitungsaufgaben werden in der Vorlesung vertieft. Daher wird vorausgesetzt, dass Sie die Aufgaben bearbeitet haben. Geben Sie Ihre Lösungen in Moodle ab, wenn Sie Rückmeldung zu Ihrem Verständnis der Themen einholen wollen.

**Aufgabe 1.5: Vorbereitung Software-Projekt**

<b>Präsenz:</b> Nein	<b>Punkte:</b> 8	<b>Team:</b> Nein	
----------------------	------------------	-------------------	--

- 1) Bereiten Sie sich auf die nächsten Inhalte der Vorlesung vor, indem Sie das Kapitel **7 „Das Software-Projekt – Begriffe und Organisation“** aus dem Buch von Ludewig und Lichter lesen ([Literatur - Buchkapitel Ludewig und Lichter](#) in **Dateien und Materialien** in Confluence).
- 2) Halten Sie anschließend die Inhalte zu den folgenden Punkten anhand der Informationen aus dem Kapitel in einem PDF-Dokument fest:

- Wichtige Begriffe zu Projekt und Prozess, insbesondere auch die allgemeinen Eigenschaften eines Projektes
- Projekttypen und ihre Unterschiede
- Formen der Teamorganisation und ihre Vor- und Nachteile
- Formen der Sekundärorganisation und ihre Vor- und Nachteile

Diese Aufgabe dient der Vorbereitung des Themas Software-Projekt. Dieses Thema wird in der kommenden Vorlesung am Dienstag, 22.10. vertieft.

**Ergebnis:**

Speichern Sie bitte Ihr PDF-Dokument bis **Montag 21.10.2019 um 10.00 Uhr** in Moodle.