

# Einführung in die Programmierung

WS 2018/2019, Blatt 04

**Ausgabe:** 12.11.2018  
**Abgabetermin:** Montag, 19.11.2018, 11:00 Uhr

Prof. Dr.-Ing. Norbert Siegmund  
M.Sc. André Karge

Bitte lesen Sie die folgenden Informationen zum Übungsablauf **sorgfältig** durch.

Grundsätzlich – wenn nicht anders angegeben – sind die Lösungen zu den Übungen zu Einführung in die Programmierung jeden **Montag bis spätestens 11:00 Uhr** an André Karge per E-Mail zu schicken.

Schreiben Sie bitte im Betreff Ihrer E-Mail Ihre **Teamnummer** sowie die Nummer des Übungsblattes. In der E-Mail schreiben Sie bitte zusätzlich Ihren **Namen** und **Matrikelnummer**. Die Lösungen für Sie bitte als Java Dateien als Anlage hinzu. Es werden **keine** kompilierten Dateien, wie \*.class oder \*.jar angenommen.

Übungen müssen von **minimal zwei** und **maximal drei** Studierenden aus derselben Übungsgruppe in einem festen Team bearbeitet werden (Ausnahmen nur auf Anfrage beim Übungsleiter). Pro Team soll die Lösung nur einmal abgegeben werden. Aufgaben sollen **im Team gelöst** und nicht nur vom Team abgegeben werden. Sie müssen mindestens **50%** dieser Punkte für eine Zulassung zur Prüfung erreichen. Das **Abschreiben** identischer Lösungen wird jeweils mit 0 Punkten bewertet.

Bei Fragen oder Unklarheiten wenden Sie sich bitte **vor der Abgabe** des Übungsblattes an den Übungsleiter (per E-Mail oder persönlich). Es soll nie jemand sagen müssen: „Wir haben die Aufgabe nicht verstanden und konnten sie daher nicht bearbeiten.“

Weitere Informationen, wie aktuelle Ankündigungen, finden Sie online (<https://www.uni-weimar.de/de/medien/professuren/intelligente-softwaresysteme/lehre/>) unter Einführung in die Programmierung

## Aufgabe 1 Schleifen und Arrays (3 Punkte)

Erstellen Sie ein Array, welches die Zahlen von 1 bis 100 enthält. Iterieren Sie anschließend mit einer while-Schleife über dieses Array. Geben Sie in jedem Schleifenschritt folgendes aus:

- Hop wenn die Zahl durch 7 teilbar ist.
- Hip wenn die Zahl eine 7 in der Einerstelle hat.
- die Zahl in Ziffern, wenn weder a) noch b) gilt.

## Aufgabe 2 BattleShip I (8 Punkte)

Schreiben Sie ein Programm, das ein „Ein-Schuss-Schiffeversenken“ realisiert. Legen Sie hierzu in ihrem Programmcode ein festes zweidimensionales Array der Größe  $10 \times 10$  an, das für jede Koordinate  $(x, y)$  speichert, ob sich an dieser Stelle ein Schiffsteil befindet. Überlegen sie sich, welcher Datentyp hierfür am Besten geeignet ist. Als Beispiel können Sie folgende Seekarte verwenden:

						o		o	
	o	o	o			o		o	
								o	
					o	o		o	
o	o								
				o	o	o	o	o	
	o		o	o	o	o			o
	o								o
	o			o	o				o

Fragen Sie den Benutzer nach einer  $x$  und  $y$  Koordinate. Geben Sie dann aus, ob sich an der entsprechenden Stelle ein Schiffsteil befindet. Prüfen Sie auch ab, ob die angegebene Koordinate zulässig ist.

### Aufgabe 3 Battleship II (6 Punkte)

Passen Sie Ihr "Ein-Schuss-Schiffeversenken" von Aufgabe 2 so an, dass der Benutzer mehrere Schüsse abgeben kann. Hierzu setzen Sie nach jedem Treffer auf ihrer Seekarte das getroffene Schiffsteil auf `false`.

**Hinweis:** Um den Benutzer mehrfach schießen zu lassen, benötigen Sie eine `while`-Schleife. Überlegen Sie sich, wie Sie feststellen können, ob das Spiel zu Ende ist.