Einführung in die Programmierung

WS 2017/2018, Blatt 4

Bauhaus-Universität Weimar

Prof. Norbert Siegmund
Nathalie Dittrich

Ausgabe: 20.11.2017
Abgabetermin: Montag,27.11.2017, 11:00 Uhr

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

Grundsätzlich – wenn nicht anders angegeben – sind die Lösungen zu den Übungen zu Programmierung I jeden **Dienstag bis spätestens 12:00 Uhr** an die jeweiligen Tutoren 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 ein** und **maximal zwei** 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, die Angaben finden Sie online (`https://www.uni-weimar.de/de/medien/professuren/intelligente-softwaresysteme/lehre/`) unter Einführung in die Programmierung

Aufgabe 1 Schleifen (3 Punkte)

Iterieren sie mit einer while Schleife über die Zahlen 1 bis 100. Geben sie in jedem Schleifenschritt Folgendes aus:

a) `Hop` wenn die Zahl durch 7 teilbar ist.

b) `Hip` wenn die Zahl eine 7 in der Einerstelle hat.

c) 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 10×10 an, das für jede Koordinate (x, y) speichert, ob sich an dieser Stelle ein Schiff(teil) 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 Schiff(teil) befindet. Prüfen Sie auch ab, ob die angegebene Koordinate zulässig ist.

Aufgabe 3 Battleship II (4 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 versenkte Schiff(steil) 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.